

**IMPLEMENTASI RASPBERRY PI UNTUK MENDIAGNOSIS
PENYAKIT DIABETES MELITUS MELALUI WARNA LIDAH
MENGUNAKAN METODE OTSU'S TRESHOLDING**

SKRIPSI

Untuk memenuhi sebagian persyaratan
memperoleh gelar Sarjana Teknik

Disusun oleh:
Tri Putra Anggara
NIM: 145150301111063



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

PENGESAHAN

IMPLEMENTASI RASPBERRY PI UNTUK MENDIAGNOSIS
PENYAKIT DIABETES MELITUS MELALUI WARNA LIDAH
MENGUNAKAN METODE OTSU'S TRESHOLDING

SKRIPSI

Diajukan untuk memenuhi sebagian persyaratan
memperoleh gelar Sarjana Teknik

Disusun Oleh :
Tri Putra Anggara
NIM: 145150301111063

Skripsi ini telah diuji dan dinyatakan lulus pada
28 Desember 2018
Telah diperiksa dan disetujui oleh:

Dosen Pembimbing I



Rizal Maulana, S.T., M.T., M.Sc
NIK: 2016078910091001

Dosen Pembimbing II



Tibyani, S.T, M.T
NIP: 19691101 199512 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, 28 Desember 2018



Tri Putra Anggara

NIM: 145150301111063

KATA PENGANTAR

Dengan menyebut nama Allah SWT yang Maha Pengasih dan Maha Penyayang, penulis memanjatkan puji syukur karena rahmatnya yang melimpah sehingga laporan skripsi “IMPLEMENTASI RASPBERRY PI UNTUK MENDIAGNOSIS PENYAKIT DIABETES MELITUS MELALUI WARNA LIDAH MENGGUNAKAN METODE OTSU’S TRESHOLDING” ini dapat terselesaikan. Atas banyak dukungan dan bantuan yang diberikan dalam penyusunan skripsi ini, maka penulis ingin menyampaikan rasa hormat dan terimakasih kepada:

1. Alm. Bapak Rusmono dan Ibu Wati selaku orangtua penulis, yang telah sabar mendidik dan membesarkan penulis, serta memberikan doa dan semangat terus-menerus demi terselesainya skripsi ini.
2. Bapak Rizal Maulana , S.T., M.T., M.Sc. dan Tibyani, S.T, M.T selaku dosen pembimbing skripsi penulis yang 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 civitas akademika Fakultas Ilmu Komputer Universitas Brawijaya yang telah banyak memberi bantuan dan dukungan kepada penulis selama menempuh studi Informatika di Universitas Brawijaya dan selama penyelesaian skripsi ini.
5. Sahabat penulis Lamidi, Misbah, Dian Bagus, Fikri Fauzan, Rizky Noriawansyah, Raushan Fikri, Yunus Bahroni dan khususnya Viki Ulva Risqi yang telah memberi banyak dukungan moril maupun materil beserta Keluarga Besar Teknik Komputer khususnya angkatan 2014 yang sudah memberikan semangat.

Penulis menyadari dalam penyusunan skripsi ini masih belum sempurna dan masih memiliki banyak kekurangan, oleh karena itu saran dan kritik sangat dibutuhkan untuk penyempurnaan penelitian ini. Akhir kata penulis berharap semoga skripsi ini dapat memberikan banyak manfaat bagi semua pihak yang menggunakannya.

Malang, 28 Desember 2018

Penulis

Putra.anggara036@gmail.com

ABSTRAK

Kesalahan diagnosis menjadi penyebab perawatan dan pengobatan suatu penyakit menjadi tidak sesuai, bahkan menyebabkan kematian. Beberapa penerapan teknologi mampu mengatasi kesalahan diagnosis. Pada umumnya diagnosis menggunakan teknologi memanfaatkan citra sebagai data yang akan diolah. Salah satu penyakit yang dapat didiagnosis menggunakan citra ialah penyakit diabetes melitus. Pada penelitian ini menggunakan citra lidah untuk mendiagnosis penyakit diabetes melitus. Warna dari citra lidah digunakan sebagai parameter yang digunakan untuk mendiagnosis penyakit diabetes melitus. Proses penentuan penyakit diabetes melitus melalui warna diperoleh dari akuisisi citra menggunakan *webcam* oleh Raspberry Pi dengan metode *Otsu's Thresholding*. Citra lidah sebelum diproses menggunakan *Otsu's Thresholding* dikonversi menjadi *grayscale*, lalu diperbaiki histogramnya dengan metode *Histogram Equalization*. Berdasarkan hasil pengujian, sistem dapat mengakuisisi citra, mengkonversi citra berwarna menjadi *Grayscale*, menyamaratakan histogram dengan metode *histogram equalization* dan mampu mendiagnosis penyakit diabetes melitus melalui warna lidah. Hasil dari pengujian akuisisi citra diperoleh hasil 100%, pengujian sistem diagnosis penyakit diabetes melitus terhadap penderita lidah diabetes melitus diperoleh akurasi 80% dan pengujian sistem diagnosis penyakit diabetes melitus terhadap lidah non penderita diabetes melitus diperoleh akurasi sebesar 90%. Waktu komputasi rata-rata selama pengujian terhadap lidah penderita diabetes melitus 0.3 detik dan waktu komputasi rata-rata selama pengujian terhadap lidah non penderita diabetes melitus 0.4 detik.

Kata kunci: Diagnosis Penyakit, Warna Lidah, Diabetes Melitus, Otsu's Thresholding

ABSTRACT

Misdiagnosis is the cause of treatment and maintenance of the disease to be inappropriate, perhaps even cause of death. Some technological applications are able to overcome misdiagnosis. In general, diagnosis uses technology used images as data to be processed. One disease that can be diagnosed is diabetes mellitus. In this research tongue images used to diagnose diabetes mellitus. The color of the tongue image is a parameter used to diagnose diabetes mellitus. Tongue image obtained from webcam by Raspberry Pi with the Otsu's Thresholding method. The tongue image before processing uses Otsu's Thresholding, image must be converted into grayscale, then Improves the histogram using the Histogram Equalization method. Based on the test results, the company can acquire images, become Grayscale imagery, generalize histograms by histogram equalization method and be able to diagnose diabetes mellitus through tongue color. The result of testing the acquisition image result was obtained on 100 %, testing the system the diagnosis of diseases diabetes mellitus against patients diabetes mellitus obtained accuracy 80 % and testing the system the diagnosis of diseases diabetes mellitus against patients non diabetics mellitus obtained much as 90 % accuracy. The average of the past computing time towards the tongue diabetics mellitus 0.3 seconds and computing the average of the past time towards the tongue non diabetics mellitus 0.4 seconds.

Keywords: Disease Diagnosis, Tongue Color, Diabetes Mellitus, Otsu's Thresholding

DAFTAR ISI

PENGESAHAN	ii
PERNYATAAN ORISINALITAS	iii
KATA PENGANTAR.....	iv
ABSTRAK.....	v
ABSTRACT	vi
DAFTAS ISI	vii
DAFTAR TABEL.....	x
DAFTAR GAMBAR.....	xi
BAB 1 PENDAHULUAN.....	1
1.1 Latar belakang	1
1.2 Rumusan masalah.....	3
1.3 Tujuan penelitian	3
1.4 Manfaat penelitian	3
1.5 Batasan penelitian	4
1.6 Sistematika penulisan	4
BAB 2 LANDASAN KEPUSTAKAAN	6
2.1 Kajian pustaka	6
2.2 Dasar teori	9
2.2.1 Lidah.....	9
2.2.2 Diabetes Melitus	10
2.2.3 Webcam	11
2.2.4 Raspberry Pi	12
2.2.5 Citra digital	16
2.2.6 Jenis citra	17
2.2.7 Pemerataan historam (<i>Histogram Equalization</i>)	20
2.2.8 Pengambangan (<i>Thresholding</i>)	21
2.2.9 <i>Otsu's Thresholding</i>	21
2.2.10 OpenCV	23
2.2.11 <i>Liquid Crystal Display (LCD)</i>	23
BAB 3 METODOLOGI	25
3.1 Studi literatur	26

3.2 Analisis kebutuhan.....	27
3.2.1 Kebutuhan perangkat keras	27
3.2.2 Kebutuhan perangkat lunak	28
3.3 Pengumpulan data.....	28
3.4 Perancangan sistem.....	28
3.4.1 Perancangan perangkat keras	28
3.4.2 Perancangan perangkat lunak	29
3.5 Implementasi sistem.....	30
3.6 Pengujian	30
3.7 Kesimpulan.....	30
BAB 4 REKAYASA KEBUTUHAN.....	31
4.1 Gambaran umum sistem	31
4.2 Identifikasi pengguna.....	31
4.3 Kebutuhan fungsional.....	31
4.4 Kebutuhan non-fungsional	32
4.4.1 Kebutuhan perangkat keras	32
4.4.2 Kebutuhan perangkat lunak	33
4.5 Batasan sistem	33
BAB 5 PERANCANGAN DAN IMPLEMENTASI	34
5.1 Perancangan Sistem.....	34
5.1.1 Perancangan prototype alat	34
5.1.2 Perancangan perangkat keras	34
5.1.3 Perancangan perangkat lunak	36
5.1.3.1 Perancangan akuisisi citra	36
5.1.3.2 Perancangan konversi citra RGB ke Grayscale	37
5.1.3.3 Perancangan <i>Histogram Equalization</i>	38
5.1.3.4 Perancangan <i>Otsu's Thresholding</i>	47
5.2 Implementasi sistem.....	51
5.2.1 Implementasi perangkat keras	51
5.2.2 Implementasi perangkat lunak	52

5.2.2.1 Implementasi kode program inialisasi <i>Library</i>	52
5.2.2.2 Implementasi kode program akuisisi citra	52
5.2.2.3 Implementasi kode program RGB ke Grayscale.....	53
5.2.2.4 Implementasi kode program <i>Histogram Equalization</i>	53
5.2.2.5 Implementasi kode program <i>Otsu's Thresholding</i>	54
BAB 6 PENGUJIAN DAN ANALISIS.....	56
6.1 Pengujian fungsionalitas <i>Webcam</i>	56
6.1.1 Tujuan pengujian	56
6.1.2 Prosedur pengujian.....	56
6.1.3 Hasil dan analisis pengujian.....	56
6.2 Pengujian diagnosis penyakit Diabetes Melitus	57
6.2.1 Tujuan pengujian	57
6.2.2 Prosedur pengujian.....	58
6.2.3 Hasil dan analisis pengujian.....	58
6.3 Pengujian waktu eksekusi program.....	60
6.3.1 Tujuan pengujian	60
6.3.2 Prosedur pengujian.....	60
6.3.3 Hasil dan analisis pengujian.....	60
BAB 7 PENUTUP	62
7.1 Kesimpulan.....	62
7.2 Saran	63
DAFTAR PUSTAKA.....	64
LAMPIRAN A SURAT IZIN PENGAMBILAN DATA	67
LAMPIRAN B DATA LATIH.....	70
B.1 DATA UJI LIDAH PENDERITA DIABETES MELITUS	70
B.2 DATA UJI LIDAH NON PENDERITA DIABETES MELITUS.....	72
LAMPIRAN C DATA UJI	74
C.1 DATA UJI LIDAH PENDERITA DIABETES MELITUS	74
C.2 DATA UJI LIDAH NON PENDERITA DIABETES MELITUS.....	75
LAMPIRAN D KODE PROGRAM SISTEM DIAGNOSIS DIABETES MELITUS.....	75

DAFTAR TABEL

Tabel 2.1 Perbandingan penelitian yang dilakukan	6
Tabel 2.2 Spesifikasi USB <i>Webcam</i>	11
Tabel 2.3 Cara kerja komponen <i>webcam</i>	12
Tabel 2.4 Spesifikasi Raspberry Pi	15
Tabel 2.5 Jenis-jenis ekstensi citra digital	19
Tabel 2.6 Deskripsi pin LCD	24
Tabel 5.1 Kabel hubungan LCD dengan Raspberry Pi	35
Tabel 5.2 Hasil perhitungan <i>Otsu's Thresholding</i> tiap intensitas.....	50
Tabel 5.3 Kode program inialisasi <i>Library</i>	52
Tabel 5.4 kode program akuisisi citra	53
Tabel 5.5 Implementasi kode program konversi citra RGB ke <i>Grayscale</i>	53
Tabel 5.6 Kode program <i>Histogram Equalization</i>	54
Tabel 5.7 Kode program <i>Otsu Thresholding</i>	54
Tabel 6.1 Hasil pengujian fungsionalitas <i>webcam</i>	56
Tabel 6.2 Hasil pengujian diagnosis penyakit diabetes.....	59
Tabel 6.3 Hasil pengujian waktu eksekusi program akuisisi citra	61
Tabel 6.4 Hasil pengujian waktu eksekusi program diagnosis penyakit diabetes melitus.....	61

DAFTAR GAMBAR

Gambar 2.1	Fisik lidah untuk periksa gejala penyakit.....	9
Gambar 2.2	Warna lidah mengindikasikan penyakit	10
Gambar 2.3	<i>Webcam</i>	11
Gambar 2.4	Komponen <i>Webcam</i>	11
Gambar 2.5	Raspberry Pi.....	13
Gambar 2.6	Pin GPIO Raspberry Pi.....	14
Gambar 2.7	Citra berwarna.....	17
Gambar 2.8	Citra <i>Grayscale</i>	18
Gambar 2.9	Citra biner	18
Gambar 2.10	Proses <i>Histogram Equalization</i>	20
Gambar 2.11	LCD	24
Gambar 3.1	Diagram alir metodologi penelitian	25
Gambar 3.2	Blok diagram sistem	28
Gambar 3.3	Flowchart cara kerja	29
Gambar 5.1	Desain <i>prototype</i> alat	34
Gambar 5.2	Diagram skematik diagnosis diabetes melitus	35
Gambar 5.3	Diagram alir akuisisi citra	36
Gambar 5.4	Diagram alir konversi citra RGB ke <i>Grayscale</i>	37
Gambar 5.5	Hasil konversi RGB ke <i>Grayscale</i>	37
Gambar 5.6	Diagram alir <i>Histogram Equalization</i>	38
Gambar 5.7	Histogram data latih	39
Gambar 5.8	Hasil <i>Histogram Equalization</i> data latih	46
Gambar 5.9	Diagram alir <i>Otsu's Thresholding</i>	47
Gambar 5.10	Pembacaan histogram data latih	48
Gambar 5.11	Histogram <i>background</i> dan <i>foreground</i> data latih	48
Gambar 5.12	Hasil segmentasi metode <i>Otsu's Thresholding</i>	50
Gambar 5.13	Rangkaian perangkat komponen sistem.....	51
Gambar 6.1	Hasil akuisisi citra	57
Gambar 6.2	Citra data uji lidah penderita Diabetes Melitus	58
Gambar 6.3	Citra data uji lidah non penderita Diabetes Melitus	59

BAB 1 PENDAHULUAN

1.1 Latar belakang

Diagnosis adalah langkah-langkah untuk menemukan penyakit (*disease*) yang diderita oleh seseorang dengan melalui pengujian dan studi yang seksama mengenai gejala-gejalanya (*symptoms*) (Alang, 2015). Masalah yang berkaitan dengan diagnosis sering menjadi penyebab perawatan dan pengobatan suatu penyakit menjadi tidak sesuai. Penelitian yang dimuat di jurnal *Diabetes Care* menyebutkan sering terjadi kesalahan diagnosis pada pasien yang mengidap diabetes tipe 2. Riset yang dilakukan oleh University of Surrey pada dua juta rekam medis pasien menunjukkan 97.3% pasien penderita penyakit pankreas didiagnosis diabetes tipe 2 (CNN, 2013). Banyaknya kesalahan diagnosis yang terjadi ini mendorong bidang kesehatan untuk menerapkan teknologi dalam membantu mendiagnosis suatu penyakit.

Perkembangan dunia medis beriringan dengan perkembangan teknologi, utamanya untuk kebutuhan diagnosis suatu penyakit. Demi meningkatkan kualitas pelayanan kesehatan, penyediaan alat medis canggih menjadi diperlukan. Beberapa penerapan teknologi dalam mendiagnosis suatu penyakit ialah penggunaan MRI (*Magnetic Resonance Imaging*) yang digunakan untuk merekam gambar potongan organ manusia menggunakan medan magnet berkekuatan 0,064 sampai 1.5 tesla dan resonansi getaran terhadap inti atom hydrogen (Notosiswoyo, 2004). CT (*Computer Tomography*) Scan digunakan untuk memperhatikan suatu kombinasi dari jaringan, pembuluh darah dan tulang secara bersamaan (Ramadhani, 2006). BMD (*Bone Mineral Density*) digunakan untuk mendiagnosis osteoporosis, menilai resiko patah tulang dan memantau perkembangan respon pasien terhadap pengobatan yang sedang dijalani (Hamdy, 2016). Penerapan teknologi untuk mendiagnosis juga diterapkan dalam mendiagnosis penyakit diabetes melitus.

Diabetes melitus adalah penyakit yang terjadi pada gangguan metabolisme, dikarenakan pankreas tidak cukup memproduksi insulin atau tubuh kurang bisa memanfaatkan insulin secara efektif, akibatnya kadar gula dalam darah meningkat (Kemenkes, 2014). *World Health Organization* (WHO) menyatakan Diabetes merupakan satu penyakit dari empat prioritas penyakit tidak menular. Diabetes merupakan penyakit yang menjadi penyebab utama kebutaan, serangan jantung, stroke maupun amputasi kaki (South-East Asia Regional Office WHO, 2015). Pada tahun 2015 terdapat 415 juta orang dewasa menderita diabetes. Tahun 2040 diperkirakan jumlah penderita diabetes melitus meningkat hingga mencapai 642 juta jiwa penderita. Penderita diabetes di Indonesia pada tahun 2015 menduduki peringkat ke tujuh dunia dengan jumlah estimasi penderita 10 juta jiwa (IDF Atlas, 2015). Jumlah keseluruhan kasus penyakit diabetes yang terjadi di Indonesia tahun 2007 sebesar 5.7% meningkat hingga 6.9% pada tahun 2013.

Diagnosis penyakit diabetes melitus dapat dilakukan dengan berbagai cara, yang pertama ialah tes plasma acak, tes dilakukan tanpa harus puasa gula darah sebelumnya, jika glukosa pada darah menunjukkan 200 mg/dl atau lebih dari 200 mg/dl kemungkinan menderita diabetes. Kedua tes puasa gula darah sebelumnya harus melakukan puasa gula darah selama delapan jam sebelum tes, jika glukosa yang terkandung pada darah melebihi 126 mg/dl pada dua atau lebih tes yang dilakukan maka kemungkinan menderita diabetes. Ketiga Tes toleransi glukosa oral ini dilakukan ketika tes glukosa plasma acak adalah 160-200 mg/dl dan tes puasa gula darah adalah 110-125 mg / dl. Tes darah ini mengevaluasi respons tubuh terhadap glukosa. Tes ini membutuhkan puasa gula darah setidaknya delapan jam dan tidak lebih dari 16 jam. Kadar puasa gula darah ditentukan, selama dua atau tiga jam, tiap 30 menitnya diberikan 75 gram glukosa dan 100 gram glukosa untuk wanita hamil. Tes dikatakan tidak menderita diabetes jika kadar gula darah selama tes kurang dari 140 mg / dl dan dikatakan diabetes jika kadar glukosa serum pada dua jam pertama akan tetap sama bahkan lebih dari 200 mg/dl (Baynest, 2015).

Beberapa cara diagnosis penyakit diabetes melitus tersebut menghabiskan biaya yang banyak, untuk mendapatkan hasil diagnosis juga diperlukan waktu yang lama, dan rasa sakit yang ditimbulkan ketika pasien dilakukan pengambilan sample darah. Untuk mengatasi hal tersebut penelitian ini menggunakan citra sebagai sampel yang diteliti. Pengolahan citra yang dapat dilakukan dalam mendiagnosis suatu penyakit salah satunya ialah penelitian yang dilakukan oleh Rahmadewi dan Kurnia yang berjudul "Klasifikasi Penyakit Paru Berdasarkan Citra Rontgen dengan Metoda Segmentasi Sobel" Citra hasil rontgen paru-paru dideteksi tepi dengan operator sobel dengan ukuran matrik 3x3. Setelah hasil region paru didapatkan langkah selanjutnya proses deteksi tepi untuk memperjelas perbedaan rentang grayscale satu piksel terhadap piksel tetangganya (Rahmadewi, Kurnia, 2016). Penelitian kedua dilakukan oleh Muwardi dan Fadlil yang berjudul "Sistem Pengenalan Bunga Berbasis Pengolahan Citra dan Pengklasifikasi Jarak" Sistem ini melakukan pengenalan jenis-jenis bunga berbasis komputer merupakan proses memasukkan informasi berupa citra jenis bunga ke dalam komputer. Selanjutnya dilakukan *pre-processing* dengan mengubah citra berwarna menjadi *grayscale* serta operasi pemotongan gambar, untuk ekstraksi ciri menggunakan histogram dari warna citra dan statistik orde 1, dan pengklasifikasi jarak manhattan dan Euclidean (Muwardi, Fadlil, 2017). Penelitian yang terakhir dilakukan oleh Sidiq dan Irmawati yang berjudul "Pengolahan Citra untuk Identifikasi Telur Berdasarkan Ukuran" Penelitian ini untuk mengembangkan sistem yang dapat mendeteksi dan memisah telur berdasarkan ukuran menggunakan citra digital. Pengolahan citra dimulai proses konversi warna RGB ke *Grayscale*, konversi biner menggunakan metode *threshold* dari citra warna skala keabu-abuan dan menghilangkan *noise*. Hasil keluaran berupa klasifikasi ukuran telur. Sistem kendali cerdas yang digunakan untuk menghasilkan keputusan adalah kendali Logika Fuzzy dengan metode Mamdani (Sidiq, Irmawati, 2016). Peneliti dalam pengolahan citra untuk deteksi warna lidah guna mendiagnosis penyakit diabetes dalam penelitian ini mengusulkan Otsu's Tresholding.

Metode *Otsu's Thresholding* sesuai untuk mencari nilai *threshold* dari suatu citra warna lidah. Sebelum dilakukan proses *Otsu's Thresholding* dilakukan pengkonversian citra warna RGB ke *Grayscale*. Hasil citra akan lebih maksimal dengan perbaikan citra dengan menggunakan *Histogram Equalization* sehingga penyebaran citra menjadi rata dan kondisi citra makin baik setelah itu dilakukan perhitungan nilai *threshold* menggunakan *Otsu's Thresholding*. Pada *Otsu's Thresholding* perhitungan *threshold* dilakukan secara otomatis berdasarkan citra lidah penderita diabetes. *Otsu's Thresholding* menggunakan pendekatan analisis diskriminan, dengan menentukan suatu variable yang dapat membedakan antara dua atau lebih kelompok yang muncul secara alami. Untuk memisahkan objek dengan latar belakang analisis diskriminan akan memaksimumkan variable tersebut (Utami, 2017). Dengan demikian, metode *Otsu's Thresholding* sesuai untuk mencari nilai *threshold* dari sebuah citra. Berdasarkan latar belakang tersebut, dengan adanya sistem diagnosis penyakit diabetes melitus melalui warna lidah menggunakan metode *Otsu's Thresholding* pengguna dapat membantu membuat keputusan dalam mendiagnosis penyakit diabetes melitus. Sehingga sistem ini dapat digunakan dengan banyak pengguna, dengan semakin banyak pengguna maka dapat mengetahui suatu penyakit diabetes melitus sedini mungkin.

