

**SISTEM PORTABEL DIAGNOSA DINI RETINOPATI
DIABETIK DENGAN MENERAPKAN ALGORITMA
*CONVOLUTIONAL NEURAL NETWORK***



UNIVERSITAS ANDALAS

LAPORAN TUGAS AKHIR TEKNIK KOMPUTER

RAHMATUL HUSNA

1611513011



UNTUK KEDJAJAAN BANGSA

**JURUSAN TEKNIK KOMPUTER
FAKULTAS TEKNOLOGI INFORMASI
UNIVERSITAS ANDALAS
PADANG
2021**

**SISTEM PORTABEL DIAGNOSA DINI RETINOPATI
DIABETIK DENGAN MENERAPKAN ALGORITMA
*CONVOLUTIONAL NEURAL NETWORK***



**JURUSAN TEKNIK KOMPUTER
FAKULTAS TEKNOLOGI INFORMASI
UNIVERSITAS ANDALAS
PADANG
2021**

HALAMAN PENGESAHAN

Nama : Rabmatul Husna
No BP : 1611513011
Judul Tugas Akhir : Sistem Portabel Diagnosa Dini Retinopati Diabetik dengan
Menerapkan Algoritma *Convolutional Neural Network*

Tugas Akhir ini disetujui oleh Dosen Pembimbing dan disahkan oleh Ketua Jurusan
Teknik Komputer, Fakultas Teknologi Informasi Universitas Andalas.

Demikianlah lembaran pengesahan ini dibuat untuk diketahui bersama

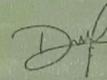
Padang, 09 Juni 2021

Pembimbing 1



Lathifah Arief, M.T
NIP.198109122014042001

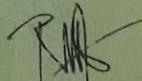
Pembimbing 2



Desta Yolanda, M.T
NIDK.8835411019

Mengetahui :

Ketua Jurusan Teknik Komputer
Fakultas Teknologi Informasi
Universitas Andalas
Padang, 09 Juni 2021



Dr. Eng. Rian Ferdian, M.T
NIP.198809162014041001

LEMBARAN PERSETUJUAN TUGAS AKHIR

Dengan ini dinyatakan bahwa :

Nama : Rahmatul Husna

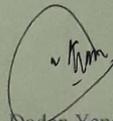
No.BP : 1611513011

Judul Tugas Akhir : **Sistem Portabel Diagnosa Dini Retinopati Diabetik dengan
Menerapkan Algoritma *Convolutional Neural Network***

Telah diujikan dan telah disetujui Seminar Hasil Tugas Akhirnya sebagai salah satu syarat untuk memperoleh gelar Sarjana Teknik (S.T) melalui ujian sidang TA yang diadakan pada tanggal 05 Mei 2021 berdasarkan ketentuan yang berlaku.

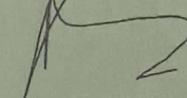
Padang, 01 Juni 2021

Penguji 1



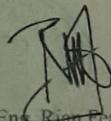
Dodon Yendri, M.Kom
NIP. 196603091986031001

Penguji 2



Ratna Aisuwarya, M.Eng
NIP. 198410302008122002

Mengetahui :
Ketua Jurusan Teknik Komputer
Fakultas Teknologi Informasi
Universitas Andalas
Padang, 01 Juni 2021



Dr. Eng. Rian Ferdian, M.T
NIP. 198609162014041001

LEMBARAN PENGESAHAN SIDANG TUGAS AKHIR

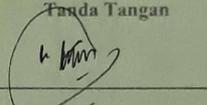
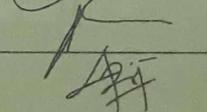
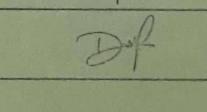
Tim Penguji Tugas Akhir Mahasiswa Jurusan Teknik Komputer, menyatakan bahwa :

Nama : Rahmatul Husna
No.BP : 1611513011
Judul Tugas Akhir : Sistem Portabel Diagnosa Dini Retinopati Diabetik dengan
Menerapkan Algoritma *Convolutional Neural Network*

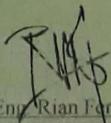
Telah diujikan dan telah disetujui Seminar Hasil Tugas Akhirnya pada seminar hasil untuk dilanjutkan pada tahap berikutnya.