1.2 Rumusan masalah

1. Bagaimana cara penerapan webcam untuk digunakan dalam pengambilan citra lidah?
2. Bagaimana perancangan dan implementasi *Otsu's Thresholding* untuk mendiagnosis penyakit diabetes melalui warna lidah dengan Raspberry Pi?
3. Bagaimana akurasi kinerja sistem diagnosis penyakit diabetes melitus melalui warna lidah?

1.3 Tujuan penelitian

1. Menerapkan webcam untuk digunakan dalam pengambilan citra lidah
2. Merancang dan mengimplementasikan *Otsu's Thresholding* untuk mendiagnosis penyakit diabetes melalui warna lidah dengan Raspberry Pi
3. Melakukan pengujian akurasi kinerja sistem diagnosis penyakit diabetes menggunakan warna lidah

1.4 Manfaat penelitian

1. Mempermudah dalam menentukan pengambilan keputusan apakah seorang tersebut menderita penyakit diabetes atau tidak
2. Memberikan pengetahuan dan wawasan kepada pembaca tentang penerapan metode *Otsu's Thresholding* dan sistem diagnosis penyakit melalui warna lidah menggunakan Raspberry Pi

1.5 Batasan penelitian

1. Faktor penyakit diabetes melitus yang diperhatikan ialah warna lidah
2. Obyek yang diamati ialah lidah penderita diabetes melitus usia 15 tahun ke atas
3. Sistem dapat bekerja optimal pada pencahayaan yang cukup
4. Jarak pengambilan citra lidah dengan *webcam* yang terletak pada alat sejauh 30 cm

1.6 Sistematika penulisan

Sistematika penulisan dalam skripsi ini sebagai berikut:

BAB 1 Pendahuluan

Bab pendahuluan berisi latar belakang, rumusan masalah, tujuan penelitian, manfaat penelitian, batasan masalah, dan sistematika pembahasan.

BAB 2 Kajian Pustaka

Bab ini membahas kajian pustaka dari penelitian-penelitian terdahulu dan dasar teori yang mendukung dalam pembuatan penelitian ini.

BAB 3 Metode Penelitian

Bab ini membahas tentang langkah-langkah dalam melakukan penelitian ini, antara lain studi literatur, analisis kebutuhan sistem, perancangan sistem, implementasi sistem, pengujian dan analisis, dan kesimpulan.

BAB 4 Rekayasa Kebutuhan

Bab ini membahas tentang rekayasa kebutuhan mulai dari pendefinisian kebutuhan perangkat lunak dan kebutuhan perangkat keras untuk dapat merealisasikan sistem diagnosis penyakit diabetes melitus melalui warna lidah menggunakan metode *otsu thresholding*.

BAB 5 Perancangan dan Implementasi

Sistem Diagnosis Penyakit Diabetes Melitus melalui Warna Lidah Menggunakan *Otsu's Tresholding* Berbasis Raspberry Pi. Proses dimulai dari perancangan dan implementasi sistem berupa pengambilan gambar lidah dimana sistem akan mengambil gambar menggunakan *webcam*, gambar hasil pengambilan akan dikirimkan ke Raspberry Pi, kemudian gambar diproses menggunakan metode *Otsu's Tresholding* untuk mendiagnosis penyakit diabetes melitus, jika terdiagnosa maka sistem akan menampilkan teks pada LCD bahwa orang tersebut terdiagnosa diabetes melitus.

BAB 6 Pengujian dan Analisis

Bab ini membahas tentang pembahasan hasil yang dihasilkan dari implementasi sistem. Terdiri dari sub Pengujian dan Analisis. Bab ini membahas tentang cara pengujian dan analisis dari hasil pengujian sistem.

BAB 7 Penutup

Penutup membahas tentang kesimpulan yang didapat dari hasil perancangan, implementasi dan pengujian serta analisis yang telah dilakukan, serta saran-saran untuk pengembangan penelitian selanjutnya.



BAB 2 LANDASAN KEPUSTAKAAN

2.1 Kajian pustaka

Pada penelitian ini, peneliti mencantumkan beberapa hasil penelitian sebelumnya sebagai acuan atau pertimbangan dalam penelitian ini. Perbandingan penelitian dahulu yang telah dilakukan dengan rencana peneliti dapat dilihat pada Tabel 2.1.

Tabel 2.1 Perbandingan Penelitian Yang Dilakukan

No.	Nama Penulis, Tahun dan Judul	Persamaan	Perbedaan	
			Penelitian Terdahulu	Rencana Penelitian
1	Saritha, dkk. [2013] <i>Disease Analysis Using Tongue Image. International Journal of Engineering Research & Technology Vol.2, Issue 4</i>	Melakukan Analisis penyakit menggunakan citra lidah	Penelitian ini melakukan diagnosis diabetes melitus berdasarkan analisis kuantitatif tentang perubahan patologis pada permukaan lidah.	Mengimplementasikan metode <i>Otsu's Thresholding</i> pada lidah penderita diabetes melitus untuk mencari nilai <i>Threshold</i>
2	Devi, Uma, dkk. [2016] <i>Diagnosis of Diabetes Using Tongue Color Analysis. IJETER Vol. 4 Issue 5</i>	Melakukan diagnosis penyakit diabetes menggunakan warna lidah	Penelitian ini membandingkan 12 warna lidah berpenyakit yang telah tersedia dengan lidah yang akan didiagnosis.	Melakukan akuisisi warna citra lidah lalu dicari nilai thresholdnya menggunakan <i>Otsu's Thresholding</i>
3	Zhang, Junwen, dkk. [2015] <i>Extraction of Tongue Feature Related to TCM Physique Based on Image Processing. Beijing University of Technology.</i>	Menggunakan pengolahan citra lidah untuk mensegmentasi penyakit.	Mengklasifikasi citra lidah untuk dimasukkan kategori sesuai ciri fisik lidah berdasarkan Asosiasi Pengobatan Cina	Menggunakan warna lidah penderita diabetes melitus untuk dijadikan data latih sistem.

No	Nama Penulis, Tahun dan Judul	Persamaan	Perbedaan	
			Penelitian Terdahulu	Rencana Penelitian
4	Prasetyo, Eko, dkk. [2017] <i>Mango Leaf Image Segmentation on HSV and YCbCr Color Spaces Using Otsu Thresholding International Conference on Science and Technology</i>	Menggunakan <i>Otsu Thresholding</i> dalam mensegmentasi objek berdasarkan warna	Menggunakan HSV dan YCbCr untuk mensegmentasi warna menggunakan <i>Otsu Thresholding</i>	Menggunakan warna <i>Grayscale</i> dalam mensegmentasi menggunakan <i>Otsu Thresholding</i>
5	Khatak, Anil dan Anju [2017] <i>A Study on Image and Its Technique. Guru Jambheshwar University Science & Technology Hisar</i>	Melakukan segmentasi menggunakan citra	Menerapkan Teknik segmentasi <i>threshold, edge base dan region based</i>	Menerapkan <i>Otsu Threhsold</i> dalam mensegmentasi citra
6	Nabella, dkk, [2013] Analisis Citra Sinar-X Tulang Tangan Menggunakan Metode <i>Thresholding Otsu</i> Untuk Identifikasi Osteoporosis. Universitas Tanjungpura Pontianak	Menggunakan <i>Otsu Thresholding</i> untuk mendiagnosis penyakit	Menggunakan <i>Otsu Thresholding</i> untuk mendeteksi osteoporosis melalui citra sinar-x tulang tangan	Menggunakan <i>Otsu Thresholding</i> untuk mendeteksi diabetes melalui warna lidah

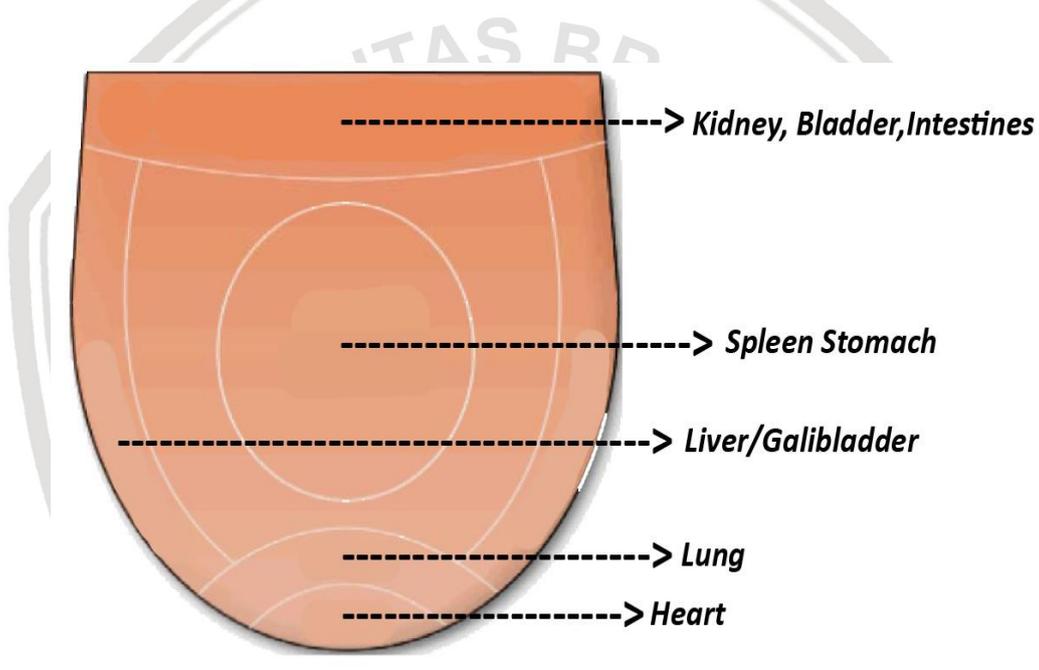


No	Nama Penulis, Tahun dan Judul	Persamaan	Perbedaan	
			Penelitian Terdahulu	Rencana Penelitian
7	Mirnasari dan Adi [2013] Aplikasi Metode Otsu Untuk Identifikasi Bakteri Tuberkulosis Secara Matematis. Universitas Diponegoro	Menggunakan otsu untuk mensegmentasi suatu objek	Menggunakan Otsu Thresholding Fuzzy dan ANFIS untuk mendeteksi bakteri tuberculosi	Mengimplementasikan metode <i>Otsu's Thresholding</i> pada lidah penderita diabetes melitus untuk mencari nilai <i>Threshold</i>
8	Huang, Bo, dkk, [2009] <i>Quantified Vector Oriented Tongue Color Classification.</i> Harbin Institute of Technology	Menerapkan diagnosis terhadap warna lidah	Menggunakan algoritma vector quantization dalam mengklasifikasikan warna lidah	Menggunakan citra <i>grayscale</i> dari lidah yang diperbaiki dengan <i>histogram Equalization</i> lalu mengklasifikasinya menggunakan <i>Otsu Thresholding</i>
9	Bob Zhang, dkk. [2014] <i>Detecting Diabetes Mellitus and Nonproliferative Diabetic Retinopathy sing Tongue Color, Texture, and Geometry Features.</i> IEEE	Menggunakan warna lidah untuk mendiagnosis penyakit diabetes melitus	Mendeteksi Diabetes Melitus dan Nonproliferative Diabetic Renopathy dengan Warna lidah, tekstur dan fitur geometri lidah	Mendeteksi diabetes melitus dengan warna lidah
10	Kawanabe, dkk. [2016] <i>Quantification of Tongue Color Using Machine Learning in Kampo Medicine.</i> Malaysia – Japan International Institute of Technology	Menkuantifikasi warna lidah untuk kegunaan yang berhubungan dengan medis	Menggunakan <i>machine learning</i> untuk mengkuantifikasi warna lidah	Menggunakan <i>Otsu Thresholding</i> untuk keperluan medis menggunakan warna lidah

2.2 Dasar teori

2.2.1 Lidah

Lidah merupakan organ muskuler yang sangat fleksibel serta memiliki bentuk yang lentur. Pada saat istirahat lidah menempati bagian terbesar rongga mulut utama. Lidah terdiri dari radiks lingua, korpus lingua dan apeks lingua. Fungsi lidah berhubungan dengan proses-proses penghancuran makanan, mengunyah makanan, mengecap, berbicara dan membersihkan mulut, tetapi fungsi utamanya untuk mengantarkan makanan ke dalam faring ketika menelan dan membentuk kata-kata saat berbicara (Tanudjaja, 2013). Fungsi lain lidah ternyata dapat menjadi indikator menderita suatu penyakit dengan melihat ciri fisik pada lidah. Bagian-bagian dari berbeda lidah ini bisa digunakan untuk pemeriksaan penyakit yang terjadi pada tubuh, seperti yang ditampilkan pada Gambar 2.1.



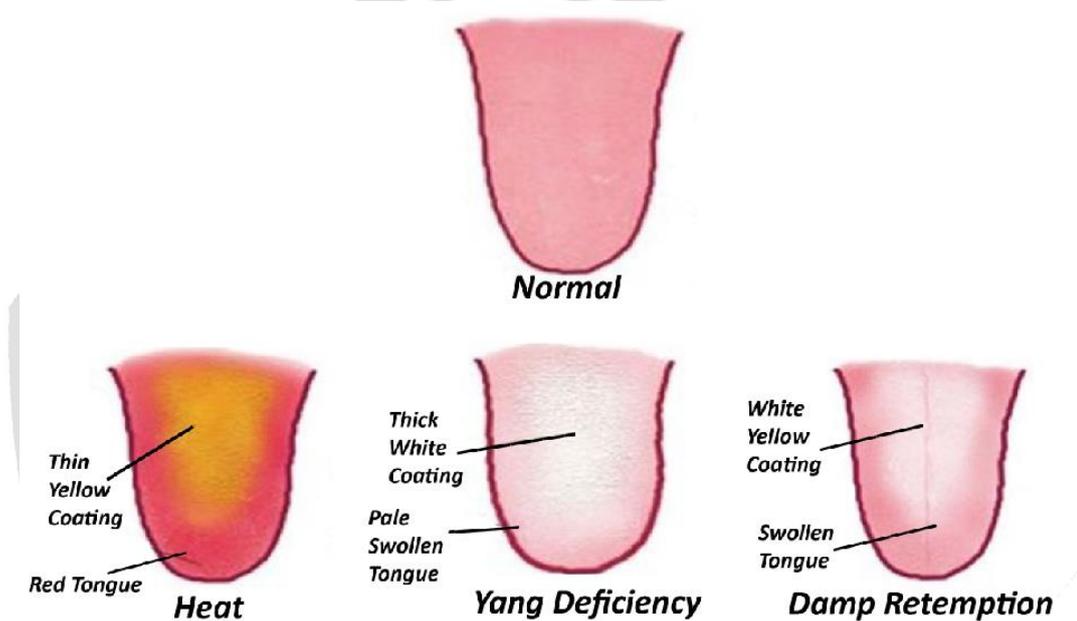
Gambar 2.1 Fisik lidah untuk periksa gejala penyakit

Sumber: Uma Devi, dkk., (2016)

Diagnosis Lidah merupakan metode diagnostik yang banyak digunakan dalam *Traditional Chinese Medicine*. Lidah manusia mengandung status kesehatan dan informasi fisik. Informasi yang didapat dari lidah diantaranya warna lidah, ukuran lidah dan distribusi lapisan lidah. Untuk memenuhi permintaan otomatisasi dan aplikasi pengumpulan informasi yang fleksibel di rumah sakit, segmentasi lidah secara otomatis menjadi hal yang sangat penting (Wei, 2010). Fisik lidah adalah salah satu unsur yang paling penting.

2.2.2 Diabetes Melitus

Diabetes merupakan penyakit gangguan metabolik yang sudah bertahun-tahun akibat pankreas tidak mampu memproduksi insulin yang cukup atau tubuh tidak mampu memanfaatkan insulin dengan efektif yang telah diproduksi oleh tubuh. Insulin merupakan hormone yang mengatur keseimbangan antara gula darah. Akibatnya terjadi hiperglikemia yaitu peningkatan konsentrasi glukosa di dalam darah. Terdapat dua kategori utama diabetes yaitu tipe 1 dan tipe 2. Tipe 1 disebut *insulin dependent* atau *juvenile/childhood-onset diabetes*. Diabetes tipe 2, disebut *non-insulin-dependent* atau *adult-onset diabetes* (InfoDATIN Kemenkes RI, 2013). Dalam mendiagnosis penyakit dapat menggunakan warna pada lidah seperti yang ditampilkan pada Gambar 2.2.



Gambar 2.2 Warna lidah Mengindikasikan penyakit

Sumber: Scheid (2007)

Lidah dianggap sebagai cermin dari viscera. Ini dapat digunakan untuk diagnosis penyakit dengan fitur warnanya menjadi sangat penting. Ujung depan sepertiga lidah berhubungan dengan paru-paru, jantung, dada dan leher. Daerah pusat berhubungan dengan hati, limpa, lambung dan pankreas. Bagian belakang sepertiga lidah dasar berhubungan dengan organ perut bagian bawah, usus kecil, dan usus besar. Setiap gangguan pada organ yang berhubungan dengan area tersebut direfleksikan oleh perubahan warna atau sensitivitas di area tersebut. Penyakit diabetes pun dapat didiagnosis dengan melihat warna lidah.

2.2.3 Webcam

Webcam merupakan istilah bagi kamera *real-time* yang gambarnya bisa dilihat melalui program pengolah pesan atau aplikasi pemanggilan video. Webcam ialah sebuah kamera digital yang berukuran kecil dihubungkan ke komputer melalui USB atau colokkan COM. (Kurniawan, Mutaqqin, 2015). Fungsi dari webcam untuk mengambil gambar dalam mengolah pesan cepat seperti percakapan melalui video secara waktu-nyata. Webcam juga berfungsi sebagai penangkap citra untuk masukkan dari Raspberry Pi seperti yang ditunjukkan pada Gambar 2.3.



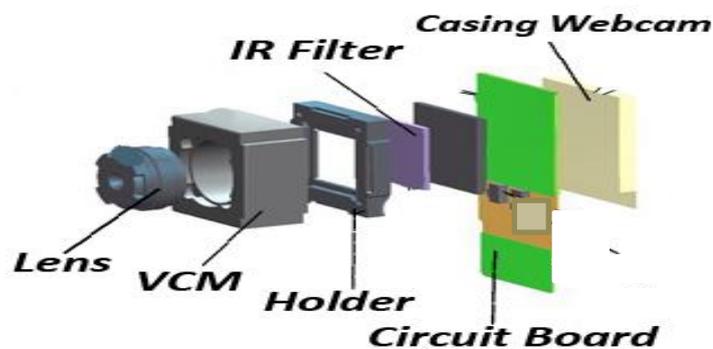
Gambar 2.3 Webcam

Webcam yang digunakan untuk mengakuisisi citra lidah memiliki spesifikasi yang ditunjukkan pada Tabel 2.2:

Tabel 2.2 Spesifikasi USB Webcam

Jarak Fokus Kamera	Resolusi Video	Resolusi Photo	Ukuran
20 MM	640*480	2 MP	3.8*1.5*3 cm

Akuisisi Citra Lidah menggunakan USB Webcam dengan resolusi 2 MP, citra dari lidah dapat diolah dengan cepat. Namun jika menggunakan webcam yang memiliki resolusi lebih besar maka proses pengolahan citra akan semakin lambat. Komponen pembentuk webcam dapat dilihat pada Gambar 2.4.



Gambar 2.4 Komponen Webcam

Webcam terdiri dari lensa standar, dipasang pada sebuah papan sirkuit untuk menangkap sinyal gambar, VCM mengatur focus kamera, dilengkapi casing untuk menutupi lensa. Kabel terbuat dari bahan fleksibel, salah satu ujungnya di hubungkan ke sirkuit ujung lainnya ke *connector*. Cara kerja masing-masing komponen penyusun kamera dijelaskan pada Tabel 2.3.

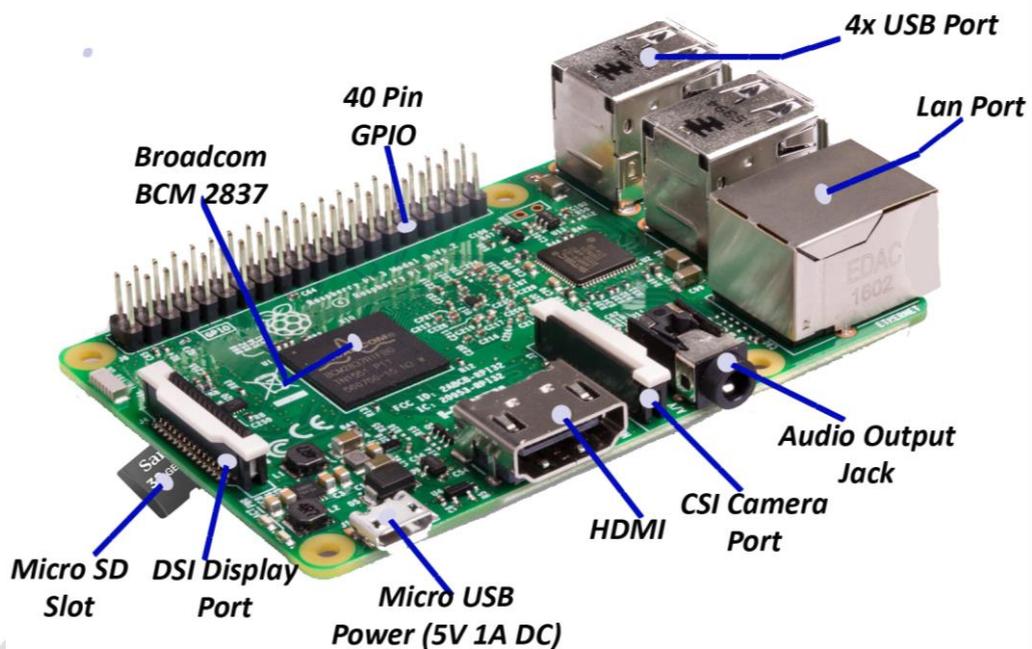
Tabel 2.3 Cara kerja komponen Webcam

Komponen	Cara Kerja
Lens	Fungsi utama lensa kamera ialah mengumpulkan cahaya. Aperture dari sebuah lensa adalah diameter bukaan lensa dan biasanya dikontrol oleh iris. Semakin besar diameter aperture, semakin banyak cahaya yang masuk kedalam lensa
VCM (<i>Voice Coil Motor</i>)	VCM menggunakan prinsip magnetik, dengan menggerakkan posisi lensa kamera untuk mendapatkan fokus. Gerak maju mundurnya VCM akan didapatkan tingkat focus gambar.
Holder	Holder berfungsi sebagai penyangga slave unit. Pada webcam sudah tertanam di bagian dalam.
IR Filter	filter IR digunakan untuk menyaring yang hanya cahaya Infra Merah dan menghadang spektrum cahaya selain infra merah.
Circuit Boards	Circuit Board digunakan sebagai tempat menyusun komponen-komponen elektronik sehingga terpasang lebih terorganisir, dan pengganti kabel untuk menyambung berbagai komponen, sehingga lebih efisien.
Casing Webcam	Casing digunakan sebagai rumah untuk pemasangan komponen-komponen webcam dan untuk melindungi komponen yang ada didalam casing.

Sumber: Shibuya (2018)

2.2.4 Raspberry Pi

Raspberry Pi merupakan *single board computer* yang dapat disebut juga *mini komputer personal*. Sistem operasi Raspberry Pi utamanya ialah Linux atau Debian GNU dengan bahasa pemrograman Python. Raspberry yang digunakan pada penelitian ini yaitu model B yang memiliki penambahan port ethernet dan 2 port USB dengan model sebelumnya. *Peripheral* Raspberry Pi ditunjukkan Gambar 2.5.



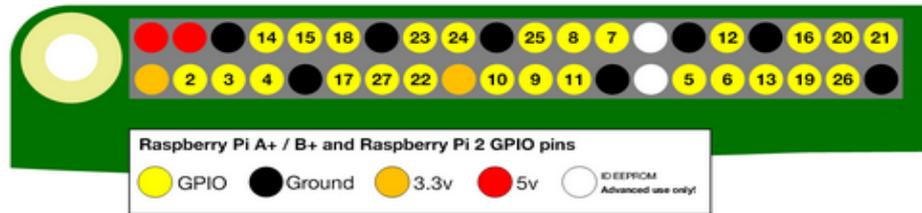
Gambar 2.5 Raspberry Pi

Sumber: <https://www.raspberrypi.org/products/raspberry-pi-3-model-b-plus/> (2016)

Berikut penjelasan dari gambar 2.5:

1. Prosesor: Broadcom BCM 2837 berfungsi sebagai otak dari Raspberry Pi.
2. USB 2.0: Pada Raspberry Pi 3 Model B+ terdapat 4 port USB. *Universal Serial Bus* merupakan standar bus serial untuk perangkat penghubung
3. *Lan Port*: Kabel *Ethernet* memungkinkan Raspberry Pi terhubung internet dengan stabil dan cepat
4. *Audio Output Jack*: *Socket headphone audio* yang memiliki ukuran 3.5 mm, sehingga Raspberry Pi mampu terhubung dengan perangkat audio
5. *CSI (Camera Serial Interface) Port*: CSI memiliki satu jalur jam dan satu port 12C yang digunakan untuk menkonfigurasi kamera dan mengirim perintah maupun gambar/video yang diterima melalui jalur data.
6. HDMI: Port yang mampu menghubungkan Raspberry Pi dengan monitor atau televisi.
7. *Micro USB Power*: Power yang memungkinkan raspberry untuk mendapat daya dari *charger smartphone* atau adaptor yang cocok dengan *port* tersebut.
8. *DSI (Display Serial Interface)*: DSI mempunyai 15 yang digunakan untuk display LCD
9. *Micro SD Slot*: Merupakan tempat untuk meletakkan micro SD sebagai media penyimpanan Raspberry Pi.

10. *Pin GPIO (General Purpose Input/Output)*: berguna untuk menghubungkan Raspberry Pi dengan rangkaian elektronik lainnya. Gambaran 40 Pin GPIO dalam Raspberry Pi dapat dilihat pada Gambar 2.6



Gambar 2.6 Pin GPIO Raspberry Pi

Sumber: <https://www.raspberrypi.org/documentation/usage/gpio/> (2017)

GPIO Terletak disisi bagian atas Raspberry PI. 40 pin GPIO ini terdapat pada semua jenis Raspberry Pi kecuali Pi Zero. Penjelasan detail gambar 2.4 akan dijelaskan sebagai berikut:

1. **Tegangan:** Terdapat Dua pin 5V dan dua pin 3.3 V di papan GPIO, serta sejumlah pin ground (0V), yang tidak dapat dikonfigurasi. Pin yang tersisa adalah semua pin 3.3 V, yang berarti output dan input memiliki tegangan sekitar 3.3 V.
2. **Output:** Pin GPIO yang ditunjuk sebagai pin output dapat disetel ke *HIGH* (3.3 V) atau *LOW* (0V).
3. **Input:** Masukan Pin GPIO yang ditetapkan sebagai pin *input* dapat dibaca sebagai *HIGH* (3.3 V) atau *LOW* (0V). Ini menjadi lebih mudah karena pada pin GPIO2 dan GPIO3 telah terdapat resistor pull-up internal atau resistor pull-down.
4. **PWM (*Pulse-Width Module*):** GPIO12, GPIO13, GPIO18, dan GPIO19 merupakan pin PWM
5. **SPI:** Terdapat 2 SPI Pada pin GPIO Raspberry Pi 3 Model B+ yaitu SPI0 yang terdiri dari pin GPIO10 pin GPIO20 berfungsi sebagai MOSI, Pin GPIO9 pin GPIO19 sebagai MISO), pin GPIO11 pin GPIO21 sebagai SCLK, pin GPIO8 pin GPIO18 sebagai CE0, dan pin GPIO7 pin GPIO17 sebagai CE1 dan pin GPIO16 sebagai CE2
6. **I2C (*Inter-integrated Circuit*):** pada I2C terdapat pin Data pada GPIO2, clock pada GPIO3, EEPROM Data pada GPIO0, dan EEPROM Clock pada GPIO1
7. **Serial:** Pada serial terdapat TX yang terletak di GPIO14 dan RX di GPIO15

Spesifikasi Raspberry Pi dijelaskan pada Tabel 2.4 menurut (Sauki & Septian, 2013):

Tabel 2.4 Spesifikasi Raspberry Pi

Spesifikasi	Raspberry Pi
<i>System on Chip (SoC)</i>	<i>Broadcam BCM2837</i>
<i>CPU</i>	<i>4X ARM Cortex-A53, 1.2 GHz</i>
<i>GPU</i>	<i>Broadcam Video Core IV</i>
<i>RAM</i>	<i>1GB SDRAM</i>
<i>Bluetooth</i>	<i>4.1 Classic</i>
<i>Networking</i>	<i>10/100 Ethernet (8P8C)</i>
<i>Storage</i>	<i>MicroSD</i>
<i>GPIO</i>	<i>40-pin header</i>
<i>Ports</i>	<i>3.5 mm, analogue, audio-video jack. HDMI, ethernet, 4x USB 2.0, CSI, DSI</i>
<i>Suplai Daya</i>	<i>5V melalui GPIO Header atau Micro USB</i>
<i>Konsumsi Daya</i>	<i>3.5 W (700mA)</i>
<i>Panjang</i>	<i>85.6 mm</i>
<i>Lebar</i>	<i>53.98 mm</i>
<i>Berat</i>	<i>45 gram</i>
<i>Sistem Operasi</i>	<i>Arch Linux RAM, Raspbian OS, Gentoo, Debian, FreeBSD, RISC OS</i>

Sumber: Sauki & Septian (2013)

Cara mengoperasikan Raspberry Pi menggunakan laptop harus disiapkan beberapa hal dibawah ini:

1. Kabel Cross
2. Raspberry Pi yang telah dipasang Operasi Sistem
3. Catu daya raspberry Pi dengan keluaran 5V dan 2.1 A
4. Koneksi WiFi
5. Laptop
6. Software Putty
7. Software Ip Scanner
8. Software VNC

Cara kerja Raspberry Pi menggunakan laptop, ialah:

1. Pasang catu daya Raspberry Pi
2. Hubungkan kabel cross ke laptop
3. Aktifkan berbagi koneksi jaringan wifi ke kabel cross
4. Buka aplikasi Ip Scanner untuk melihat IP dari Raspberry Pi
5. Buka aplikasi putty dengan memasukkan IP Raspberry Pi
6. Masukkan *username* dan *password* Raspberry Pi
7. *Install VNC Server* pada Raspberry Pi melalui putty
8. Buka aplikasi *VNC Server*
9. Masukkan IP Raspberry Pi lalu masukkan *username* dan *password*
10. Raspberry Pi telah siap dioperasikan melalui laptop

2.2.5 Citra digital

Citra digital adalah sekumpulan piksel, yang disetiap piksel diilustrasikan sebagai satu kotak dan di tiap-tiap piksel memiliki koordinat posisi. Sebuah piksel memiliki koordinat berupa x y , x merupakan kolom, y merupakan baris dan piksel sudut kiri-atas memiliki koordinat $x=0$, $y=0$ sedangkan piksel sudut kanan-bawah memiliki koordinat $x=N-1$, $y=M-1$. Seluruh spektrum elektromagnetik, mulai dari gamma hingga gelombang radio akan menghasilkan gambar (Kadir, dkk., 2013). Pada penelitian ini, teknik pengolahan citra digital digunakan untuk mengolah hasil citra lidah. Dengan melakukan langkah-langkah pengolahan citra, sistem mampu mendiagnosis penyakit diabetes melitus melalui citra lidah.

Pengolahan citra ialah berbagai teknik yang digunakan untuk memanipulasi dan memodifikasi citra dengan berbagai cara. Pengertian pengolahan citra dalam arti yang lebih luas, yaitu: memperbaiki kualitas gambar seperti kontras citra ditingkatkan, merestorasi citra maupun mentransformasi warna sedangkan pengolahan citra dari aspek geometrik berupa rotasi, translasi. Pengolahan citra juga memiliki arti mengambil informasi dari objek atau deskripsi ciri suatu objek atau pengenalan objek yang terdapat pada citra, dan melakukan reduksi data yang bertujuan untuk penyimpanan data, pengiriman data dan waktu proses data (Hermawati, 2013). Terdapat teknik-teknik pengolahan citra digital yang terdiri dari tiga tingkat pengolahan, yaitu:

1. *Low-level Processing* teknik ini merupakan operasi yang dasar dalam pengolahan citra, proses yang dilakukan seperti pengurangan noise, memperbaiki, dan restorasi citra
2. *Mid-level Processing* merupakan operasi menengah pengolahan citra meliputi segmentasi, deskripsi objek, dan klasifikasi objek secara terpisah
3. *High-level Processing* merupakan operasi tertinggi pengolahan citra yang meliputi analisis citra.

2.2.6 Jenis citra

Suatu piksel memiliki rentang nilai minimal hingga rentang hingga rentang nilai maksimal. Rentang nilai tiap jenis warna citra memiliki rentang nilai berbeda-beda. Secara umum suatu piksel memiliki rentang nilai dari 0 sampai 255. Penggambaran citra seperti ini digolongkan menjadi citra integer. Berdasarkan warna-warna penyusunnya, citra digital dapat dibagi menjadi tiga macam, yaitu (Wijaya, 2007):

1. Citra Berwarna

Citra berwarna merupakan citra dengan nilai piksel merepresentasikan warna tertentu. Citra ini terbentuk dari kombinasi-kombinasi warna dasar yaitu warna merah, warna hijau, dan warna biru atau RGB (*Red, Green, Blue*). Setiap warna dasar menggunakan 8 bit dengan kisaran angka 0 hingga 255. Citra berwarna ditampilkan pada Gambar 2.7.



Gambar 2.7 Citra berwarna

Sumber: Siregar (2017)

2. Citra *Grayscale*

Citra *Grayscale* merupakan citra memiliki rentang nilai 0 hingga 255 yang merepresentasikan skala keabuan atau intensitas warna putih. Nilai intensitas paling rendah merepresentasikan warna hitam dan nilai intensitas paling tinggi direpresentasikan oleh warna putih. Citra *grayscale* memiliki kedalaman piksel 16 bit untuk penggunaan yang memerlukan ketelitian tinggi. Citra *grayscale* didapatkan dari konversi citra berwarna, secara matematis konversi citra berwarna ke citra *grayscale*, ditunjukkan pada Persamaan 2.1

$$\textit{Grayscale} = (R + G + B)/3 \quad (2.1)$$

Dengan R merupakan komponen nilai merah, G merupakan komponen nilai hijau, dan B merupakan komponen nilai biru. Hasil konversi citra berwarna menjadi citra *grayscale* ditunjukkan pada Gambar 2.8.



Gambar 2.8 Citra grayscale

Sumber: Siregar (2017)

3. Citra Biner

Citra biner merupakan citra yang hanya terdiri dari 2 warna yaitu hitam dan putih. Setiap piksel pada citra biner cukup direpresentasikan dengan 1 bit. Pada beberapa aplikasi, citra biner masih tetap dibutuhkan misalnya penerapan pada citra kode barang (barcode) yang tertera pada kemasan barang. Pada citra biner, nilai 1 merupakan nilai pada piksel objek dan nilai 0 merupakan nilai latar belakang. Citra biner diperoleh dari hasil konversi citra grayscale, secara matematis konversi citra grayscale ke citra biner, ditunjukkan pada Persamaan 2.2

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

$g(x,y)$: citra biner

$f(x,y)$: citra grayscale

T : threshold

Objek akan menjadi nilai 1 jika nilai grayscale lebih besar atau sama dengan threshold, selain itu akan bernilai 0.

Citra biner ditunjukkan pada Gambar 2.9.



Gambar 2.9 Citra biner

Sumber Siregar (2017)

Selain itu jenis-jenis citra lainnya dapat dilihat dari ekstensinya, seperti yang ditunjukkan pada Tabel 2.5

Tabel 2.5 Jenis-jenis ekstensi citra digital

Ekstensi	Deskripsi
JPG	JPG digunakan untuk mengurangi bagian-bagian dari gambar untuk memnahan piksel dalam gambar tersebut. JPG mendukung 24-bit RGB dan CMYK, dan 8-bit Grayscale
GIF	GIF ekstensi yang memiliki kedalaman warna 8-bit, yang dapat dipilih dari model RGB dan disimpan ke "Color Table". GIF menggunakan metode Lossless Compression, Lossless Compression merupakan teknik kompres yang dapat memperkecil ukuran file dengan tidak menurunkan mengurangi kualitas citra. Pada GIF terjadi penurunan jumlah warna menjadi 256, sehingga warna menjadi kurang realistis.
PNG	<i>Portable Network Graphics</i> (PNG) merupakan salah satu format yang merupakan gabungan dari format JPG dan GIF. PNG juga mendukung kelas 8-bit seperti GIF, sekaligus 24-bit seperti JPG dan mampu mengkompresi citra dari proses fotografi tanpa banyak mengurangi kualitas citra. PNG memiliki ukuran yang besar dibandingkan dengan GIF dan JPG.
BMP	Bitmap (BMP) merupakan representasi dari citra yang terdiri dari susunan titik/piksel yang tersimpan di memori komputer. BMP sangat bergantung pada resolusi, karena itu BMP akan memiliki ukuran yang besar jika resolusinya pun besar akan terjadi sebaliknya jika resolusi kecil maka ukuran file juga akan berukuran kecil.
TIFF	TIFF adalah jenis ekstensi citra terbaik dengan pengertian bahwa semua data dan informasi (data RGB, data CMYK) yang berkaitan dengan koreksi atau manipulasi terhadap gambar tersebut tidak hilang. Format TIFF memiliki kualitas gambar yang sangat baik sehingga biasa digunakan untuk pencetakan sejalan dengan itu ekstensi TIFF memiliki ukuran yang besar. Format ini memiliki kedalaman warna hingga 32 bit.

Dalam penelitian ini menggunakan ekstensi gambar tipe JPG, karena kemampuannya mengkompresi gambar sehingga memungkinkan sistem bekerja dengan cepat karena mengolah data yang tidak besar. JPG mendukung 24-bit RGB, CMYK dan 8-bit *Grayscale* kemampuan ini mendukung peneliti dalam mengolah citra lidah, karena dibutuhkan susunan warna pada citra untuk kemudian diambil tiap warna-warna pada lidah untuk diolah oleh sistem.

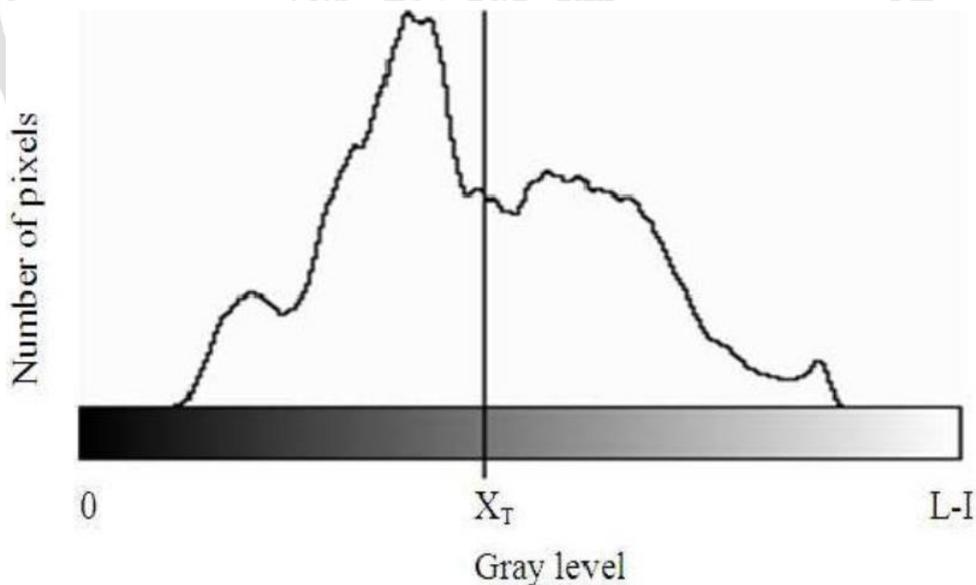
2.2.7 Pemerataan histogram (*Histogram Equalization*)

Pemerataan Histogram merupakan teknik pengolahan suatu citra untuk menormalisasi fitur visual citra digital, skala gray-level, dan kontras. Histogram menampilkan banyak piksel dalam suatu citra yang dikelompokkan berdasarkan level nilai intensitas piksel yang berbeda. Pada citra *grayscale* 8 bit, terdapat 256 level nilai intensitas yang berbeda maka pada histogram akan ditampilkan secara grafik distribusi dari masing-masing 256 level nilai piksel tersebut (Ahmad dan Hadinegoro, 2012).

Histogram equalization bekerja dengan cara meregangkan histogram, sehingga perbedaan tiap pikselnya menjadi lebih besar sehingga dapat menampilkan citra yang lebih tajam. Citra kontras ditentukan oleh rentang dinamis, yang didefinisikan sebagai perbandingan antara bagian paling terang dan paling gelap intensitas piksel. Histogram memberikan informasi untuk kontras dan intensitas keseluruhan distribusi dari suatu gambar. Misalkan gambar input $f(x, y)$ terdiri dari tingkat abu-abu diskrit dalam kisaran dinamis $[0, L-1]$ maka fungsi transformasi $C(r_k)$ dapat didefinisikan sebagai Persamaan 2.3

$$S_k = T(r_k) = (L - 1) \sum_{j=0}^k P_r(r_j) = \frac{L-1}{MN} \sum_{k=0}^k n_j \quad k = 0, 1, 2, \dots, L - 1 \quad (2.3)$$

Untuk persamaan transformasi histogram equalization pada gambar digital, variabel $M \times N$ menunjukkan total jumlah piksel, L jumlah tingkat abu-abu, dan $P_r(r_j)$ jumlah piksel dalam gambar masukan dengan intensitas nilai r_j . Rentang nilai input dan output abu-abu berada di kisaran $0, 1, 2, \dots, L-1$. Kemudian, transformasi histogram equalization memetakan input nilai r_k (di mana $k = 0, 1, 2, \dots, L-1$) hingga nilai output S_k . Dapat dilihat pada gambar 2.10



Gambar 2.10 Proses *Histogram Equalization*

Histogram equalization merupakan salah satu teknik dalam pengolahan gambar bekerja dengan meningkatkan kontras gambar secara umum. Teknik yang

digunakan pada metode ini menghasilkan intensitas dari citra dapat didistribusikan dengan lebih baik. Hal ini memungkinkan daerah kontras lokal yang lebih rendah untuk mendapatkan kontras yang lebih tinggi tanpa mempengaruhi kontras global.

2.2.8 Pengambangan (*Thresholding*)

Metode pengambangan (*Thresholding*) adalah cara yang umumnya digunakan dalam pengolahan citra digital untuk memisahkan objek dengan non objek berdasarkan pada nilai *threshold* T (Fauzi, 2010).

Proses pengambangan pada citra *grayscale* bertujuan untuk mendapatkan citra biner, secara matematis dapat ditulis seperti Persamaan 2.4:

$$g(x, y) = \{ 1 \text{ if } f(x, y) \geq T, 0 \text{ if } f(x, y) < T \} \quad (2.4)$$

Untuk menampilkan warna yang nilai intensitas pikselnya lebih kecil, digunakan Persamaan 2.5 sebagai berikut:

$$g(x, y) = \{ 1 \text{ if } f(x, y) \geq T, 0 \text{ if } f(x, y) < T \} \quad (2.5)$$

$g(x,y)$ merupakan citra biner dari citra *grayscale* $f(x,y)$ dan T menyatakan nilai *threshold*. Nilai T mempunyai peranan yang penting karena dapat menentukan kualitas citra biner yang baik (Kumaseh, 2013).

Terdapat dua jenis *thresholding*, yaitu (Putra, 2010):

1. Pengambangan global (*global thresholding*)

Pada pengambangan ini semua piksel ditransformasikan berdasarkan nilai ambang menjadi hitam atau putih, sehingga mengakibatkan banyak informasi yang hilang karena hanya menggunakan satu nilai ambang untuk keseluruhan piksel

2. Pengambangan local (*local thresholding*)

Pada pengambangan lokal, citra dikelompokkan menjadi bagian-bagian kecil lalu pengambangan dilakukan berdasarkan nilai ambang di tiap-tiap kelompok.

2.2.9 Otsu's Thresholding

Thresholding merupakan cara yang dilakukan untuk menyeleksi objek dengan yang bukan objek berdasarkan pemilihan nilai ambang yang memisahkan dua mode tersebut. Contohnya, citra yang terdiri atas dua objek terang diatas *background* gelap. Citra hasil *thresholding* didefinisikan sebagai: Piksel yang menjadi bagian dari objek dilabeli 1, dan piksel yang menjadi bagian dari *background* atau bukan objek yang diminati akan berlabel 0. *Threshold Optimal* ialah *threshold* yang menghasilkan rata-rata kesalahan segmentasi minimum. Salah satu metode yang digunakan untuk proses *thresholding* ialah metode Otsu.

Metode Otsu memaksimalkan tingkat kemiripan (*likelihood*) yang dapat memisahkan antara objek dengan *background*. Langkah-langkah metode Otsu adalah sebagai berikut (Gonzales, dkk., 2002):

- Nilai *Threshold* yang akan dicari dari suatu citra *grayscale* dinyatakan dengan k . Nilai k berkisar antara 1 sampai dengan L , dengan nilai $L = 255$. Probabilitas setiap pixel pada level ke i dapat dinyatakan:

$$P_i = n_i/N \tag{2.6}$$

Dengan : P_i = probabilitas piksel ke i
 n_i = jumlah piksel dengan nilai tingkat keabuan i
 N = total jumlah piksel pada citra

- Nilai momen kumulatif ke-nol (2.7), nilai momen kumulatif ke-satu (2.8) dan nilai rata-rata (2.9) berturut-turut adalah

$$\omega(k) = \sum_{i=1}^k P_i \tag{2.7}$$

$$\mu(k) = \sum_{i=1}^k i \cdot P_i \tag{2.8}$$

$$\mu_T = \sum_{i=1}^L i \cdot P_i \tag{2.9}$$

Dimana : $\omega(k)$ = momen kumulatif ke-nol
 $\mu(k)$ = momen kumulatif ke-satu
 μ_T = nilai rata-rata

- Nilai ambang k dapat ditentukan dengan memaksimumkan persamaan:

$$\sigma_B^2(K^*) = \max_{1 \leq K \leq L} \sigma_B^2(k) \tag{2.10}$$

dengan

$$\sigma_B^2(K^*) = \frac{[\mu_T \omega(k) - \mu_{(k)}^2]}{\omega(k)[1-\omega(k)]} \tag{2.11}$$

Keterangan:

$$\sigma_B^2 = \text{Nilai Ambang}$$

- Intensitas dibawah batas bernilai 0, sedangkan diatas batas bernilai 1

$$g(x,y) = \{ 1 \text{ if } f(x,y) \geq T, 0 \text{ if } f(x,y) < T \} \tag{2.12}$$

$g(x,y)$ adalah citra biner dari citra gray scale $f(x,y)$ dan T menyatakan nilai ambang.