Demikianlah lembaran pengesahan ini dibuat untuk diketahui bersama.

Padang, 01 Juni 2021

	Nama	Tanda Tangan
Penguji I	: Dodon Yendri, M.Kom	
Penguji II	: Ratna Aisuwarya, M.Eng	
Pembimbing I	: Lathifah Arief, M.T	
Pembimbing II	: Desta Yolanda, M.T	

Mengetahui :
Ketua Jurusan Teknik Komputer
Fakultas Teknologi Informasi
Universitas Andalas
Padang, 01 Juni 2021


Dr. Eng Rian Ferdian, M.T
NIP. 198609162014041001

PERNYATAAN

Dengan ini menyatakan bahwa :

1. Tugas Akhir dengan judul "Sistem Portabel Diagnosa Dini Retinopati Diabetik Dengan Menerapkan Algoritma *Convolutional Neural Network*" adalah asli dan belum pernah diajukan untuk mendapatkan gelar akademik (Sarjana, Magister, dan Doktor), baik di Universitas Andalas maupun di perguruan tinggi lainnya.
2. Tugas Akhir ini murni gagasan dan rancangan saya sendiri, tanpa bantuan tidak sah dari pihak lain, kecuali bantuan dan arahan dari tim pembimbing.
3. Tugas Akhir ini tidak terdapat hasil karya atau pendapat yang ditulis atau dipublikasikan oleh orang lain, kecuali dikutip secara tertulis dengan jelas dan dicantumkan sebagai acuan dalam tulisan saya dengan disebutkan nama pengarang dan dicantumkan dalam daftar pustaka.
4. Pernyataan ini saya buat dengan sesungguhnya, dan apabila dikemudian hari terdapat penyimpangan dan ketidakbenaran dalam pernyataan ini, maka saya bersedia menerima sanksi akademik serta sanksi lainnya sesuai dengan norma dan ketentuan lain yang berlaku.

Demikianlah surat ini dibuat, untuk dapat dipergunakan sebagaimana mestinya.

Padang, 17 Juni 2021

Yang membuat pernyataan,


Rahmatul Husna
1611513011

HALAMAN PENGESAHAN TUGAS AKHIR

بِسْمِ اللَّهِ الرَّحْمَنِ الرَّحِيمِ

Dengan menyebut Nama Allah SWT Yang Maha Pengasih dan Maha Penyayang.

Alhamdulillahirabbil'alamin, segala puji bagi Allah SWT yang selalu memberikan rahmat dan karunia-NYA yang tiada hentinya, hingga setiap langkah dan setiap waktu dalam kehidupan ini diberkati oleh rahmat-NYA, hingga akhirnya sampai kepada salah satu nikmat yang diberikan yaitu berhasil menyelesaikan karya yang sederhana ini, semoga karya ini dapat bermanfaat bagi orang-orang. Shalawat beserta salam kuucapkan kepada suri tauladan yang baik, kepada baginda Rasulullah Muhammad SAW, Allahumma shali'ala Muhammad wa'ala aali Muhammad.

Aku persembahkan karya ini untuk...

AYAH (Efrison)

dan

UMI (Misyenti)

yang telah memberikan segalanya untuk ku, segala perjuangan saya hingga titik ini saya persembahkan pada dua orang paling berharga dalam hidup saya. Hidup menjadi begitu mudah dan lancar ketika kita memiliki orang tua yang lebih memahami kita daripada diri kita sendiri. Terima kasih telah menjadi orang tua yang sempurna ayah dan umi ku..