2.2.10 OpenCV

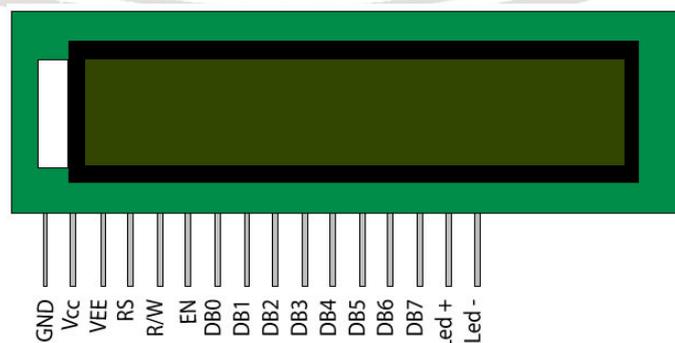
Open Source Computer Vision (OpenCV) Library merupakan sebuah *library* bertujuan untuk pengolahan citra dinamis secara *real-time*. OpenCV memberikan kebebasan sepenuhnya untuk dimanfaatkan secara komersil tanpa perlu mencantumkan kode sumbernya. OpenCV memiliki antarmuka yang

mencakup bahasa pemrograman C, Python, C++ dan Java dan OpenCV dapat digunakan pada sistem operasi Linux, Mac OS, Windows dan Android. Library ini dibuat sebagai optimasi realtime aplikasi, mempunyai API (*Application Programming Service*) untuk *high level* maupun *low level*, terdapat fungsi untuk *loading, saving, akuisisi citra dan video* (Arihutomo, 2010). Library OpenCV memiliki fitur-fitur sebagai berikut:

1. Memanipulasi data citra seperti alokasi memori, menyalin gambar dan konversi gambar.
2. *Image* atau *video I/O*
3. Manipulasi matriks dan vector serta terdapat juga *routines linear algebra (products, solvers, eigenvalues, SVD)*.
4. Pengolahan citra dasar (*filtering, edge detection, sampling, dan interpolasi, konversi warna, operasi morfologi, histogram, image pyramids*)
5. Analisis struktural
6. Kalibrasi kamera
7. Pendeteksi gerak
8. Pengenalan objek
9. Basic GUI (*menayangkan citra atau video, mouse atau keyboard, scrollbar*)
10. *Image labelling (line, Conic, polygon, text drawing)*

2.2.11 Liquid Crystal Display (LCD)

Liquid Crystal Display (LCD) merupakan modul display elektronik. Layar LCD 16x2 merupakan salah satu komponen dasar dan sangat umum digunakan di berbagai perangkat dan sirkuit. LCD 16x2 itu artinya karakter hingga 16 kolom dan 2 baris. Pada LCD ini setiap karakter ditampilkan dalam matriks 5x7 piksel. LCD ini memiliki dua register, yaitu, *Command* dan *Data*. Register *command* adalah instruksi yang diberikan ke LCD untuk melakukan tugas yang telah ditetapkan misalnya menginisialisasi, membersihkan layar, mengatur posisi kursor, dan mengontrol tampilan. Register data menyimpan data yang akan ditampilkan pada LCD. Data adalah nilai ASCII karakter yang ditampilkan pada LCD (Kushagra, 2018). Gambar LCD dapat dilihat pada Gambar 2.11.



Gambar 2.11 LCD

Sumber: 16x2-lcd-module-datasheet (2018)

Penjelasan gambar 2.11 dijelaskan pada Tabel 2.6 sebagai berikut:

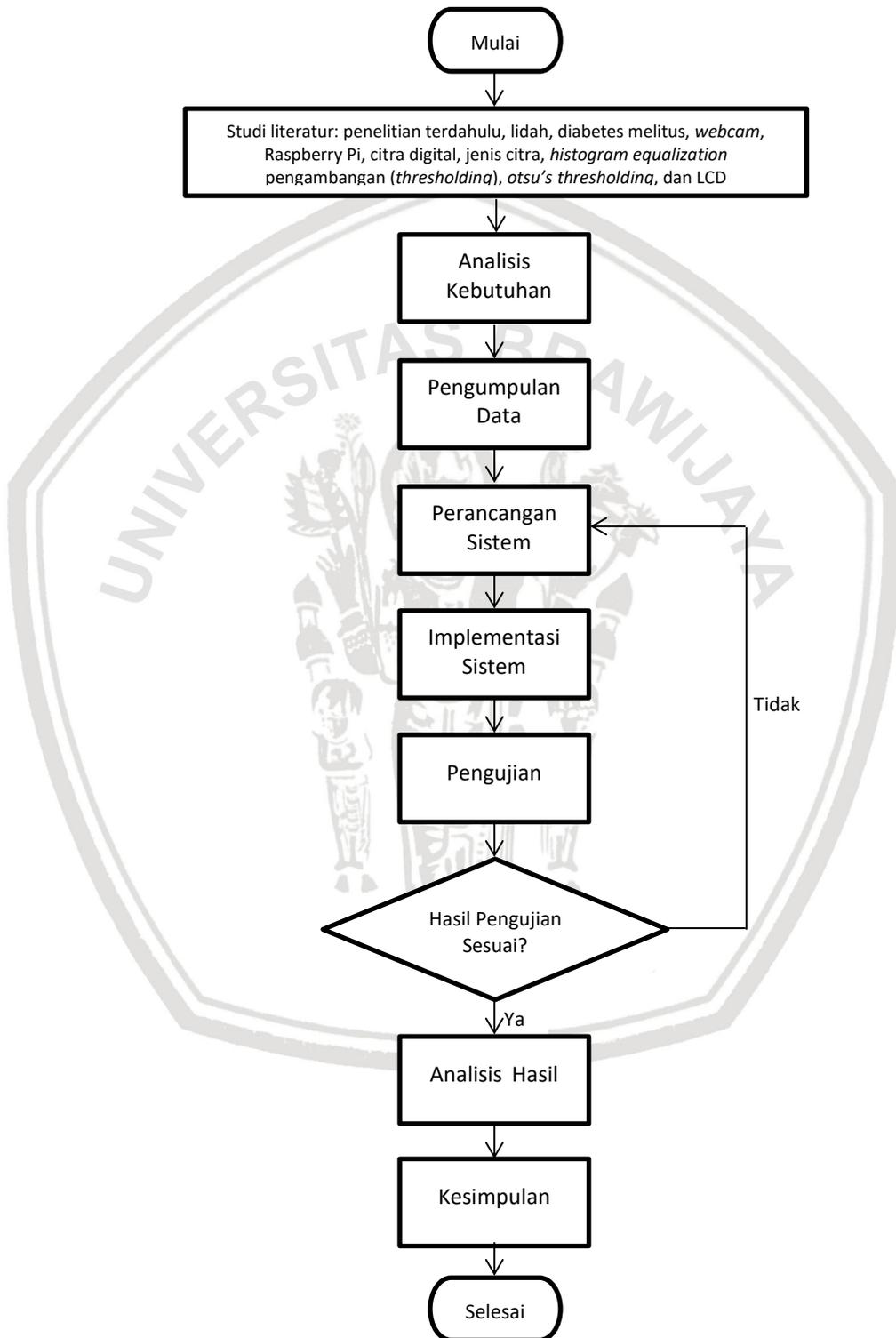
Tabel 2.6 Deskripsi Pin LCD

Nomor Pin	Simbol	Fungsi
1	Vss	<i>Ground</i>
2	Vdd	+3V atau +5V
3	V0	Mengatur Kontras LCD
4	RS	<i>H/L Register Select Signal</i>
5	RW	<i>H/L Read/Write Signal</i>
6	E	<i>H → L Enable Signal</i>
7	DB0	<i>H/L Data Bus Line</i>
8	DB1	<i>H/L Data Bus Line</i>
9	DB2	<i>H/L Data Bus Line</i>
10	DB3	<i>H/L Data Bus Line</i>
11	DB4	<i>H/L Data Bus Line</i>
12	DB5	<i>H/L Data Bus Line</i>
13	DB6	<i>H/L Data Bus Line</i>
14	DB7	<i>H/L Data Bus Line</i>
15	A/Vee	+4.2V untuk LED/Negative Voltage Output
16	K	Catu Daya untuk B/L (0V)

Sumber: Datasheet LCD (2018)

BAB 3 METODOLOGI

Pada bab ini menjelaskan alur metodologi penelitian dalam penyusunan skripsi. Metode penelitian yang digunakan pada skripsi ini dapat dilihat dari diagram alir pada Gambar 3.1.



Gambar 3.1 Diagram alir metodologi penelitian



3.1 Studi literatur

Pada perancangan dan implementasi dari penelitian ini dilakukan studi literatur untuk mendukung penelitian implementasi Raspberry Pi untuk mendiagnosis diabetes melitus melalui warna lidah menggunakan *Otsu's Thresholding*. Literatur ini bertujuan untuk teori penguat serta landasan dasar dalam penelitian. Teori pendukung bersumber dari buku, jurnal, *paper*, *website* resmi, artikel, dan buku elektronik. Beberapa literatur yang digunakan dalam penelitian ini, yaitu:

1. Lidah
Mempelajari tentang lidah, warna pada lidah, ciri-ciri lidah yang terindikasi menunjukkan suatu penyakit diabetes melitus
2. Diabetes melitus
Mempelajari pengertian diabetes melitus dan tanda-tanda klinis tubuh yang dapat menyebabkan timbulnya penyakit diabetes melitus
3. Webcam
Mempelajari tentang spesifikasi kamera dan cara konfigurasi kamera serta *coding* untuk dapat membaca nilai warna objek yang diteliti.
4. Raspberry Pi
Mempelajari penggunaan mini komputer Raspberry Pi sebagai pemroses utama dalam mengolah sistem baik dari segi konfigurasi pin secara perangkat keras maupun dari segi penggunaan algoritma dan logika yang tepat agar sesuai dengan hasil yang diharapkan
5. Citra digital
Mempelajari tentang citra digital, cara menjadikan citra digital sebagai masukan dari sistem, dan mengolah agar menjadi data sesuai dengan hasil yang diharapkan
6. Jenis Citra
Mempelajari tentang jenis-jenis citra digital, menentukan jenis citra digital yang tepat agar menjadi data sesuai dengan hasil yang diharapkan
7. Histogram Equalization
Mempelajari teknik histogram equalization sebagai pengolahan citra yang berfungsi untuk meningkatkan kualitas citra dengan cara meregangkan nilai histogram citra
8. Pengambangan (*thresholding*)
Mempelajari terkait pengertian pengambangan (*thresholding*), mengerti kelebihan serta kekurangan metode ini dan rumus yang digunakan dalam melakukan pengklasifikasian

9. Otsu's Tresholding

Mempelajari terkait pengertian *Otsu's thresholding*, mengerti kelebihan serta kekurangan metode ini dan rumus yang digunakan dalam melakukan pengklasifikasian

10. Library OpenCV

Mempelajari terkait library OpenCV yang akan digunakan sebagai bahasa pemrograman keseluruhan sistem

11. LCD (*Liquid Crystal Display*)

Mempelajari tentang spesifikasi LCD dan cara konfigurasi LCD serta *coding* untuk dapat menampilkan hasil sistem pada layer LCD.

3.2 Analisis kebutuhan

Analisis kebutuhan sistem berguna untuk menganalisa semua kebutuhan yang diperlukan oleh sistem yang akan dirancang. Analisis kebutuhan sistem dilakukan dengan mengidentifikasi kebutuhan sistem. Analisis kebutuhan tersebut, terdiri dari perangkat lunak dan perangkat keras. Perangkat lunak yang dibutuhkan dalam sistem ini digunakan untuk membuat program dan memproses hasil gambar yang telah diambil. Perangkat keras yang dibutuhkan untuk menyusun sistem ini memiliki spesifikasi sebagai perangkat yang dapat mengambil gambar, media penyimpanan Raspberry Pi dan tampilan hasil diagnosis. Kebutuhan fungsional dari sistem yang akan dirancang terdiri dari:

1. *Webcam* digunakan untuk mengambil citra lidah yang ingin dideteksi.
2. Data dari kamera yang berupa citra lidah diolah untuk menghasilkan diagnosis penyakit diabetes mellitus berdasarkan metode *Otsu's Thresholding* menggunakan *library OpenCV* pada Raspberry Pi.
3. Sistem dapat menampilkan hasil diagnosis penyakit diabetes melitus pada layer LCD.

3.2.1 Kebutuhan perangkat keras

Berikut beberapa kebutuhan perangkat keras yang digunakan untuk membuat sistem pada penelitian ini:

1. Mini Komputer Raspberry Pi
2. *Webcam*
3. *SD Card*
4. Kabel LAN
5. Kabel USB
6. Kabel *Jumper*
7. Laptop

3.2.2 Kebutuhan Perangkat Lunak

Pada kebutuhan perangkat lunak yang dibutuhkan oleh sistem pada penelitian ini ialah Angry IP untuk mengetahui IP dari Raspberry Pi, VNC Viewer untuk menghubungkan Raspberry Pi dengan Laptop, Python sebagai bahasa pemrograman disertai library OpenCV.

3.3 Pengumpulan data

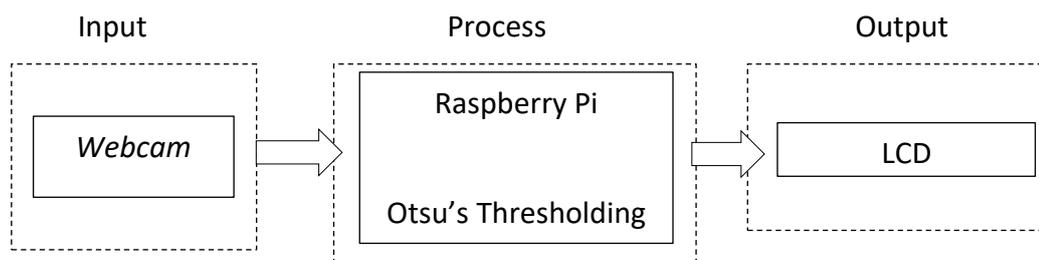
Proses akuisisi data yang dilakukan yaitu mengumpulkan data set berupa lidah pasien penderita diabetes melitus Rumah Sakit Lavalette. Waktu pelaksanaan untuk pengambilan data dilakukan mulai bulan juni hingga Juli 2018. Data yang diperoleh akan digunakan sebagai data latih untuk basis pengetahuan Sistem Diagnosis Penyakit Diabetes Melitus yang dikembangkan, data tersebut juga digunakan sebagai data uji untuk mengukur akurasi sistem. Pengambilan sampel dilakukan dengan cara judgement sampling. Judgement sampling merupakan suatu teknik pengambilan sampel berdasarkan beberapa karakteristik yang disesuaikan dengan maksud dan tujuan. Sebanyak 30 data citra lidah dikumpulkan dan dibagi menjadi 40 data latih dan 20 data uji. 40 data latih, terdiri dari 20 data latih lidah penderita diabetes dan 20 data latih lidah non penderita diabetes. Data uji sebanyak 20 citra, terbagi menjadi 10 data uji lidah penderita diabetes dan 10 data uji lidah non penderita diabetes. Pengambilan data citra lidah menggunakan *webcam* 2 MP dengan jarak 30 cm melalui pencahayaan yang cukup.

3.4 Perancangan sistem

Perancangan sistem dilakukan setelah semua kebutuhan sistem diperoleh melalui tahap analisis kebutuhan yang terdiri dari kebutuhan perangkat lunak maupun kebutuhan perangkat keras. Setelah kebutuhan sistem terpenuhi langkah selanjutnya ialah perancangan perangkat lunak dan perangkat keras.

3.4.1 Perancangan perangkat keras

Pada perancangan perangkat keras ini yaitu berupa blok diagram dari sistem yang akan dibuat. Terbagi menjadi 3 bagian yang terdiri dari pertama Input berupa *webcam* untuk mengambil citra lidah. Kedua *Process* terdapat Raspberry Pi yang berfungsi sebagai pengolah data dari *webcam* untuk dilakukan keputusan berdasarkan metode *Otsu's Thresholding*. Ketiga Output akan ditampilkan pada LCD. Blok diagram sistem ditunjukkan pada Gambar 3.2.

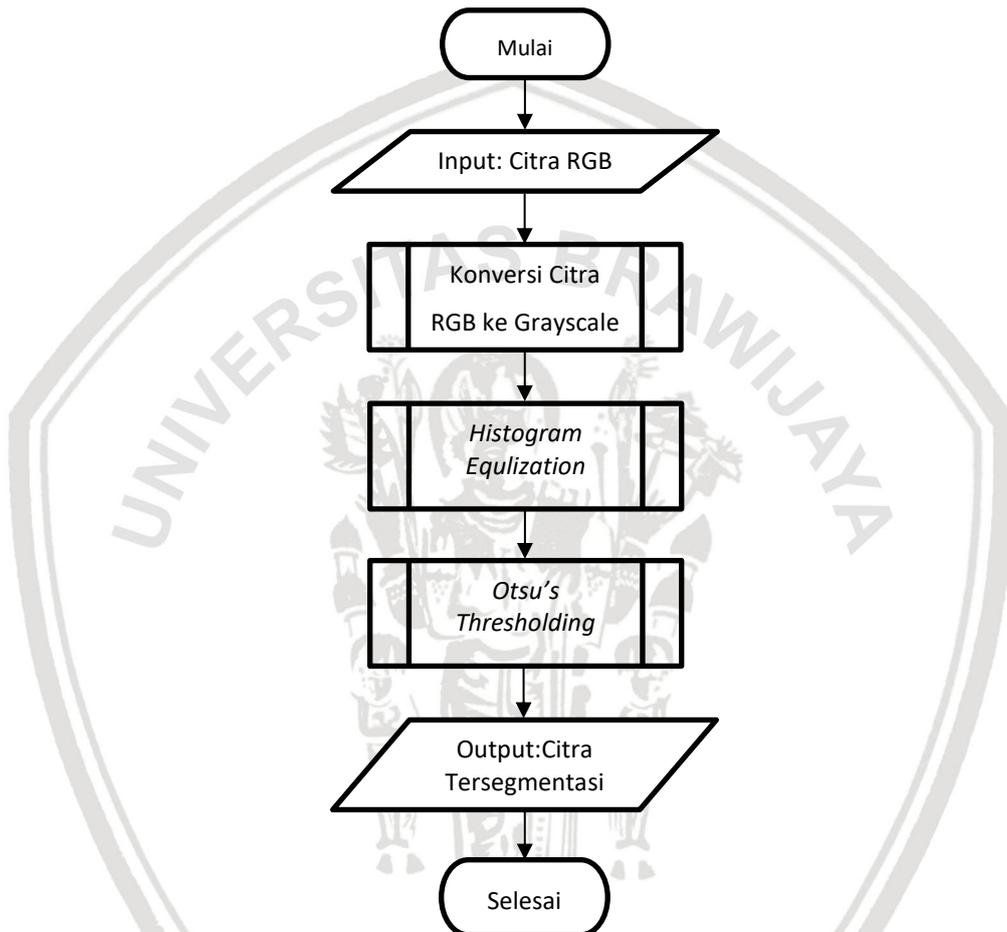


Gambar 3.2 Blok diagram sistem

Dari gambar bagan diatas dapat dilihat bahwa sistem saling terhubung. Raspberry Pi sebagai perangkat kontrol utama memiliki kendali terhadap sensor Webcam guna mengolah data yang didapat dari webcam. Saat tombol ditekan maka webcam akan mengambil gambar, gambar diteruskan ke Raspberry Pi diolah sehingga mengeluarkan output pada LCD.

3.4.2 Perancangan perangkat lunak

Pada perancangan perangkat keras ini yaitu berupa flowchart dari sistem yang akan dibuat. Flowchart cara kerja sistem ditunjukkan pada Gambar 3.3.



Gambar 3.3 Flowchart cara kerja sistem

Untuk proses pengolahan citra pada sistem ini hal pertama yang dilakukan ialah mengubah Citra RGB ke Grayscale. Histogram pada citra Grayscale diperbaiki menggunakan Histogram Equalization agar nilai histogram merata. Otsu's thresholding diterapkan untuk mencari ambang batas dari nilai warna lidah yang kemudian dilakukan segmentasi berdasarkan nilai ambang batas warna lidah. Keluaran berupa diabetes jika nilai ambang batas warna lidah termasuk ke dalam nilai warna lidah diabetes, selain kondisi tersebut keluaran menjadi tidak menderita diabetes.

3.5 Implementasi sistem

Pada tahap Implementasi dilakukan pembuatan perangkat keras dari implementasi Otsu's Tresholding untuk mendiagnosis penyakit diabetes melitus berdasarkan warna lidah. Tahap pembuatan perangkat keras diawali dengan perancangan dan implementasi Raspberry Pi dengan Webcam. Gambar hasil tangkapan webcam akan menjadi data masukan. Pada semua proses tersebut tentunya semua alat perangkat keras didefinisikan dan dikontrol melalui program yang ada pada Raspberry Pi dan proses implementasi Otsu's Tresholding untuk mendiagnosis penyakit diabetes melitus berdasarkan warna lidah pada Raspberry Pi menggunakan library OpenCV. Pada tahap implementasi ini, sisi perangkat lunak dan perangkat keras dibuat dan dikonfigurasi agar dapat membentuk sistem yang sinkron dengan perancangan perangkat keras.

3.6 Pengujian

Pengujian sistem pada penelitian ini dilakukan agar dapat mendiagnosis penyakit diabetes melitus berdasarkan warna lidah dengan hasil keakuratan yang tinggi. Pengujiannya meliputi:

1. Pengujian webcam apakah dapat bekerja sesuai dengan spesifikasinya
2. Pengujian akurasi metode Otsu's Tresholding dalam mensegmentasi warna lidah yang menderita penyakit diabetes melitus
3. Pengujian waktu komputasi pemrosesan sistem

Hasil pengujian kemudian dianalisa sehingga dapat mengetahui apakah sistem sudah sesuai dan tepat dengan kebutuhan yang telah ditetapkan yaitu untuk dapat menangkap objek, melakukan capture gambar dari objek tersebut, mengenali warna lidah penderita penyakit diabetes melitus. Dari hasil ini akan dijadikan pertimbangan untuk penarikan kesimpulan atau evaluasi untuk perbaikan selanjutnya.

3.7 Kesimpulan

Kesimpulan dari penelitian mengenai implementasi Otsu's Tresholding untuk mendiagnosis penyakit diabetes melitus berdasarkan warna lidah berbasis Raspberry Pi diperoleh setelah melakukan perancangan, pengimplementasian, pengujian dan dilanjutkan analisis sistem. Kesimpulan ini diambil berdasarkan hasil dari pengujian dan analisis penerapan dari metode dan algoritma yang digunakan terhadap perancangan. Isi dari kesimpulan diharapkan menjadi acuan pada penelitian lain untuk mengembangkan penelitian ini. Selain itu pada akhir penulisan terdapat saran yang berujuan untuk memperbaiki kesalahan-kesalahan yang terjadi dan menyempurnakan penulisan serta untuk memberikan pertimbangan sebagai pengembangan sistem selanjutnya.

BAB 4 REKAYASA KEBUTUHAN

4.1 Gambaran umum sistem

Implementasi Raspberry Pi untuk mendiagnosis penyakit diabetes melitus ini menggunakan warna lidah sebagai masukan sistem dan menerapkan metode Otsu's Tresholding untuk menyeleksi lidah yang terdiagnosis penyakit diabetes berdasarkan nilai batas. Sistem dapat memenuhi kebutuhan Pengguna apabila dapat menyeleksi mana lidah yang terdiagnosis penyakit diabetes dan mana yang tidak terdiagnosis penyakit diabetes. LCD digunakan untuk antarmuka Pengguna dengan sistem.

4.2 Identifikasi pengguna

Identifikasi Pengguna yang dapat menggunakan sistem ini ialah dokter. Identifikasi Pengguna bertujuan untuk menguji alat ini berdasarkan pengetahuan dokter dalam menggunakan alat ini. Orang yang ingin diperiksa harus menempatkan lidah pada tempat yang sudah disediakan, pengguna sistem ini dapat mengambil gambar pada kamera. Pengguna dapat melihat informasi gambar sedang diambil, gambar sedang memproses hingga hasil diagnosa lidah yang menderita penyakit diabetes melalui LCD.

4.3 Kebutuhan fungsional

Kebutuhan fungsional menjelaskan sistem berjalan sesuai yang dikehendaki, sehingga kebutuhan fungsional harus dapat terpenuhi. Kebutuhan yang harus dipenuhi oleh sistem ialah:

1. Sistem dapat mengambil citra lidah
Webcam bertugas mengakuisisi citra lidah. *Webcam* mengambil citra lidah dalam jenis warna RGB.
2. Sistem mampu mengkonversi warna citra lidah
Sistem diagnosis diabetes melalui warna lidah mampu mengkonversi warna citra RGB menjadi *grayscale*
3. Sistem dapat memperbaiki histogram dari citra lidah
Fungsi ini berguna untuk memperbaiki histogram dari citra yang diambil menggunakan *histogram equalization*
4. Sistem mendapatkan nilai batas citra data uji
Fungsi ini berguna untuk mendapatkan nilai batas dari data uji menggunakan *Otsu's Tresholding*. Nilai batas yang didapat digunakan untuk menyeleksi lidah yang terdiagnosis penyakit diabetes

5. Sistem dapat menyeleksi lidah yang terdiagnosis penyakit diabetes

Fungsi ini berguna untuk menyeleksi lidah berdasarkan proses *otsu's thresholding*. Data uji digunakan untuk mendapatkan rentang nilai batas, setelah itu binerisasi yaitu nilai batas yang berada di rentang nilai batas data uji akan bernilai 1 selain itu 0.

4.4 Kebutuhan non-fungsional

Kebutuhan non-fungsional pada sistem terbagi menjadi 2, kebutuhan perangkat keras dan kebutuhan perangkat lunak.

4.4.1 Kebutuhan perangkat keras

Dibutuhkan perangkat keras untuk membuat prototipe dari sistem ini, diantaranya adalah:

1. Raspberry Pi merupakan mini komputer yang berperan sebagai inti pemrosesan dari prototipe. Pin GPIO dan USB Raspberry Pi digunakan sebagai masukan dan keluaran komponen-komponen pendukung prototipe
2. *Webcam* digunakan untuk mengambil citra lidah. *Webcam* diletakkan pada bagian depan prototipe
3. SD Card

SD Card dibutuhkan sebagai media penyimpanan Raspberry Pi, selain itu SD Card digunakan sebagai disk lokasi inisialisasi *Operating System* Raspberry Pi

4. Kabel LAN

Kabel LAN dibutuhkan sebagai menghubungkan laptop dengan Raspberry Pi agar dapat berkomunikasi melalui jaringan.

5. Kabel USB

Kabel USB dibutuhkan untuk kabel catu daya Raspberry Pi yang dihubungkan ke Laptop.

6. LCD sebagai interface antara sistem dengan pengguna. LCD akan menampilkan perintah mengambil gambar, gambar sedang diproses dan hasil keluaran sistem

7. Laptop

Laptop berfungsi sebagai pengontrol Raspberry Pi melalui jaringan. Selain itu Laptop berfungsi sebagai sumber daya dari Raspberry Pi, spesifikasi laptop yang digunakan ialah:

-Model Perangkat : Lenovo Ideapad 300

-Prosesor : Intel Core i5-6200U CPU @ 2.3 GHz

-Sistem Operasi : Windows 10 Home

4.4.2 Kebutuhan perangkat lunak

Kebutuhan perangkat lunak dalam sistem ini terdiri dari:

1. Sistem Operasi Rasbian Jessie sebagai aplikasi yang membantu pemrograman pada Raspberry Pi
2. Angry IP dibutuhkan untuk memindai IP dari Raspberry Pi yang terhubung pada satu jaringan yang sama dengan laptop
3. VNC Viewer merupakan perangkat lunak yang dibutuhkan untuk *remote-control* yang memungkinkan laptop mengendalikan perangkat lain yang terkoneksi di satu jaringan
4. GPIO merupakan pin-pin Raspberry Pi sebagai *input* dan *output* dari komponen-komponen yang terhubung dengan system
5. Python merupakan bahasa pemrograman yang digunakan untuk *compile* program dan mengunggah program ke Raspberry Pi
6. OpenCV sebagai library Open Source yang memiliki fungsi-fungsi untuk melakukan pemrosesan citra digital.

4.5 Batasan sistem

Implementasi Raspberry Pi untuk mendiagnosis penyakit diabetes melalui warna lidah dapat dilakukan sesuai yang diinginkan, maka diperlukan Batasan-batasan implementasi sistem, yaitu:

1. Sistem melakukan segmentasi terhadap penyakit diabetes melitus
2. Sistem diagnosis penyakit diabetes melitus melalui warna lidah ini menggunakan Raspberry Pi sebagai otak dari keseluruhan system
3. Pendeteksi penyakit diabetes melalui warna lidah dibuat dalam bentuk *prototype*
4. Pengoperasian utama pada sistem ini ketika ada *trigger* dari pengguna
5. Sistem dapat bekerja optimal jika cahaya tiap pengambilan data selalu sama
6. Jarak pengambilan citra lidah menggunakan kamera berjarak 30 cm
7. Sistem memiliki layar LCD 16x2 sebagai tampilan antarmuka pengguna

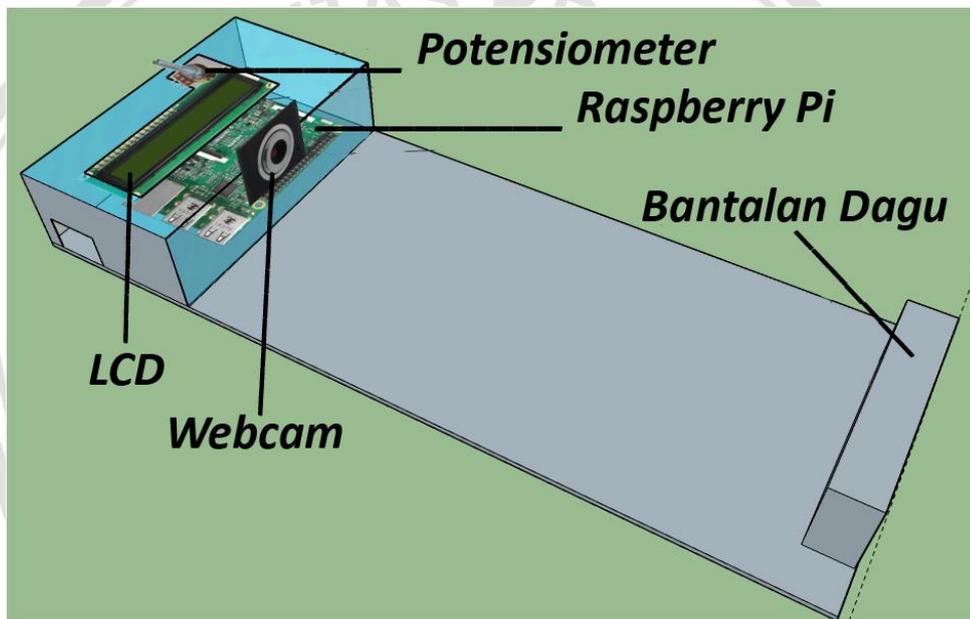
BAB 5 PERANCANGAN DAN IMPLEMENTASI

5.1 Perancangan sistem

Pada sub-bab ini, menjelaskan mengenai proses perancangan *prototype* alat, perancangan perangkat keras, perangkat lunak dan implementasinya.

5.1.1 Perancangan *prototype* alat

Perancangan *prototype* dari Sistem Diagnosis Penyakit Diabetes Melitus melalui Warna Lidah ini perlu diperhatikan peletakkan tiap-tiap komponen serta ukuran alat yang akan dikembangkan. Pembuatan desain sistem dirancang menggunakan aplikasi *SketchUp Pro 2018* untuk menggambarkan bentuk *prototype* yakni berupa persegi panjang berukuran panjang 30 cm, lebar 10 cm dan tinggi 10 cm. Bentuk *prototype* alat ditunjukkan pada Gambar 5.1 berikut.

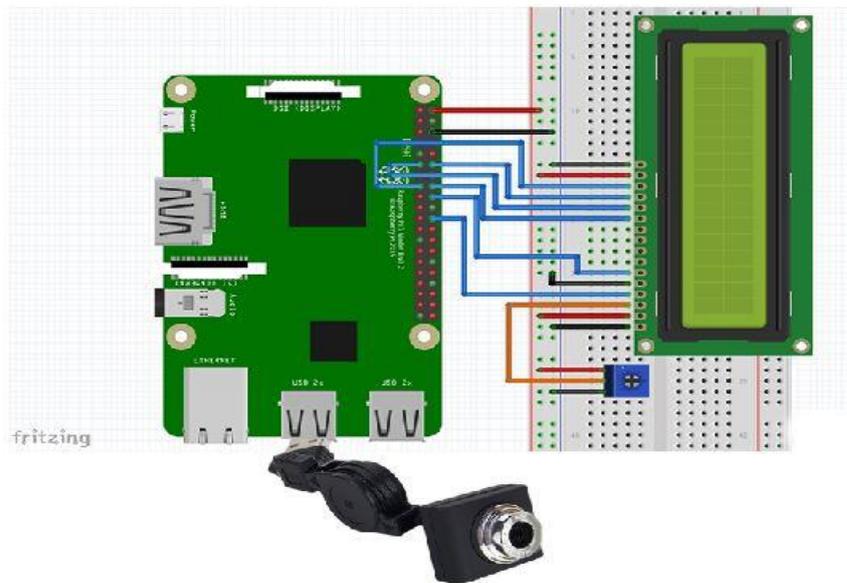


Gambar 5.1 Desain *prototype* alat

Pada Gambar 5.1 terlihat bahwa LCD diletakkan pada bagian atas *prototype* untuk memudahkan Pengguna agar mengerti proses-proses apa saja yang sedang berlangsung, peletakkan kamera bertepatan lurus dengan bantalan dagu agar gambar yang terambil tepat pada bagian lidah dari Pengguna. Raspberry Pi diletakkan dibagian dalam kotak.

5.1.2 Perancangan Perangkat Keras

Perancangan perangkat keras terdiri dari Raspberry Pi, USB Camera dan LCD yang akan diimplementasikan pada *prototype* system diagnosis penderita penyakit diabetes melitus. Skematik rangkaian sistem diagnosis penderita diabetes melalui warna lidah ditunjukkan pada Gambar 5.2



Gambar 5.2 Diagram skematik sistem diagnosis Diabetes Melitus

USB Camera sebagai pengambil gambar lidah Pengguna yang akan didiagnosis warna lidahnya dihubungkan pada Port USB Raspberry Pi. LCD digunakan untuk antarmuka antara Pengguna dengan system yang terhubung pada pin-pin GPIO Raspberry Pi. Tabel 5.1 dibawah ini merupakan penjelasan lebih rinci dari diagram skematik rangkaian perangkat keras pada Gambar 5.2:

Tabel 5.1 Kabel hubungan LCD dengan Raspberry Pi

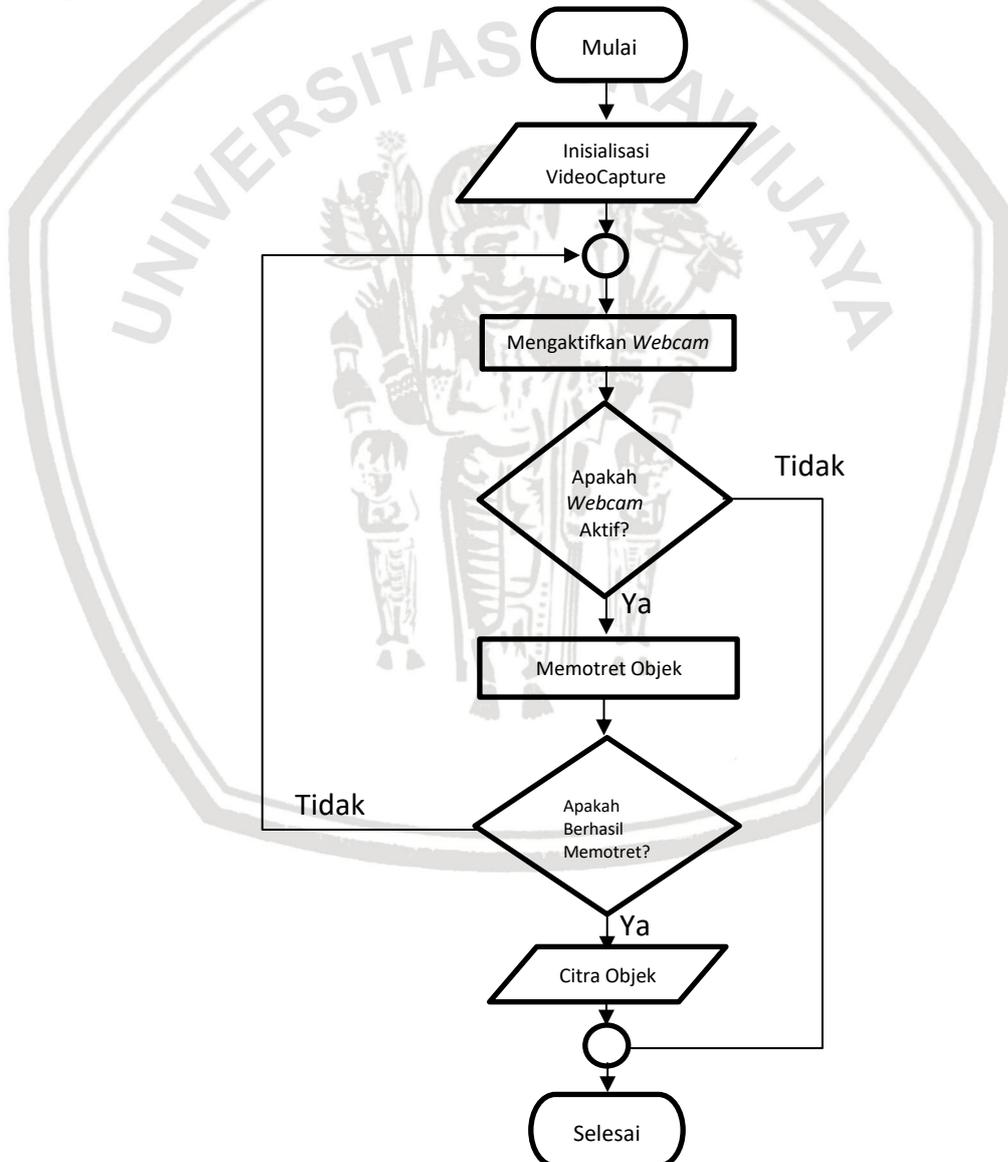
Pin LCD	Simbol LCD	Pin Raspberry Pi	Deskripsi
1	VSS	2	Ground
2	VDD	6	5V
3	V0	-	Pin 2 Potensiometer
4	RS	22	Register Select Signal
5	R/W	2	Read/Write Signal
6	E	18	Enable
7	DB0	-	-
8	DB1	-	-
9	DB2	-	-
10	DB3	-	-
11	DB4	16	Data Bus Line
12	DB5	11	Data Bus Line
13	DB6	12	Data Bus Line
14	DB7	15	Data Bus Line
15	A	6	5V untuk LED
16	K	2	Ground

5.1.3 Perancangan perangkat lunak

Pada sistem ini perancangan perangkat lunak terdiri dari 4 pembahasan yaitu perancangan akuisisi citra akan dilakukan pengakuasian citra lidah. Setelah akuisisi citra dilakukan proses perancangan konversi warna RGB menjadi *Grayscale* lalu dilanjutkan dengan perancangan histogram equalization. Setelah itu dilakukan proses perancangan Otsu's thresholding untuk memisahkan antara objek lidah dengan yang bukan lidah selain itu, Otsu's Thresholding diterapkan untuk mencari ambang batas dari nilai warna lidah, yang kemudian dilakukan segmentasi berdasarkan nilai ambang batas warna lidah.

5.1.3.1 Perancangan akuisisi citra

Pada source code akuisisi citra berfungsi untuk mengakuisasi citra lidah. Source code akan memanggil library webcam lalu webcam mulai mengambil gambar. Diagram alir akuisisi citra dapat dilihat pada Gambar 5.3



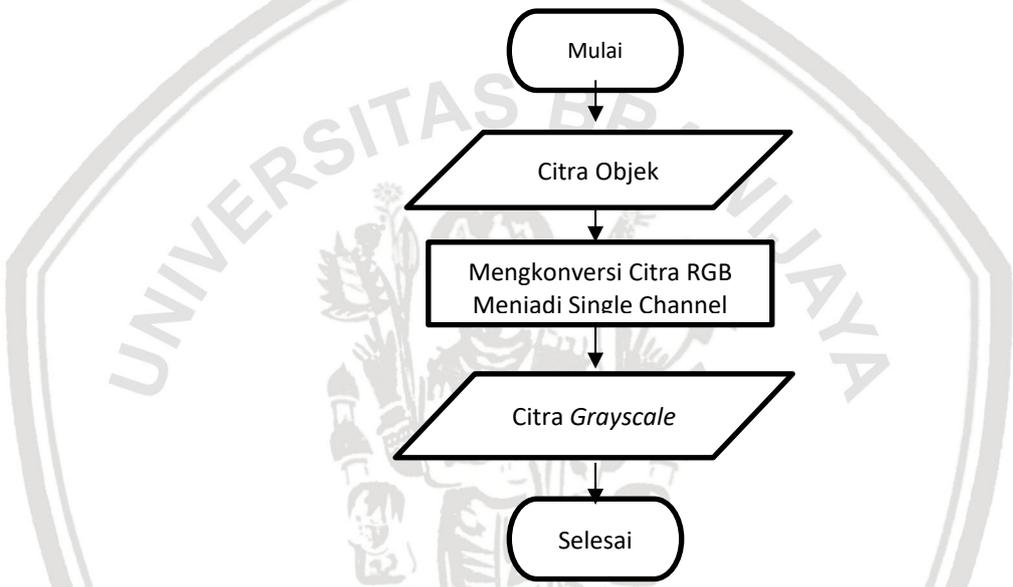
Gambar 5.3 Diagram alir akuisisi citra



Untuk proses akuisisi citra pada sistem ini hal pertama yang dilakukan menginisialisasi fungsi *VideoCapture*, lalu mengaktifkan *webcam*. Pengkondisian *webcam* aktif, jika ya maka *webcam* akan memotret objek, jika tidak maka selesai. Kemudian memotret objek, pengkondisian *webcam* berhasil memotret objek, jika iya maka citra objek terambil, jika tidak maka akan memeriksa keaktifan kamera. Hal ini terus dilakukan berulang hingga semua data terakuisisi. Hasil akuisisi citra semua data pada penelitian ini ditampilkan pada LAMPIRAN B.

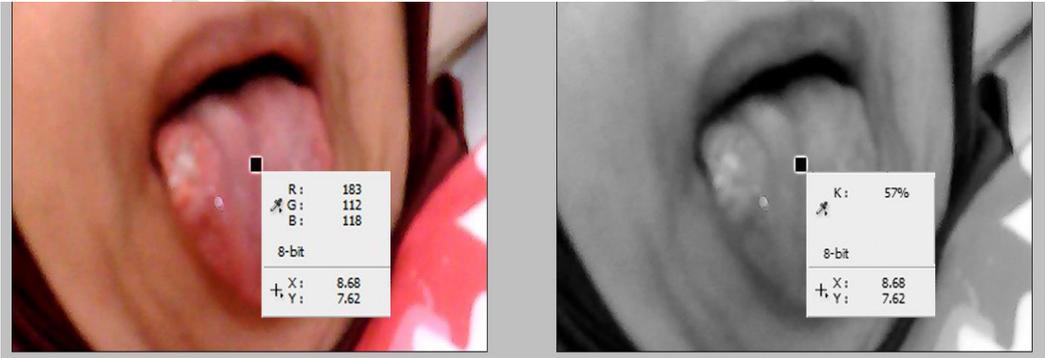
5.1.3.2 Perancangan konversi citra RGB ke *Grayscale*

Source code ini berfungsi untuk mengubah citra RGB yang telah diambil untuk diubah menjadi citra *grayscale*. Pada source code ini memanggil *library* open cv *rgb to grayscale*. Diagram alir konversi citra RGB ke Grayscale ditunjukkan pada Gambar 5.4.