*kepada **Kakak Khadijah, Kakak Aisyah dan Abang Iman**, terima kasih sebesar-besarnya untuk semangat dan dukungan yang kalian berikan.*

*Untuk Dosen Pembimbing bu **Lathifah Arief, M.T** dan bu **Desta Yolanda, M.T**, Terima Kasih Telah Mengantarku Sampai Gelar Sarjana, izinkan aku mengantarkan ucapan terima kasih, untukmu dosen pembimbing yang telah bersedia mengantarkanku mengantungi gelar sarjana, Terima kasih juga penulis ucapkan kepada Bapak-Ibu Dosen yang telah dengan sabar mendidik dan mengajar selama penulis berkuliah di jurusan teknik komputer. Terima kasih juga*

kepada seluruh staff dan tenaga pendidikan Fakultas Teknologi Informasi
Universitas Andalas.

Untuk Keluarga Besar **A.I.X.P**

Terima kasih telah memberikan kesan dan pesan serta telah memberikan kenangan terindah selama masa perkuliahan penulis, **Ara, Ayeng, Nindy, Opa, Mbak Hana, Ipi, Icin, Ela, Shelin, Daput, Ike, Anggy, Cn, Icha, Farah, Age, Rafki, Khairul, Tony, Dony, Eno, Pije, Ghel, ...** Semoga kita semua sukses di jalan masing-masing.

Untuk Keluarga Besar **MPM KM Unand**

Terima kasih telah menemani dan memberikan kepercayaan kepada penulis untuk berjalan bersama membangun lembaga legislatif Majelis Permusyawaratan Mahasiswa Keluarga Mahasiswa Universitas Andalas 2019-2020. Terima kasih kepada **Muhammad Azardi, Rani Pratama Putra, Antoni, Shintya Emilda Hannum, Ridha Rifki, Taufik, Afifah Della** telah membantu serta membimbing penulis selama berada di DPM KM Unand. Serta terima kasih kepada seluruh anggota dan staff MPM KM Unand yang tidak bisa penulis sebutkan satu per satu.

Hidup LEGISLATIVA!

Untuk orang-orang terdekat dan sahabat

Terima kasih untuk sahabat-sahabat penulis **Rani Pratama Puta, Hanadiah Pertiwi, Annisa Winahyu Anjani, Roudhotul Syifa**, atas semangat dan motivasi yang diberikan, tanpa inspirasi, dorongan dan dukungan yang telah kalian berikan kepada saya, saya mungkin bukan apa-apa saat ini. Terima kasih telah menjadi manusia terbaik di dunia.

Terima Kasih kepada **Tazkia Imani, Nabila Putri Bakri dan Reyana Yunindya Sabilla** sebagai database kisah hidup penulis. Yang selalu siap mendengarkan celoteh penulis, yang selalu berusaha menanggapi keluh kesah penulis, yang selalu sabar menghadapi penulis. Kalian pantas mendapatkan kebahagiaan dan hadiah terbaik dari Allah. Semoga kalian menjadi manusia yang lebih baik, terus menjadi lebih baik dari yang sebelumnya.

Terima kasih kepada '**Partai Tua**' yang selalu bisa mengalihkan pikiran penulis dan memberikan support berupa tumpukan tugas yang diberikan di akhir tahun. Kalian sangat membantu penulis dalam mengefektifitaskan waktu menjadi rapi

dan teratur. Dan kalian selalu ada untuk memberikan semangat kepada penulis berupa guyon. Bagi penulis, kalian adalah **the real MPV at campus**. Terus produktif di kala orang tertidur, tetap berdiri di kala orang terpuruk, tetap bersama di kala orang bercerai-berai. Semoga kita selalu menapaki jalan yang benar sampai akhir hayat.

Terima kasih untuk **Firra Azzarah Trie Uthami, Yelchi Multri Meisya, Rafki Maisya dan M. Gerald Yusha** atas waktu, motivasi dan semangat yang diberikan kepada penulis dalam menyelesaikan tugas akhir ini, kalian menjadi orang yang layak penulis persembahkan bentuk perjuangan ini. Semoga kita bisa menjadi manusia bermanfaat bagi masyarakat setelah ini.

Terima kasih untuk **Bang Rakan El-Fawwaz dan Kak Ardia Fadilah** yang telah membantu penulis dengan memberikan masukan, saran, serta semangat ketika penulis mendapatkan kendala dalam pengerjaan Tugas Akhir ini.