Gambar 5.4 Diagram alir konversi citra RGB ke *Grayscale*

Untuk proses konversi RGB ke *grayscale* pada sistem ini hal pertama yang dilakukan menginput berupa citra objek, lalu mengkonversi citra RGB menjadi *single channel*. Keluaran dari proses ini berupa Citra *Grayscale*. Berikut merupakan contoh perhitungan konversi RGB ke *Grayscale* manual dengan salah satu data latih pada LAMPIRAN B, hasil konversi RGB ke *Grayscale* ditunjukkan pada Gambar



Gambar 5.5 Hasil konversi RGB ke *Grayscale*



Pada piksel dengan koordinat (8.68, 7.62), hasil konversi nilai *average* adalah:

$$R = 183$$

$$G = 112$$

$$B = 118$$

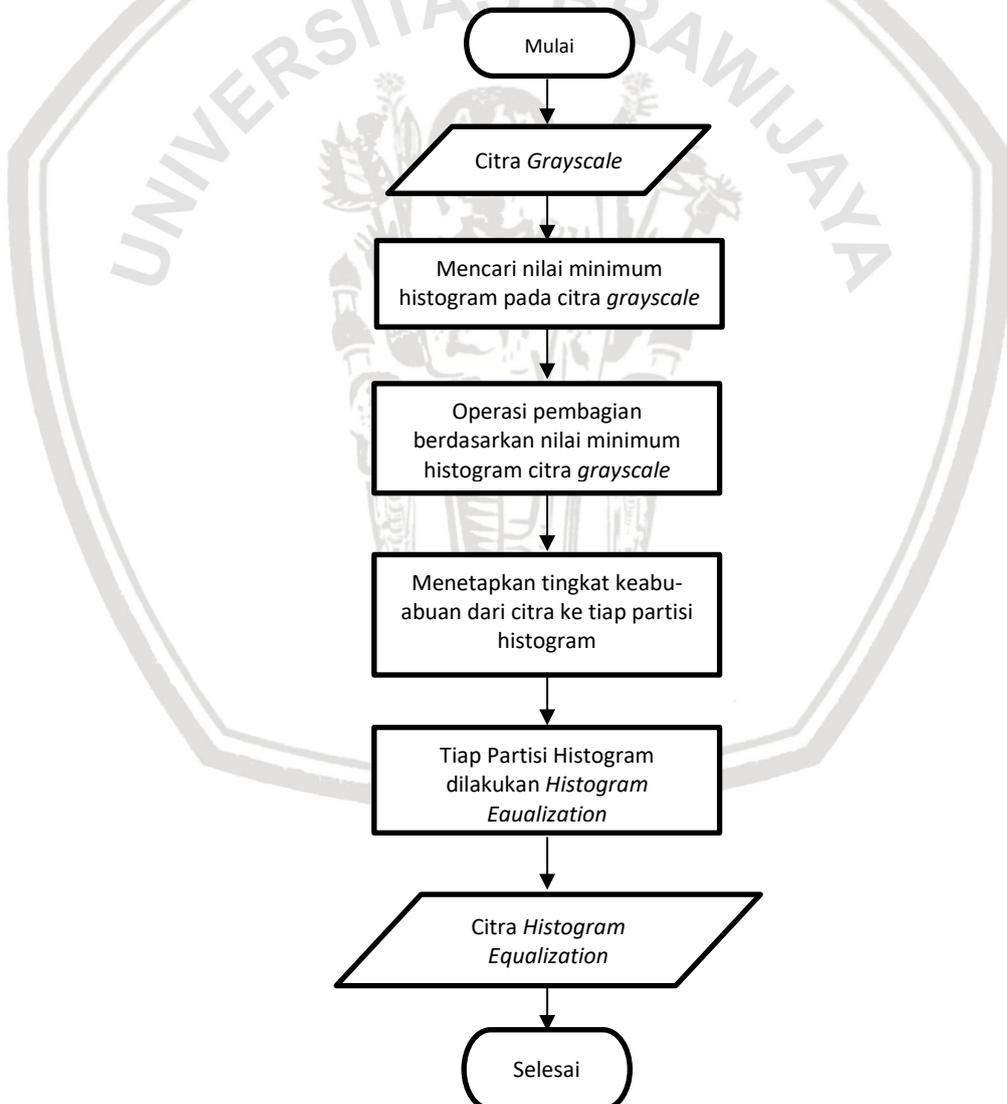
$$\text{Grayscale} = (R + G + B) / 3$$

$$\text{Grayscale} = (183 + 112 + 118) / 3$$

$$\text{Grayscale} = 137.667$$

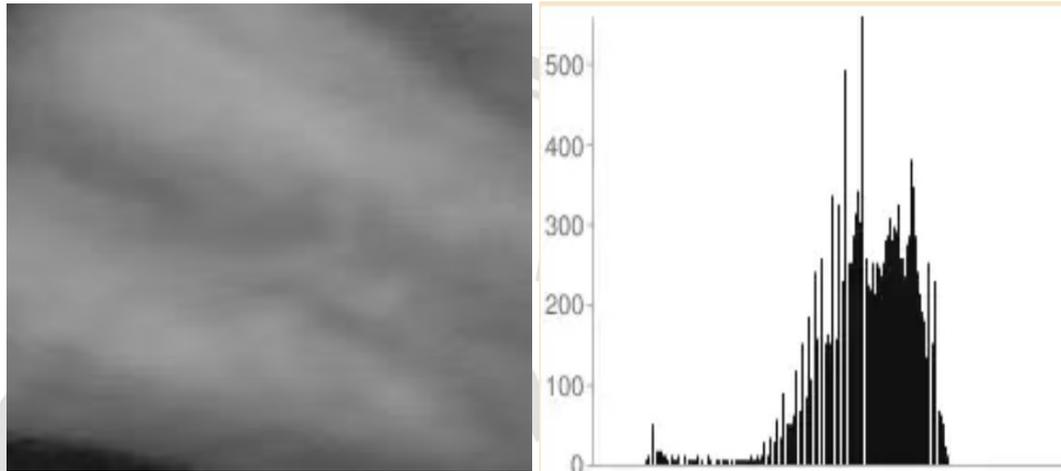
5.1.3.3 Perancangan *Histogram Equalization*

Histogram Equalization merupakan teknik memproses citra untuk menyesuaikan kontras citra dengan mengubah distribusi intensitas histogram. *Histogram Equalization* bertujuan untuk memberikan tren linier ke fungsi probabilitas kumulatif yang terkait dengan citra. Pengolahan pemerataan histogram berdasarkan pada penggunaan cumulative distribution function (CDF). Diagram alir *Histogram Equalization* ditunjukkan pada Gambar 5.6.



Gambar 5.6 Diagram alir *Histogram Equalization*

Untuk proses histogram equalization sistem ini hal pertama yang dilakukan menginput berupa citra objek grayscale, lalu menentukan nilai minimum histogram. Melakukan operasi pembagian berdasarkan nilai minimum histogram, setelah dibagi maka akan terdapat pada tingkatan-tingkatan keabuan yang berbeda-beda, tiap-tiap tingkatan keabuan dikelompokkan dalam satu partisi histogram. Tiap partisi histogram ini dilakukan operasi *Histogram Equalization* satu persatu. Berikut merupakan contoh perhitungan *histogram equalization* manual dengan salah satu data latih pada LAMPIRAN B, diketahui pembacaan histogram pada data latih ditunjukkan pada Gambar 5.7.



Gambar 5.7 Histogram data latih

Diketahui sebuah gambar dalam integer 558 nilai intensitas (0-255), dicuplik untuk dihitung histogram equalizationnya

Langkah 1. Menghitung probabilitas dari masing-masing nilai intensitas

Intensitas	Frekuensi	Probabilitas
0	0	0
1	0	0
2	0	0
3	0	0
4	0	0
5	0	0
6	0	0
7	0	0
8	0	0
9	0	0
10	0	0
11	0	0
12	0	0
13	0	0
14	0	0
15	0	0
16	0	0
17	0	0
18	0	0
19	0	0

Intensitas	Frekuensi	Probabilitas
20	0	0
21	0	0
22	0	0
23	0	0
24	6	0.00040404
25	10	0.000673401
26	0	0
27	49	0.003299663
28	0	0
29	16	0.001077441
30	14	0.000942761
31	15	0.001010101
32	11	0.000740741
33	9	0.000606061
34	6	0.00040404
35	0	0
36	11	0.000740741
37	8	0.000538721
38	5	0.0003367
39	11	0.000740741

Intensitas	Frekuensi	Probabilitas
40	0	0
41	2	0.00013468
42	13	0.000875421
43	0	0
44	5	0.0003367
45	3	0.00020202
46	3	0.00020202
47	3	0.00020202
48	9	0.000606061
49	0	0
50	8	0.000538721
51	0	0
52	0	0
53	9	0.000606061
54	6	0.00040404
55	2	0.00013468
56	0	0
57	6	0.00040404
58	8	0.000538721
59	0	0
60	5	0.0003367
61	5	0.0003367
62	5	0.0003367
63	0	0
64	4	0.00026936
65	1	6.73401E-05
66	5	0.0003367
67	6	0.00040404
68	4	0.00026936
69	4	0.00026936
70	7	0.00047138
71	4	0.00026936
72	3	0.00020202
73	10	0.000673401
74	6	0.00040404
75	5	0.0003367
76	10	0.000673401
77	8	0.000538721
78	12	0.000808081
79	29	0.001952862
80	0	0
81	9	0.000606061
82	32	0.002154882
83	0	0
84	27	0.001818182
85	58	0.003905724
86	0	0
87	32	0.002154882
88	91	0.006127946
89	0	0

Intensitas	Frekuensi	Probabilitas
90	53	0.003569024
91	53	0.003569024
92	51	0.003434343
93	61	0.004107744
94	119	0.008013468
95	0	0
96	66	0.004444444
97	152	0.01023569
98	0	0
99	85	0.005723906
100	182	0.012255892
101	106	0.007138047
102	0	0
103	243	0.016363636
104	154	0.01037037
105	0	0
106	258	0.017373737
107	0	0
108	149	0.01003367
109	162	0.010909091
110	151	0.01016835
111	337	0.022693603
112	0	0
113	159	0.010707071
114	324	0.021818182
115	0	0
116	227	0.015286195
117	491	0.033063973
118	0	0
119	250	0.016835017
120	252	0.016969697
121	284	0.019124579
122	312	0.021010101
123	341	0.022962963
124	303	0.02040404
125	558	0.037575758
126	0	0
127	255	0.017171717
128	224	0.015084175
129	218	0.014680135
130	250	0.016835017
131	215	0.014478114
132	252	0.016969697
133	244	0.016430976
134	232	0.015622896
135	250	0.016835017
136	278	0.018720539
137	286	0.019259259
138	305	0.020538721
139	281	0.018922559



Intensitas	Frekuensi	Probabilitas
140	296	0.01993266
141	292	0.0196633
142	324	0.021818182
143	255	0.017171717
144	256	0.017239057
145	236	0.015892256
146	273	0.018383838
147	285	0.019191919
148	381	0.025656566
149	346	0.023299663
150	285	0.019191919
151	243	0.016363636
152	215	0.014478114
153	190	0.012794613
154	181	0.012188552
155	136	0.009158249
156	254	0.017104377
157	0	0
158	150	0.01010101
159	231	0.015555556
160	0	0
161	69	0.004646465
162	64	0.004309764
163	51	0.003434343
164	25	0.001683502
165	9	0.000606061
166	0	0
167	0	0
168	0	0
169	0	0
170	0	0
171	0	0
172	0	0
173	0	0
174	0	0
175	0	0
176	0	0
177	0	0
178	0	0
179	0	0
180	0	0
181	0	0
182	0	0
183	0	0
184	0	0
185	0	0
186	0	0
187	0	0
188	0	0
189	0	0

Intensitas	Frekuensi	Probabilitas
191	0	0
192	0	0
193	0	0
194	0	0
195	0	0
196	0	0
197	0	0
198	0	0
199	0	0
200	0	0
201	0	0
202	0	0
203	0	0
204	0	0
205	0	0
206	0	0
207	0	0
208	0	0
209	0	0
210	0	0
211	0	0
212	0	0
213	0	0
214	0	0
215	0	0
216	0	0
217	0	0
218	0	0
219	0	0
220	0	0
221	0	0
222	0	0
223	0	0
224	0	0
225	0	0
226	0	0
227	0	0
228	0	0
229	0	0
230	0	0
231	0	0
232	0	0
233	0	0
234	0	0
235	0	0
236	0	0
237	0	0
238	0	0
239	0	0
240	0	0



Intensitas	Frekuensi	Probabilitas
241	0	0
242	0	0
243	0	0
244	0	0
245	0	0
246	0	0
247	0	0
248	0	0

Intensitas	Frekuensi	Probabilitas
249	0	0
250	0	0
251	0	0
252	0	0
253	0	0
254	0	0
255	0	0

Langkah 2. Menghitung kumulatif probabilitas dari masing-masing intensitas lalu Mengkalikan nilai kumulatif probabilitas dengan maksimum intensitas pada integer 558 (255)

Intensitas	Kumulatif Probabilitas	K.P x 255
0	0	0
1	0	0
2	0	0
3	0	0
4	0	0
5	0	0
6	0	0
7	0	0
8	0	0
9	0	0
10	0	0
11	0	0
12	0.00040404	0
13	0.001077441	0
14	0.001077441	0
15	0.004377104	0
16	0.004377104	0
17	0.005454545	0
18	0.006397306	0
19	0	0
20	0	0
21	0	0
22	0	0
23	0	0
24	0	0.103030303
25	0	0.274747475
26	0	0.274747475
27	0	1.116161616
28	0	1.116161616
29	0	1.390909091
30	0	1.631313131
31	0.00040404	1.888888889
32	0.001077441	2.077777778
33	0.008754209	2.232323232
34	0.009158249	2.335353535

Intensitas	Kumulatif Probabilitas	K.P x 255
35	0.009158249	2.335353535
36	0.00989899	2.524242424
37	0.01043771	2.661616162
38	0.010774411	2.747474747
39	0.011515152	2.936363636
40	0.011515152	2.936363636
41	0.011649832	2.970707071
42	0.012525253	3.193939394
43	0.012525253	3.193939394
44	0.012861953	3.27979798
45	0.013063973	3.331313131
46	0.013265993	3.382828283
47	0.013468013	3.434343434
48	0.014074074	3.588888889
49	0.014074074	3.588888889
50	0.014612795	3.726262626
51	0.014612795	3.726262626
52	0.014612795	3.726262626
53	0.015218855	3.880808081
54	0.015622896	3.983838384
55	0.015757576	4.018181818
56	0.015757576	4.018181818
57	0.016161616	4.121212121
58	0.016700337	4.258585859
59	0.016700337	4.258585859
60	0.017037037	4.344444444
61	0.017373737	4.43030303
62	0.017710438	4.516161616
63	0.017710438	4.516161616
64	0.017979798	4.584848485
65	0.018047138	4.602020202
66	0.018383838	4.687878788
67	0.018787879	4.790909091
68	0.019057239	4.85959596
69	0.019326599	4.928282828



Intensitas	Kumulatif Probabilitas	K.P x 255
70	0.01979798	5.048484848
71	0.02006734	5.117171717
72	0.02026936	5.168686869
73	0.020942761	5.34040404
74	0.021346801	5.443434343
75	0.021683502	5.529292929
76	0.022356902	5.701010101
77	0.022895623	5.838383838
78	0.023703704	6.044444444
79	0.025656566	6.542424242
80	0.025656566	6.542424242
81	0.026262626	6.696969697
82	0.028417508	7.246464646
83	0.028417508	7.246464646
84	0.03023569	7.71010101
85	0.034141414	8.706060606
86	0.034141414	8.706060606
87	0.036296296	9.255555556
88	0.042424242	10.81818182
89	0.042424242	10.81818182
90	0.045993266	11.72828283
91	0.04956229	12.63838384
92	0.052996633	13.51414141
93	0.057104377	14.56161616
94	0.065117845	16.60505051
95	0.065117845	16.60505051
96	0.06956229	17.73838384
97	0.07979798	20.34848485
98	0.07979798	20.34848485
99	0.085521886	21.80808081
100	0.097777778	24.93333333
101	0.104915825	26.75353535
102	0.104915825	26.75353535
103	0.121279461	30.92626263
104	0.131649832	33.57070707
105	0.131649832	33.57070707
106	0.149023569	38.0010101
107	0.149023569	38.0010101
108	0.159057239	40.55959596
109	0.16996633	43.34141414
110	0.18013468	45.93434343
111	0.202828283	51.72121212
112	0.202828283	51.72121212
113	0.213535354	54.45151515
114	0.235353535	60.01515152
115	0.235353535	60.01515152
116	0.250639731	63.91313131
117	0.283703704	72.34444444
118	0.283703704	72.34444444

Intensitas	Kumulatif Probabilitas	K.P x 255
119	0.300538721	76.63737374
120	0.317508418	80.96464646
121	0.336632997	85.84141414
122	0.357643098	91.1989899
123	0.380606061	97.05454545
124	0.401010101	102.2575758
125	0.438585859	111.8393939
126	0.438585859	111.8393939
127	0.455757576	116.2181818
128	0.470841751	120.0646465
129	0.485521886	123.8080808
130	0.502356902	128.1010101
131	0.516835017	131.7929293
132	0.533804714	136.120202
133	0.55023569	140.310101
134	0.565858586	144.2939394
135	0.582693603	148.5868687
136	0.601414141	153.3606061
137	0.620673401	158.2717172
138	0.641212121	163.5090909
139	0.66013468	168.3343434
140	0.68006734	173.4171717
141	0.69973064	178.4313131
142	0.721548822	183.9949495
143	0.738720539	188.3737374
144	0.755959596	192.769697
145	0.771851852	196.8222222
146	0.79023569	201.510101
147	0.809427609	206.4040404
148	0.835084175	212.9464646
149	0.858383838	218.8878788
150	0.877575758	223.7818182
151	0.893939394	227.9545455
152	0.908417508	231.6464646
153	0.921212121	234.9090909
154	0.933400673	238.0171717
155	0.942558923	240.3525253
156	0.9596633	244.7141414
157	0.9596633	244.7141414
158	0.96976431	247.289899
159	0.985319865	251.2565657
160	0.985319865	251.2565657
161	0.98996633	252.4414141
162	0.994276094	253.540404
163	0.997710438	254.4161616
164	0.999393939	254.8454545
165	1	255
166	1	255
167	1	255



Intensitas	Kumulatif Probabilitas	K.P x 255
168	1	255
169	1	255
170	1	255
171	1	255
172	1	255
173	1	255
174	1	255
175	1	255
176	1	255
177	1	255
178	1	255
179	1	255
180	1	255
181	1	255
182	1	255
183	1	255
184	1	255
185	1	255
186	1	255
187	1	255
188	1	255
189	1	255
190	1	255
191	1	255
192	1	255
193	1	255
194	1	255
195	1	255
196	1	255
197	1	255
198	1	255
199	1	255
200	1	255
201	1	255
202	1	255
203	1	255
204	1	255
205	1	255
206	1	255
207	1	255
208	1	255
209	1	255
210	1	255
211	1	255
169	1	255

Intensitas	Kumulatif Probabilitas	K.P x 255
212	1	255
213	1	255
214	1	255
215	1	255
216	1	255
217	1	255
218	1	255
219	1	255
220	1	255
221	1	255
222	1	255
223	1	255
224	1	255
225	1	255
226	1	255
227	1	255
228	1	255
229	1	255
230	1	255
231	1	255
232	1	255
233	1	255
234	1	255
235	1	255
236	1	255
237	1	255
238	1	255
239	1	255
240	1	255
241	1	255
242	1	255
243	1	255
244	1	255
245	1	255
246	1	255
247	1	255
248	1	255
249	1	255
250	1	255
251	1	255
252	1	255
253	1	255
254	1	255
255	1	255



Langkah 3. Melakukan pembulatan hasil perkalian menggunakan fungsi Floor

Intensitas	K.P x 255	Intensitas	K.P x 255	Intensitas	K.P x 255
0	0	49	3	98	20
1	0	50	3	99	21
2	0	51	3	100	24
3	0	52	3	101	26
4	0	53	3	102	26
5	0	54	3	103	30
6	0	55	4	104	33
7	0	56	4	105	33
8	0	57	4	106	38
9	0	58	4	107	38
10	0	59	4	108	40
11	0	60	4	109	43
12	0	61	4	110	45
13	0	62	4	111	51
14	0	63	4	112	51
15	0	64	4	113	54
16	0	65	4	114	60
17	0	66	4	115	60
18	0	67	4	116	63
19	0	68	4	117	72
20	0	69	4	118	72
21	0	70	5	119	76
22	0	71	5	120	80
23	0	72	5	121	85
24	0	73	5	122	91
25	0	74	5	123	97
26	0	75	5	124	102
27	1	76	5	125	111
28	1	77	5	126	111
29	1	78	6	127	116
30	1	79	6	128	120
31	1	80	6	129	123
32	2	81	6	130	128
33	2	82	7	131	131
34	2	83	7	132	136
35	2	84	7	133	140
36	2	85	8	134	144
37	2	86	8	135	148
38	2	87	9	136	153
39	2	88	10	137	158
40	2	89	10	138	163
41	2	90	11	139	168
42	3	91	12	140	173
43	3	92	13	141	178
44	3	93	14	142	183
45	3	94	16	143	188
46	3	95	16	144	192
47	3	96	17	145	196
48	3	97	20	146	201

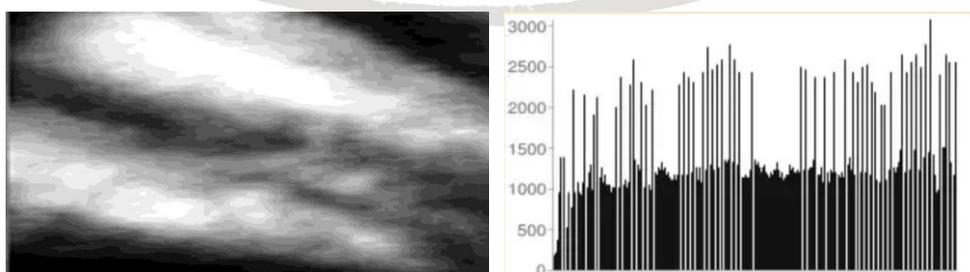


Intensitas	K.P x 255
147	206
148	212
149	218
150	223
151	227
152	231
153	234
154	238
155	240
156	244
157	244
158	247
159	251
160	251
161	252
162	253
163	254
164	254
165	255
166	255
167	255
168	255
169	255
170	255
171	255
172	255
173	255
174	255
175	255
176	255
177	255
178	255
179	255
180	255
181	255
182	255
183	255

Intensitas	K.P x 255
184	255
185	255
186	255
187	255
188	255
189	255
190	255
191	255
192	255
193	255
194	255
195	255
196	255
197	255
198	255
199	255
200	255
201	255
202	255
203	255
204	255
205	255
206	255
207	255
208	255
209	255
210	255
211	255
212	255
213	255
214	255
215	255
216	255
217	255
218	255
219	255
220	255

Intensitas	K.P x 255
221	255
222	255
223	255
224	255
225	255
226	255
227	255
228	255
229	255
230	255
231	255
232	255
233	255
234	255
235	255
236	255
237	255
238	255
239	255
240	255
241	255
242	255
243	255
244	255
245	255
246	255
247	255
248	255
249	255
250	255
251	255
252	255
253	255
254	255
255	255

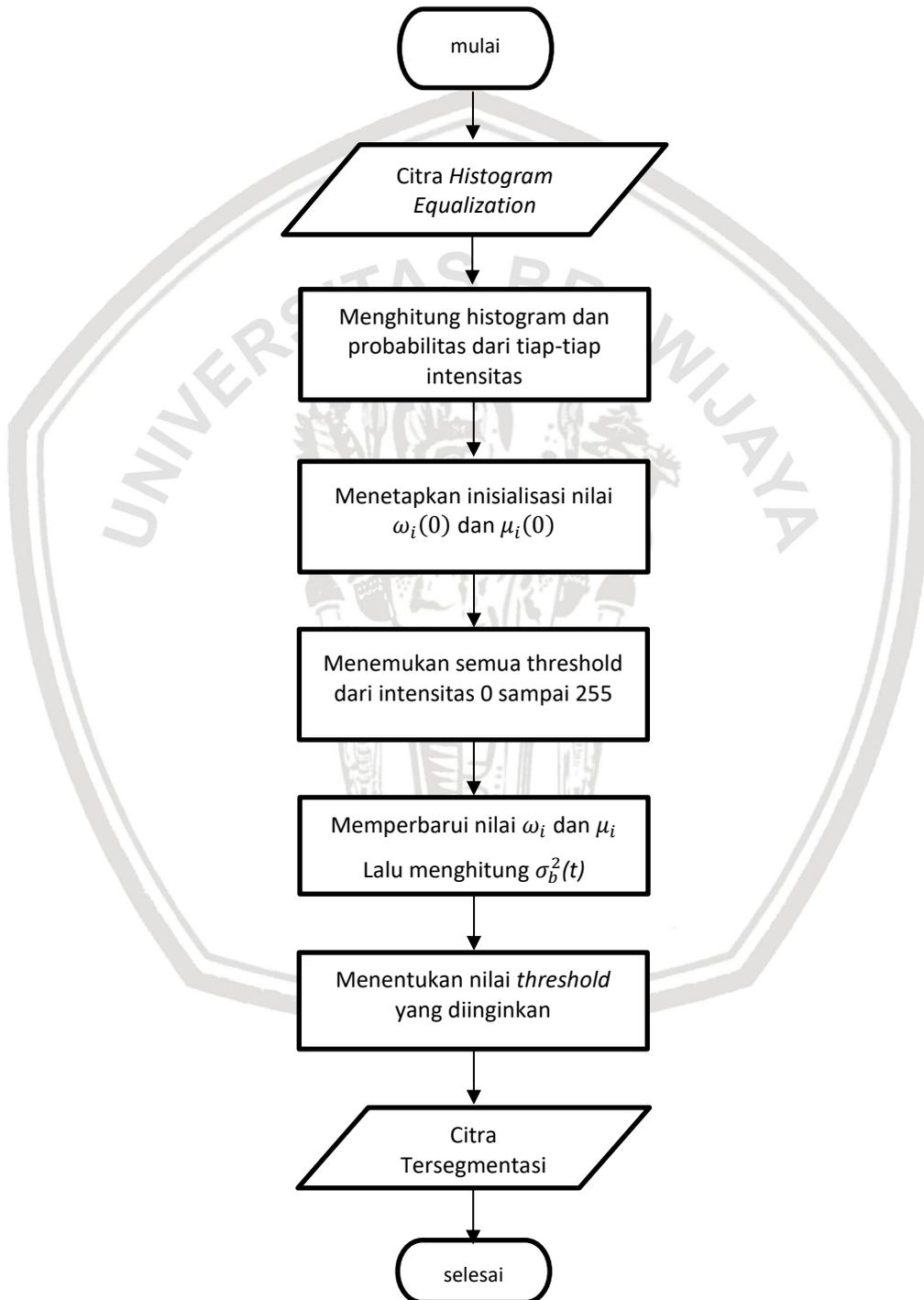
Sehingga hasil dari *Histogram Equalization* pada Gambar 5.7 histogram menjadi lebih rata dan gambar semakin tajam seperti yang ditunjukkan pada Gambar 5.8



Gambar 5.8 Hasil *Histogram Equalization* data latih

5.1.3.4 Perancangan *Otsu's Thresholding*

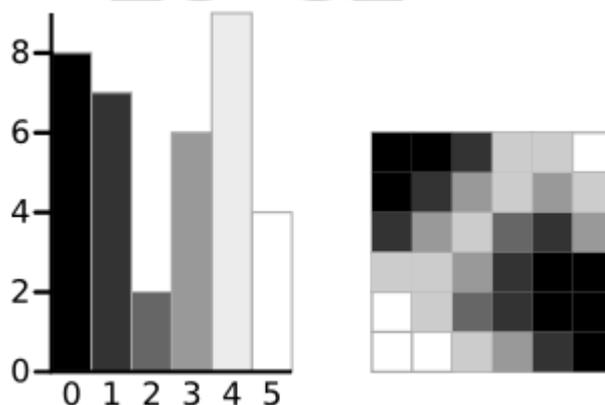
Metode Otsu bertujuan untuk membagi histogram citra *grayscale* menjadi dua daerah yang berbeda secara otomatis tanpa membutuhkan memasukkan nilai *threshold* dari pengguna. Metode otsu melakukan analisis diskriminan dengan menentukan suatu variable dengan membedakan antara dua atau lebih kelompok secara alami. Diagram alir source code *Otsu's Thresholding* ditunjukkan pada Gambar 5.9.



Gambar 5.9 Diagram alir *Otsu's Thresholding*

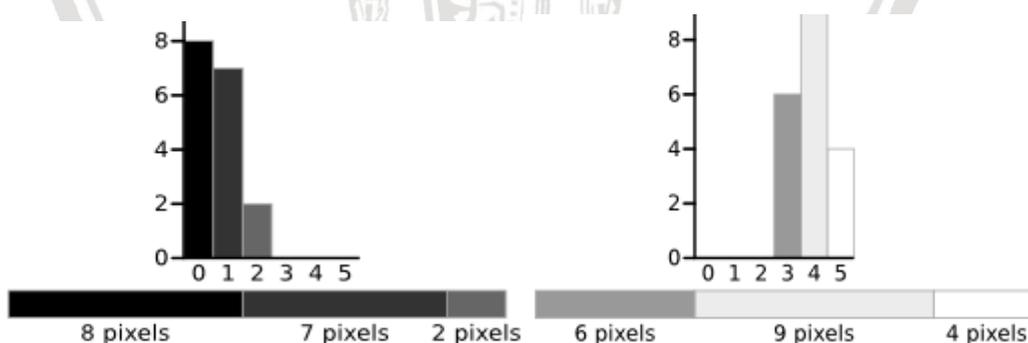
Untuk proses *Otsu's thresholding* pada sistem ini hal pertama yang dilakukan menginput berupa citra *histogram equalization*, lalu menghitung histogram dan probabilitas dari tiap-tiap intensitas. Dari hasil perhitungan histogram dan probabilitas tiap intensitas diketahui nilai $\omega_i(0)$ dan $\mu_i(0)$, dengan nilai ini digunakan untuk mencari nilai threshold dari semua intensitas 0 sampai 255. Memperbarui nilai $\omega_i(0)$ dan $\mu_i(0)$, menetapkan nilai threshold sesuai dengan yang diinginkan. Menggunakan perkondisian biner pada Persamaan 2.3 didapatkan keluaran berupa citra yang tersegmentasi.

Berikut merupakan contoh perhitungan *Otsu's thresholding* manual dengan salah satu data latih pada LAMPIRAN B, diketahui pembacaan histogram pada data latih ditunjukkan pada Gambar 5.10.



Gambar 5.10 Pembacaan histogram data latih

Perhitungan untuk mencari nilai Within-Class-Variance dan Between Class Variance dengan batas masing-masing intensitas. Pada histogram ini nilai Intensitas yang digunakan ialah 3. Dimulai dari perhitungan *background* histogram lalu *foreground* histogram seperti yang ditunjukkan pada Gambar 5.11



Gambar 5.11 Histogram *background* dan *foreground* data latih

Background

$$\text{Weight, } W_B = \frac{\text{sum(frekuensi setiap intensitas)}}{\text{total frekuensi}}$$

$$\text{Weight, } W_B = \frac{8 + 7 + 2}{36} = 0.4722$$

$$\text{Mean, } \mu_B = \frac{(0 \times 8) + (1 \times 7) + (2 \times 2)}{17} = 0.6471$$

$$\text{Variance, } \sigma_B^2 = \frac{((0 - 0.6471)^2 \times 8) + ((1 - 0.6471)^2 \times 7) + ((2 - 0.6471)^2 \times 2)}{17}$$

$$\text{Variance, } \sigma_B^2 = \frac{(0.4187 \times 8) + (0.1246 \times 7) + (1.8304 \times 2)}{17} = 0.4637$$

Kemudian perhitungan *foreground* histogram

Foreground

$$\text{Weight, } W_F = \frac{\text{sum(frekuensi setiap intensitas)}}{\text{total frekuensi}}$$

$$\text{Weight, } W_F = \frac{6 + 9 + 4}{36} = 0.5278$$

$$\text{Mean, } \mu_F = \frac{(3 \times 6) + (4 \times 9) + (5 \times 4)}{19} = 3.8947$$

$$\text{Variance, } \sigma_F^2 = \frac{((3 - 3.8947)^2 \times 6) + ((4 - 3.8947)^2 \times 9) + ((5 - 3.8947)^2 \times 4)}{19}$$

$$\text{Variance, } \sigma_F^2 = \frac{(4.8033 \times 6) + (0.0997 \times 9) + (4.8864 \times 4)}{19} = 0.5152$$

Didapatlah nilai *Weight Background*, *Mean Background*, *Variance Background*, *Weight Foreground*, *Mean Foreground*, dan *Variance Foreground*, nilai ini digunakan untuk menghitung:

$$\text{Within - Class Variance, } \sigma_{WC}^2 = (W_B \times \sigma_B^2) + (W_F \times \sigma_F^2)$$

$$\text{Within - Class Variance, } \sigma_{WC}^2 = (0.4722 \times 0.4637) + 0.5278 \times 0.5152$$

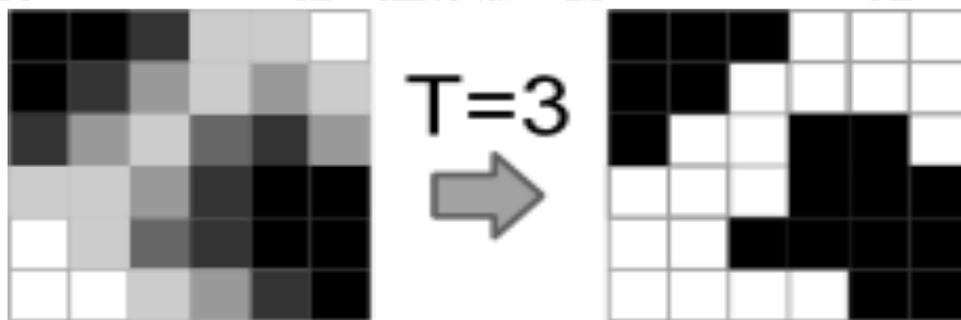
$$\text{Within - Class Variance, } \sigma_{WC}^2 = 0.4909$$



Tabel 5.2 Hasil perhitungan *Otsu's Thresholding* tiap intensitas

Threshold	T=0	T=1	T=2	T=3	T=4	T=5
Weight, Background	$W_b = 0$	$W_b = 0.222$	$W_b = 0.4167$	$W_b = 0.4722$	$W_b = 0.6389$	$W_b = 0.8889$
Mean, Background	$M_b = 0$	$M_b = 0$	$M_b = 0.4667$	$M_b = 0.6471$	$M_b = 1.2609$	$M_b = 2.0313$
Variance, Background	$\sigma_b^2 = 0$	$\sigma_b^2 = 0$	$\sigma_b^2 = 0.2489$	$\sigma_b^2 = 0.4637$	$\sigma_b^2 = 1.4102$	$\sigma_b^2 = 2.5303$
Weight, Foreground	$W_f = 1$	$W_f = 0.7778$	$W_f = 0.5833$	$W_f = 0.5278$	$W_f = 0.3611$	$W_f = 0.1111$
Mean, Foreground	$M_f = 2.3611$	$M_f = 3.0357$	$M_f = 3.7143$	$M_f = 3.8947$	$M_f = 4.3077$	$M_f = 5.000$
Variance, Foreground	$\sigma_f^2 = 3.1196$	$\sigma_f^2 = 1.9639$	$\sigma_f^2 = 0.7755$	$\sigma_f^2 = 0.5152$	$\sigma_f^2 = 0.2130$	$\sigma_f^2 = 0$
Within Class Variance	$\sigma_W^2 = 3.1196$	$\sigma_W^2 = 1.5268$	$\sigma_W^2 = 0.5561$	$\sigma_W^2 = 0.4909$	$\sigma_W^2 = 0.9779$	$\sigma_W^2 = 2.2491$

Dapat dilihat pada tabel 5.2 threshold yang bernilai 3, memiliki total berat varian yang terkecil nilainya. Semua piksel dengan nilai dibawah 3 akan menjadi background, selain itu piksel dengan nilai lebih dari atau sama dengan 3 akan menjadi foreground, seperti yang ditunjukkan pada Gambar 5.13.



Gambar 5.12 Hasil segmentasi metode *Otsu's Thresholding*

Seluruh data latih dicari nilai thresholdnya, setelah ketemu nilai threshold pada data latih yang memiliki nilai terkecil akan menjadi batas bawah dan nilai threshold terbesar akan menjadi batas atas. Jika nilai pembacaan nilai threshold pada data uji berada pada rentang batas nilai tersebut maka dinyatakan diabetes melitus, jika nilai threshold dari data uji memiliki nilai diluar batas tersebut maka dinyatakan non diabetes (Mirnasari, 2013).



5.2 Implementasi sistem

Pada sub bab implementasi sistem, dijelaskan mengenai penerapan perancangan perangkat keras dan perangkat lunak.

5.2.1 Implementasi perangkat keras

Pada implementasi perangkat keras akan dijelaskan proses merealisasikan hasil perancangan agar menjadi sebuah sistem yang pantas untuk digunakan. Sistem ini dilengkapi berbagai komponen dan memiliki fungsi yang berbeda-beda, LCD untuk menampilkan program yang sedang berlangsung hingga menampilkan hasil keluaran program, Web Camera digunakan untuk mengambil citra lidah, fungsi konversi RGB ke Grayscale berguna untuk mengubah citra lidah dengan susunan warna RGB menjadi keabu-abuan, histogram equalization memperbaiki histogram agar lebih merata dari citra *grayscale*, penerapan metode Otsu's Thresholding untuk memisahkan objek selain lidah dan mendapatkan nilai ambang batas warna citra lidah penderita diabetes, yang terakhir ialah Raspberry Pi berfungsi sebagai otak dari sistem yang gunanya untuk mengambil data, mengolah lalu menampilkan proses yang sedang berlangsung dan menampilkan hasil keluaran program.

Hasil Implementasi rangkaian perangkat komponen sistem penyakit diabetes melitus melalui warna lidah ditunjukkan pada Gambar 5.13



Gambar 5.13 Rangkaian perangkat komponen sistem

Pada Gambar 5.13 menunjukkan hasil implementasi webcam dengan Raspberry Pi dimana *webcam* yang telah dirangkai sesuai dengan perancangan diletakkan dibagian luar dari kotak begitu juga dengan LCD yang dihubungkan ke kaki-kaki GPIO Pin Raspberry Pi melalui kabel *jumper*.

5.2.2 Implementasi perangkat lunak

Pada implementasi perangkat lunak akan dijelaskan proses merealisasikan perancangan perangkat lunak.

5.2.2.1 Implementasi kode program inisialisasi *Library*

Implementasi awal program dimulai dari inisialisasi library agar program selaras dengan perangkat keras yang terhubung pada Raspberry Pi. Kode program inisialisasi library ditunjukkan pada Tabel 5.3.

Tabel 5.3 Kode program inisialisasi *Library*

Baris	Kode Program
1	import os
2	import sys
3	import subprocess
4	import cv2 as cv
5	import time
6	From datetime import datetime
7	import numpy as np
8	from matplotlib import pyplot as plt
9	Import Adafruit_CharLCD as LCD

Baris ke-1 merupakan *library os*, *library os* memungkinkan Bahasa pemrograman python dan Raspberry Pi dapat berinteraksi dengan sistem operasi. Baris ke-2 merupakan pemanggilan *library sys*, *library* ini menyediakan akses ke beberapa variabel yang digunakan dan berguna untuk menerjemahkan bahasa pemrograman ke sistem. Baris ke-3 *library subprocess*, *library* ini memungkinkan pengguna untuk dapat membuat proses baru selain itu, menghubungkan proses ke *input program*, *output program*, *error program* dan mendapatkan *return code*. Baris ke-4 memanggil *library OpenCV*, *library* ini berguna memanggil keseluruhan fungsi yang disediakan oleh *library OpenCV*. Baris ke-5 merupakan *library time*, *library* ini berguna untuk menampilkan lama durasi suatu proses dalam bentuk satuan waktu. Baris ke-6 *library datetime*, *library* ini berfungsi untuk menampilkan format tanggal, bulan dan tahun. Baris ke-7 *library numpy*, *library* ini memungkinkan program melakukan komputasi N-dimensi array, aljabar linear, dan fourier transformation. Baris ke-8 memanggil *library matplotlib*, memungkinkan program membuat plot, histogram, spektrum, dan *bar chart*. Baris ke-9 pemanggilan *library Adafruit_CharLCD*, *library* ini berguna untuk mengakses Adafruit karakter LCD dari Raspberry Pi.

5.2.2.2 Implementasi kode program akuisisi citra

Implementasi awal program dimulai dari akuisisi citra. Source Code akuisisi citra ini memanfaatkan *library openCV video capture*. Source code akuisisi citra lidah dapat dilihat pada Tabel 5.4.

Tabel 5.4 Kode program akuisisi citra

Baris	Kode Program
1	cam = cv.VideoCapture(0)
2	d=1
3	if ret:
4	ret, image = cam.read()
5	filename = '/home/pi/webcam/gambar_%d.jpg'%d
6	cv.imwrite(filename, image)
7	d+=1
8	cam.release()

Baris ke-1 menginisialisasi variabel *cam*, variabel ini berguna untuk memanggil *library* *VideoCapture* untuk mengaktifkan *webcam*. Baris ke-2 inialisasi *d=1*, variabel *d* ini berguna untuk kondisi perulangan yang nantinya sebagai penomoran citra yang telah diakuisisi. Baris ke-3 sampai baris ke-7 merupakan pengkondisian dalam akuisisi citra. Dimulai dari baris ke-3 merupakan pengkondisian *if*, kondisi *if* ini bergantung pada variabel *ret*. Baris ke-4 dilakukan pembacaan atau pemotretan citra, hasil pemotretan citra disimpan pada variable *ret* dan *image*. Baris ke-5 merupakan pemrograman untuk memberi nama file yang diakuisisi beserta letak direktori penyimpanan file tersebut. Baris ke-6 dilakukan proses penulisan data dengan ketentuan nama file dan direktori file didapat dari variable *filename* dan citranya didapat dari variable *image*. Baris ke-7 melakukan increment terhadap variabel *d*. Baris ke-8 menonaktifkan kamera jika pengkondisian *if ret* telah terpenuhi.

5.2.2.3 Implementasi kode program konversi RGB ke *Grayscale*

Pada kode program ini memanfaatkan *library* open CV *Grayscale*. Kode program Konversi Citra RGB ke Grayscale dapat dilihat pada Tabel 5.5.

Tabel 5.5 Kode program konversi citra RGB ke *Grayscale*

Baris	Kode Program
1	gray = cv.imread('/home/pi/sample/data/gambar_1.jpg', 0)
2	cv.imwrite ('/home/pi/sample/gray1.jpg', gray)

Baris ke-1 menginisialisasi variabel *gray*, variabel ini berguna untuk membaca gambar dengan memanfaatkan *flag* 0 yang artinya memuat citra berwarna menjadi citra abu-abu. Baris ke-2 dilakukan proses penyimpanan *file gray* yang diberi nama *gray_1.jpg* dan disimpan di direktori */home/pi/sample/gray*.

5.2.2.4 Implementasi kode program *Histogram Equalization*

Histogram Equalization merupakan metode yang meningkatkan kontras pada gambar, untuk meregangkan jangkauan intensitas. Kode program ini bekerja dengan cara pemetaan satu distribusi histogram ke distribusi nilai intensitas yang lebih luas dan lebih seragam sehingga nilai intensitas tersebar di seluruh rentang. Untuk mencapai pemerataan, pemetaan ulang harus menggunakan fungsi distribusi kumulatif. Kode program pemerataan histogram (*Histogram Equalization*) ditunjukkan pada Tabel 5.6.

Tabel 5.6 Kode program *Histogram Equalization*

Baris	Kode Program
1	<code>img = cv.imread('/home/pi/sample/data/abu/gray1.jpg', 0)</code>
2	<code>equ = cv.equalizeHist(img)</code>
3	<code>res = np.hstack((img,equ))</code>
4	<code>cv2.imwrite('/home/pi/sample/data/equ/equ1.jpg', res)</code>

Baris ke-1 menginisialisasi variabel `img`, variabel ini berguna untuk memuat citra `gray1.jpg`. Baris ke-2 menginisialisasi variabel `equ`, berfungsi untuk menyamaratakan histogram citra dari variabel `img`. Baris ke-3 menginisialisasi variabel `res`, berfungsi untuk menumpuk histogram dari sisi ke sisi untuk kemudian disusun menjadi suatu citra utuh dan citra baru. Baris ke-4 dilakukan proses penyimpanan file `res` yang diberi nama `equ1.jpg` dan disimpan di direktori `/home/pi/sample//data/equ`.

5.2.2.5 Implementasi kode program *Otsu Thresholding*

Otsu's Thresholding bekerja secara otomatis menghitung nilai ambang batas dari histogram citra bimodal. Nilai ambang batas yang digunakan ialah yang memiliki nilai *weighted within-class variance* yang minimum. Kode program *Otsu Thresholding* ditampilkan pada Tabel 5.7.

Tabel 5.7 Kode program *Otsu Thresholding*

Baris	Kode Program
1	<code>img = cv.imread('/home/pi/sample/data/equ/equ1.jpg',0)</code>
2	<code>blur = cv.GaussianBlur(img,(5,5),0)</code>
3	<code># find normalized_histogram, and its cumulative distribution function</code>
4	<code>hist = cv.calcHist([blur],[0],None,[256],[0,256])</code>
5	<code>hist_norm = hist.ravel()/hist.max()</code>
6	<code>Q = hist_norm.cumsum()</code>
7	<code>bins = np.arange(256)</code>
8	<code>fn_min = np.inf</code>
9	<code>thresh = -1</code>
10	<code>for i in xrange(1,256):</code>
11	<code> p1,p2 = np.hsplit(hist_norm,[i]) # probabilities</code>
12	<code> q1,q2 = Q[i],Q[255]-Q[i] # cum sum of classes</code>
13	<code> b1,b2 = np.hsplit(bins,[i]) # weights</code>
14	<code> # finding means and variances</code>
15	<code> m1,m2 = np.sum(p1*b1)/q1, np.sum(p2*b2)/q2</code>
16	<code> v1,v2 = np.sum(((b1-m1)**2)*p1)/q1,np.sum(((b2-m2)**2)*p2)/q2</code>
17	<code> # calculates the minimization function</code>
18	<code> fn = v1*q1 + v2*q2</code>
19	<code> if fn < fn_min:</code>
20	<code> fn_min = fn</code>
21	<code> thresh = i</code>
22	<code># find otsu's threshold value with OpenCV function</code>
23	<code>ret,otsu= cv.threshold(blur,0,255,cv.THRESH_BINARY+cv.THRESH_OTSU)</code>
24	<code>print("{} {}".format(thresh,ret))</code>
25	<code>if (thresh >= 127) and (thresh <= 134) :</code>
26	<code> print("diabetes")</code>
27	<code>else:</code>
28	<code> print("tidak menderita diabetes")</code>

Baris ke-1 menginisialisasi variabel `img`, variabel ini berguna untuk memuat citra `equ1.jpg`. Baris ke-2 menginisialisasi variabel `blur`, berfungsi untuk mengkaburkan citra berguna untuk mengurangi *noise*. Baris ke-3 sampai baris ke-9 merupakan proses perhitungan nilai normalisasi histogram dan fungsi distribusi kumulatif. Dimulai dari baris ke-4 menginisialisasi variable `hist`, variable ini berisi dari perhitungan *Histogram Calculation*, Citra masukkan *Histogram Calculation* didapat dari variable `blur`, `channel` yang digunakan `[0]`, `mask` bernilai "none" karena dalam program ini tidak diperlukan *masking*, ukuran array histogram sebesar 256 tiap dimensi, dengan rentan nilai array 0 sampai 256. Baris ke-5 menginisialisasi variable `hist_norm`, perhitungan normalisasi variable dengan cara membagi nilai `hist.ravel` dengan nilai maksimum histogram. Baris ke-6 menginisialisasi variabel `Q`, `Q` merupakan nilai kembalian dari jumlah kumulatif dari sepanjang elemen yang diberikan di tiap-tiap sumbu. Baris ke-7 inisialisasi variabel `bins`, `bins` merupakan total nilai intensitas yaitu sebesar 256. Baris ke-8 inisialisasi variabel `fn_min` bertipe data standar IEEE 754 floating point. Baris ke-9 menginisialisasi variable `thresh`, dimulai dari -1 karena ukuran dimensi dari citra dimulai dari 0.

Baris ke-10 merupakan perulangan perhitungan untuk masing-masing intensitas, mulai dari perhitungan intensitas 1 sampai intensitas 256. Baris ke-11 menghitung probabilitas pada indeks ke-*i*, dengan memisahkan array pada variable `hist_norm` menjadi sub-arrays dengan ukuran yang sama. Baris ke-12 menghitung jumlah kumulatif probabilitas di tiap-tiap kelas pada indeks ke-*i*. Baris ke-13 menghitung *weight* pada indeks ke-*i*, dengan memisahkan array pada variable `bins` menjadi sub-arrays dengan ukuran yang sama. Baris ke-14 sampai baris ke-16 menghitung nilai means dan variansi. Baris ke 17 sampai baris ke-21 menentukan nilai Within – Class Variance paling minimum. Baris ke-22 dan baris ke-23, menampilkan pada threshold ke-berapa yang memiliki Within – Class Variance. Baris ke-24 sampai baris ke-27 merupakan pengkondisian jika nilai `thresh` berada pada rentan 127 sampai 134 maka didiagnosis diabetes oleh sistem, selain itu didiagnosis tidak menderita diabetes.

BAB 6 PENGUJIAN DAN ANALISIS

Pada bab ini menjelaskan tentang pengujian yang dilakukan pada penelitian ini, dimulai dari prosedur serta proses pengujian yang akan menghasilkan data yang akan dianalisis dan mengetahui sistem yang dirancang apakah telah sesuai dengan analisis kebutuhan yang diinginkan, pengujian dilakukan dengan tahapan-tahapan yang telah ditentukan, hasil yang diperoleh dari pengujian dianalisis agar dapat ditarik kesimpulan dari penelitian yang telah dilakukan. Pengujian pada bab ini meliputi pengujian fungsional system, proses pengambilan gambar dan pengujian pembacaan warna lidah.

6.1 Pengujian fungsionalitas *Webcam*

Pengujian fungsionalitas *webcam* untuk mengetahui kamera dapat bekerja dengan baik dalam pengambilan gambar lidah.

6.1.1 Tujuan pengujian

Pengujian ini bertujuan untuk mengetahui apakah fungsionalitas *webcam* sudah berjalan sesuai dengan keinginan peneliti.

6.1.2 Prosedur pengujian

1. Menghubungkan Raspberry Pi dengan sumber daya agar menyala;
2. Menghubungkan Raspberry Pi dengan VNC Viewer untuk menjalankan program; dan
3. Menjalankan perintah akuisisi citra.

6.1.3 Hasil dan analisis pengujian

Hasil pengujian pada fungsionalitas *webcam* yaitu mengambil gambar 10 lidah orang yang berbeda, untuk melakukan pengujian ini, *webcam* akan mengambil gambar. Hasil akuisisi citra dapat dilihat pada Tabel 6.1.

Tabel 6.1 Hasil pengujian fungsionalitas *webcam*

Gambar ke-	Hasil	Hasil Pengamatan dari Serial Monitor
1	Berhasil	Gambar_1 telah terambil Waktu Komputasi Pengambilan Gambar_1 : 0:00:00.851302
2	Berhasil	Gambar_2 telah terambil Waktu Komputasi Pengambilan Gambar_2 : 0:00:00.993238
3	Berhasil	Gambar_3 telah terambil Waktu Komputasi Pengambilan Gambar_3 : 0:00:00.995143
4	Berhasil	Gambar_4 telah terambil Waktu Komputasi Pengambilan Gambar_4 : 0:00:00.844539
5	Berhasil	Gambar_5 telah terambil Waktu Komputasi Pengambilan Gambar_5 : 0:00:00.973503
6	Berhasil	Gambar_6 telah terambil Waktu Komputasi Pengambilan Gambar_6 : 0:00:00.846191
7	Berhasil	Gambar_7 telah terambil Waktu Komputasi Pengambilan Gambar_7 : 0:00:01.000632
8	Berhasil	Gambar_8 telah terambil Waktu Komputasi Pengambilan Gambar_8 : 0:00:00.995441
9	Berhasil	Gambar_9 telah terambil Waktu Komputasi Pengambilan Gambar_9 : 0:00:00.994068
10	Berhasil	Gambar_10 telah terambil Waktu Komputasi Pengambilan Gambar_10 : 0:00:00.877662

Hasil Pengujian 10 pengambilan gambar ditampilkan pada Gambar 6.1



Gambar 6.1 Hasil akuisisi citra

Hasil pengujian akuisisi 10 citra yang telah dilakukan dapat dihasilkan persentase nilai keberhasilan dengan menggunakan rumus.

Hasil perhitungan sebagai berikut:

$$\text{Nilai Persentase Pengambilan Gambar: } \frac{10}{10} \times 100\% = 100\%$$

Pengujian fungsionalitas *webcam* dapat disimpulkan berhasil karena memiliki persentase sebesar 100%.

6.2 Pengujian diagnosis penyakit Diabetes Melitus

Pengujian Diagnosis Penyakit Diabetes Melitus dibutuhkan untuk mengetahui tingkat keberhasilan sistem dalam mendiagnosis penyakit diabetes melitus. Tahap ini terdapat 2 kondisi, kondisi pertama yaitu 10 lidah orang penderita diabetes dan kondisi kedua 10 lidah orang yang tidak menderita diabetes. Kedua kondisi ini akan diambil gambar lidahnya masing-masing sebanyak 10 orang, lalu sistem akan mengolah.

6.2.1 Tujuan pengujian

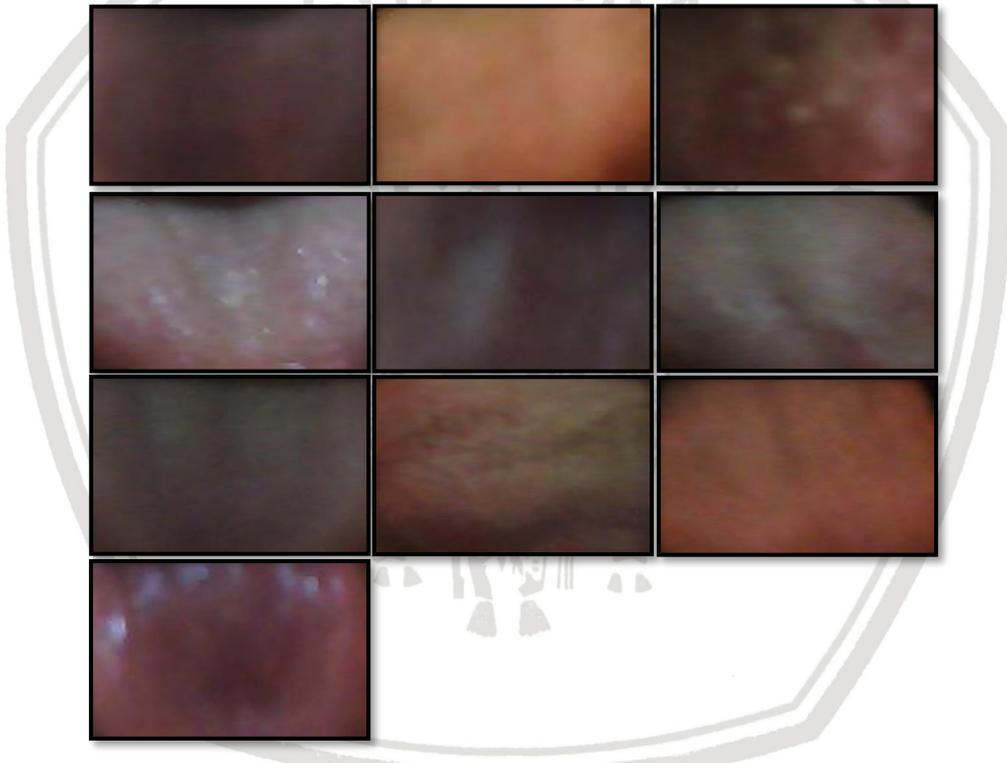
Pengujian ini bertujuan untuk mengetahui apakah sistem mampu membaca atau mendiagnosis penyakit diabetes melalui gambar lidah yang telah diambil secara benar dan akurat.

6.2.2 Prosedur pengujian

1. Menghubungkan Raspberry Pi dengan sumber daya agar menyala;
2. Menghubungkan Raspberry Pi dengan VNC Viewer untuk menjalankan program;
3. Menjalankan perintah konversi citra RGB ke *Grayscale*;
4. Menjalankan perintah *Histogram Equalization*; dan
5. Menjalankan perintah *Otsu Thresholding*.

6.2.3 Hasil dan analisis pengujian

Hasil pengujian pada sistem ini berupa 2 kondisi yang berbeda, yaitu kondisi 1 mendiagnosis penyakit diabetes melitus pada lidah penderita penyakit diabetes melitus, sedangkan kondisi 2 mendiagnosis penyakit diabetes melitus pada lidah non penderita diabetes melitus, lalu tingkat benar dan kesalahan yang dihitung. Kondisi 1 sistem melakukan pengujian lidah yang terdiagnosis penyakit diabetes melitus pada lidah penderita diabetes melitus. 10 Lidah yang akan didiagnosis ditampilkan pada Gambar 6.2



Gambar 6.2 Citra data uji lidah penderita Diabetes Melitus

Kondisi 2 sistem melakukan pengujian lidah yang terdiagnosis penyakit diabetes melitus pada lidah non penderita diabetes melitus. 10 Lidah yang akan didiagnosis ditampilkan pada Gambar 6.3.



Gambar 6.3 Citra data uji lidah non penderita Diabetes Melitus
 Hasil Pengujian dapat dilihat pada Tabel 6.2.

Tabel 6.2 Hasil pengujian diagnosis penyakit Diabetes Melitus

Orang ke-	Kondisi Yang Sebenarnya	Kondisi Yang Terdiagnosis Sistem	Keterangan
1	Diabetes	Non Diabetes	Gagal
2	Diabetes	Diabetes	Berhasil
3	Diabetes	Diabetes	Berhasil
4	Diabetes	Diabetes	Berhasil
5	Diabetes	Diabetes	Berhasil
6	Diabetes	Diabetes	Berhasil
7	Diabetes	Diabetes	Berhasil
8	Diabetes	Non Diabetes	Gagal
9	Diabetes	Diabetes	Berhasil
10	Diabetes	Diabetes	Berhasil
11	Non Diabetes	Non Diabetes	Berhasil
12	Non Diabetes	Non Diabetes	Berhasil
13	Non Diabetes	Non Diabetes	Berhasil
14	Non Diabetes	Non Diabetes	Berhasil
15	Non Diabetes	Non Diabetes	Berhasil
16	Non Diabetes	Non Diabetes	Berhasil
17	Non Diabetes	Non Diabetes	Berhasil
18	Non Diabetes	Diabetes	Gagal
19	Non Diabetes	Non Diabetes	Berhasil
20	Non Diabetes	Non Diabetes	Berhasil



Pengujian ini melibatkan 10 lidah penderita diabetes dan 10 lidah non penderita diabetes melitus. Dari 10 Lidah penderita diabetes melitus sistem dapat mendiagnosis penyakit diabetes melitus sebanyak 8 orang dan dari 10 Lidah penderita non diabetes melitus sistem dapat mendiagnosis yang tidak menderita penyakit diabetes melitus sebanyak 9 orang. Dari hasil pengujian dapat dihitung nilai akurasi persentase tingkat kesalahan dari sistem yang dirancang. Rumus menghitung nilai persentase akurasi dapat dilihat pada hasil perhitungan sebagai berikut:

$$\text{Nilai Persentase Akurasi: } \frac{17}{20} \times 100\% = 85\%$$

Kesalahan sistem dalam melakukan diagnosis dikarenakan 40 data latih memiliki pencahayaan yang tidak stabil. Hal ini telah dibuktikan dengan hasil analisa yang peneliti lakukan dengan mengakuisisi citra dengan kondisi penerangan yang sama untuk setiap data latih, hasilnya menunjukkan sistem mampu lebih akurat dalam mendiagnosis penyakit diabetes melitus.

6.3 Pengujian waktu eksekusi program

Pengujian waktu eksekusi perintah atau waktu komputasi dari pemrosesan sistem dilakukan untuk mengetahui waktu yang dibutuhkan oleh sistem untuk melakukan perintah yang diberikan oleh *user*, perintah yang akan diuji eksekusi perintahnya ialah perintah akuisisi citra dan perintah diagnosis penyakit diabetes melitus pada lidah penderita dan lidah non penderita

6.3.1 Tujuan pengujian

Tujuan dari pengujian waktu eksekusi adalah untuk mengetahui kecepatan sistem untuk melakukan komputasi terhadap suatu perintah.

6.3.2 Prosedur pengujian

Prosedur pengujian untuk menguji waktu eksekusi perintah dilakukan secara terpisah sesuai dengan perintah yang digunakan. Berikut merupakan prosedur yang akan dijalankan dengan melakukan pengujian ini:

1. Menjalankan perintah akuisisi citra sebanyak 10 kali.
2. Menjalankan perintah diagnosis penyakit diabetes melitus yang terdiri dari perintah konversi citra RGB ke *Grayscale*, *Histogram Equalization* dan *Otsu's Thresholding*.

6.3.3 Hasil dan analisis pengujian

Hasil pengujian pada perintah akuisisi citra. Pengujian ini akan menunjukkan waktu eksekusi dari setiap perintah. Sampel data yang digunakan terdapat 10 kali menjalankan setiap perintah. Hasil pengujian ditunjukkan pada Tabel 6.3.

Tabel 6.3 Hasil pengujian waktu eksekusi program akuisisi citra

Akuisisi Citra Ke-	1	2	3	4	5	6	7	8	9	10	Rata-rata
Waktu Eksekusi (ms)	851	993	995	844	973	846	1000	995	994	877	937

Berdasarkan hasil pengujian akuisisi citra dengan 10 kali pengujian diatas, didapatkan rata-rata waktu eksekusi perintah akuisisi citra ialah 937 ms.

Pengujian kedua dilakukan pada perintah diagnosis penyakit diabetes melitus yang terdiri dari konversi citra RGB ke *Grayscale*, *Histogram Equalization* dan *Otsu's Thresholding*. Pengujian ini dilakukan untuk mengetahui waktu eksekusi perintah diagnosis penyakit diabetes melitus. Hasil pengujian ditunjukkan pada Tabel 6.4.

Tabel 6.4 Waktu eksekusi diagnosis penyakit Diabetes Melitus

No Pengujian	Waktu Eksekusi Perintah RGB to Grayscale (ms)	Waktu Eksekusi Perintah Histogram Equalization (ms)	Waktu Eksekusi Perintah Otsu's Thresholding (ms)
1	5.756	25.803	393.098
2	7.749	8.164	369.256
3	16.659	7.878	235.437
4	7.288	6.362	262.347
5	7.582	7.329	232.935
6	6.241	6.469	230.653
7	9.291	11.74	333.612
8	8.27	9.918	342.613
9	8.396	9.36	437.988
10	8.379	9.661	343.339
11	24.344	38.193	341.091
12	11.38	14.035	245.336
13	14.24	20.909	333.598
14	11.177	13.932	343.319
15	10.353	14.629	236.236
16	10.277	14.323	353.027
17	18.514	27.52	369.627
18	46.933	73.148	445.982
19	37.677	62.217	442.798
20	41.884	43.116	441.559
Rata-rata	8.5611	10.2684	318.1278

Pengujian ini menghasilkan rata-rata waktu untuk eksekusi perintah RGB to Grayscale ialah 8.5611 ms, sedangkan rata-rata waktu untuk eksekusi perintah Histogram Equalization ialah 10.2684 ms dan rata-rata waktu untuk perintah Otsu's Thresholding ialah 318.1278 ms. Pada perintah Otsu's Thresholding memiliki rata-rata yang lebih tinggi karena perintah Otsu's Thresholding memiliki perhitungan yang lebih banyak dibandingkan perintah RGB to Grayscale dan Histogram Equalization.



BAB 7 PENUTUP

7.1 Kesimpulan

Berdasarkan hasil yang telah diteliti dalam sistem “Implementasi Raspberry Pi Untuk Mendiagnosis Penyakit Diabetes Melitus Melalui Warna Lidah Menggunakan Metode Otsu’s Thresholding” sehingga mendapat kesimpulan sebagai berikut:

1. Pada perancangannya, sistem diperlukan beberapa perangkat keras diantaranya Raspberry Pi dan Kamera. Pada perancangan perangkat lunak diperlukan program akuisisi citra, konversi citra RGB to *Grayscale*, *Histogram Equalization* dan *Otsu’s Thresholding*.
2. Pada implementasinya sistem ini dirancang mampu menjalankan beberapa program, diantaranya program akuisisi citra digunakan untuk mengambil gambar lidah. Program Konversi RGB to *Grayscale* digunakan untuk mengubah citra dalam kondisi warna RGB menjadi warna dengan skala keabuan. Program *Histogram Equalization* digunakan untuk memperbaiki histogram citra pada sehingga histogram pada citra menjadi merata. Program *Otsu’s Thresholding* digunakan untuk mencari nilai *threshold* dari citra lidah.
3. Pada analisis fungsionalitas kamera, terdapat rata-rata persentase berhasil mengakuisisi citra sebesar 100%. Pada analisis pengujian diagnosis penyakit diabetes melitus terdapat keakurasian sebesar 85%, sistem kurang berhasil mendiagnosis penyakit diabetes mellitus dikarenakan pencahayaan di tiap-tiap data latih tidak stabil. Pada pengujian waktu eksekusi program mendiagnosis penyakit diabetes melitus memiliki rata-rata waktu akuisisi citra sebesar 937 ms, program diagnosis diabetes melitus pada perintah RGB to *Grayscale* memiliki rata-rata waktu 15.617 ms, program *Histogram Equalization* memiliki rata-rata waktu eksekusi sebesar 18.804 ms dan *Otsu’s Thresholding* memiliki rata-rata waktu eksekusi 336.693 ms. Perintah *Otsu’s Thresholding* memiliki rata-rata yang lebih lama karena perintah *Otsu’s Thresholding* memiliki perhitungan yang lebih banyak dibandingkan perintah RGB to *Grayscale* dan *Histogram Equalization*.

7.2 Saran

Berdasarkan hasil yang telah dianalisis dalam sistem “Implementasi Raspberry Pi Untuk Mendiagnosis Penyakit Diabetes Melitus Melalui Warna Lidah Menggunakan Metode Otsu’s Thresholding” sehingga peneliti memiliki beberapa saran:

1. Dalam penelitian ini karena mengandalkan kamera, objek penelitian harus mendapat kondisi yang sama, terutama dalam pencahayaan dalam pengambilan gambar, untuk penelitian selanjutnya diharapkan sistem prototipe dibuatkan tempat khusus lidah dalam kondisi tertutup dan ditambahkan pencahayaan buatan, sehingga objek penelitian mendapat perlakuan yang sama menjadikan akurasi sistem meningkat.
2. Sistem dapat dikembangkan dengan menggunakan metode Jaringan Syaraf Tiruan.
3. Sistem dapat dikembangkan agar Pengguna dapat melihat hasil diagnosis pada smartphone.



DAFTAR PUSTAKA

- Ahmad, Hadinegoro, 2012. *Metode Histogram Equalization Untuk Perbaikan Citra Digital*. Semarang: Seminar Nasional Teknologi Informasi & Komunikasi Terapan.
- Alang, Sattu, 2015. *Urgensi Diagnosis Dalam Mengatasi Kesulitan Belajar*. Makassar: UIN Alauddin Makassar.
- Amaliah, Balqis, dkk., 2016. *Diagnosis Awal Citra Melanoma Menggunakan Klasifikasi Voted Perseptron*. Surabaya: Institut Teknologi Sepuluh November.
- Badan Pusat Statistik, 2016. *Persentase Penduduk yang Mempunyai Keluhan Kesehatan Selama Sebulan Terakhir*. Tersedia di: <<https://www.bps.go.id/linkTableDinamis/view/id/921>> [Diakses 2 Februari 2018].
- Banu, dkk., 2016. *Diagnosis of Diabete Using Tongue Color Analysis*. International Journal of Emerging Technologies in Engineering Rearch, Vol.4, Issue 5. (Diterima untuk publikasi Mei 2016).
- Baynest, Habtamu, 2015. *Classification, Pathophysiology, Diagnosis and Management of Diabetes Melitus*. Gondar: University of Gondhar.
- CNN Indonesia, 2017. *Studi Ungkap Kesalahan Diagnosis Penderita Diabetes Tipe 3* (online), (<https://www.cnnindonesia.com/gaya hidup /20171031112425-255-252373/studi-ungkap-kesalahan-diagnosis-penderita-diabetes-tipe-3>, diakses 26 November 2018).
- Devi, Uma, dkk., 2015. *Disease Diagnoss for Various Signs Using Tongue Color Image Segmentation*. Australian Journal of Basic and Applied Sciences. (Diterima untuk publikasi Februari 2015).
- Dwi, Prasetya, dkk., 2017. *Perancangan dan Implementasi Tracking Object Manusia untuk Autonomous Car*. Bandung: Universitas Telkom.
- Fajar, Ahmad, dkk., 2013. *Penggunaan Raspberry Pi sebagai Alternatif Micro Controller pada Robot Sederhana*. Jakarta: Universitas Muhammadiyah Prof.Dr.Hamka.
- Gonzales, dkk., 2002. *Digital Image Processing*. Knoxville: University of Tennessee.
- Hamdy, 2016. *Bone Mineral Density and Fractures*. Journal of Clinical Densitometry. (Diterima untuk publikasi Mei 2016)

- Herlambang, Rama A., 2013. *Sistem Pakar Mendiagnosa Penyakit Infeksi Gigi dan Mulut Pada Manusia Menggunakan Metode Certainty Factor*. Semarang: Universitas Dian Nuswantoro.
- Hermawati, F. A., 2013. *Pengolahan Citra Digital*. Yogyakarta: ANDI.
- InfoDATIN Pusat Data dan Informasi Kementerian Kesehatan RI., 2014. *Situasi dan Analisis Diabetes*. Jakarta: Kementerian Kesehatan RI Pusat Data dan Informasi.
- Kadir, Abdul, dkk., 2013. *Teori dan Aplikasi Pengolahan Citra*. Yogyakarta: ANDI.
- Muntasa, Arif, dkk., 2015. *Segmentasi Obyek Pada Citra Digital Menggunakan Metode Otsu Tresholding*. Madura: Universitas Trunojoyo Madura.
- Notosiswoyo dan Suswanti, 2004 . *Pemanfaatan Magnetic Resonance Imaging (MRI) Sebagai Sarana Diagnosa Pasien*. Makassar: Universitas Hasanuddin.
- Nugroho, Hanung, dkk., 2014. *Diagnosa Dini Penyakit Gangguan Jiwa Menggunakan Metode Fuzzy Mamdani*. Yogyakarta: Universitas Gadjah Mada.
- Prana, Duta, 2015. *Sistem Antarmuka Pada Mikroskop Refleksi Digital Berbasis Arduino Uno*. Palembang: Politeknik Negeri Sriwijaya.
- Putra, Darma, 2010. *Pengolahan Citra Digital-Ed. 1*. Yogyakarta: Penerbit ANDI.
- Ramadhani, Putri, 2006. *Elektronika Kedokteran "CT Scanner"*. Makassar: Universitas Hasanuddin.
- Saritha, Kannan, dkk., 2013. *Disease Analysis Using Tongue Image*. International Journal of Engineering Research & Technology, Vol. 2 Issue 4. (Diterima untuk publikasi April 2013).
- Scheid, Volker, 2007. *Traditional Chinese Medicine – What are We Investigating?*. London: University of Westminster.
- Tanudjaja, George N., 2013. *Persarafan Lidah*. Jurnal Biomedik, Vol. 5 Nomor 3. (Diterima untuk publikasi November 2013).
- The Sun, 2017. *New Danger There is a Third Type of Diabetes and People are Being Misdiagnosed, experts warn* (online), (<https://www.thesun.co.uk/news/4788651/diabetes-third-type-doctors-misdiagnose-warn/>), diakses 26 November 2018).
- Utami, Arum, 2017, *Implementasi Metode Otsu Thresholding Untuk Segmentasi Citra Daun*. Solo: Universitas Muhammadiyah Surakarta.

Wei, dkk., 2010 . *Using Treshold Method to Separate the Edge Coating and Body of Tongue in Automatic Tongue Diagnosis*. Taichung-Taiwan: Chaoyang University of Technology.

World Health Organization South-East Asia Regional Asia, 2015. *Diabetes dan Angka* (online), (<http://www.searo.who.int/indonesia/topics/8-whd2016-diabetes-facts-and-numbers-indonesian.pdf>, diakses 26 November 2018).

Zhang, Junwen, dkk., 2015. *Extraction of Tongue Feature Related to TCM Physique Based on Image Processing*. Beijing: Beijing University of Technology.