Terima kasih untuk **Kak Cahaya dan Kak Ami** yang terus mengingatkan dan mengajak penulis kepada hal yang baik. Pengorbanan kakak akan terbalas, kegigihan kakak akan terbayar, Keluh kesah yang kakak simpan akan menjadi saksi disuatu hari nanti. Yang terbaik dari yang terbaik semoga terus mengiringi jalan kakak.

Terima Kasih kepada **Bang Rakan, Bang Jun, Fauzan Hasya, Khairul Fikri, Rangga Pratama Putra, Robi Dwi Putra, Ghel F Sadeli, Jaka Wira Clara, Muhammad Azardi, Antoni dan Mas Yovie** karena kalian pernah menjadi pemimpin yang berpengaruh bagi kepribadian penulis untuk menjadi pribadi yang lebih baik.

Terimakasih kepada **Uni Saidah** karena telah menjadi orang pertama yang mengajarkan keteraturan, kerapian, ketelitian bagi penulis sehingga penulis dapat mengerti segala hal kesekretariatan. Pengajaran uni sangat berarti bagi penulis hampir di seluruh masa perkuliahan penulis.

Terimakasih kepada **Nisa Tekom 17** karena telah menjadi partner dalam menyelesaikan urusan di kampus Unand. Semoga kita bisa melanjutkan kesempatan yang datang bersama-sama lagi.

KATA PENGANTAR

Dengan menyebut Nama Allah SWT Yang Maha Pengasih dan Maha Penyayang. Alhamdulillah rabbil'alamin, segala puji bagi Allah SWT semata, pencipta alam semesta, semoga selalu turunkan rahmat dan karunia-Nya bagi Rasulullah SAW beserta keluarga, para sahabat, pengikutnya dan menjadi penghargaan yang setinggi-tingginya bagi penulis yang telah dapat menyelesaikan Tugas Akhir ini. Dalam penyusunan skripsi ini banyak hambatan serta rintangan yang penulis hadapi. Namun, pada akhirnya penulis dapat melaluinya berkat Rahmat Allah yang Maha Kuasa berupa bimbingan dan bantuan dari berbagai pihak baik secara moral maupun spiritual. Untuk itu, pada kesempatan ini penulis menyampaikan ucapan terimakasih kepada:

1. Kedua orang tua serta anggota keluarga besar penulis yang telah memberikan dorongan yang sangat besar kepada penulis.
2. Ibu Lahifah Arief, M.T dan Ibu Desta Yolanda, M.T selaku pembimbing yang telah meluangkan waktu, tenaga dan pikiran untuk memberikan bimbingan, semangat dan masukan dalam penulisan laporan Tugas Akhir ini.
3. Bapak Mohammad Hafiz Hersyah, M.T selaku dosen PA penulis yang telah membimbing selama masa perkuliahan penulis di Fakultas ini.
4. Bapak Dodon Yendri, M.Kom dan Ibu Ratna Aisuwarya, M.Eng selaku dosen penguji tugas akhir.
5. Bapak Dr. Eng Rian Ferdian, M.T selaku Ketua Jurusan serta jajaran Dosen dan Staf Jurusan Teknik Komputer Fakultas Teknologi Informasi Universitas Andalas.
6. Sahabat yang selalu mendampingi dalam suka maupun duka dalam mengerjakan Tugas Akhir dari awal sampai akhir.
7. Teman-teman seangkatan, AIXP (SKO-TENG)
8. Keluarga besar Majelis Permusyawaratan Mahasiswa Keluarga Mahasiswa Universitas Andalas 2019/2020.
9. Semua pihak yang tidak dapat saya sebutkan satu persatu yang telah membantu memberikan dukungan.

Penulisa mohon maaf atas segala kesalahan yang pernah dilakukan. Semoga skripsi ini dapat memberikan manfaat untuk mendorong penelitian-penelitian selanjutnya.

Padang, 23 Juni 2020



SISTEM PORTABEL DIAGNOSA DINI RETINOPATI DIABETIK DENGAN MENERAPKAN ALGORITMA *CONVOLUTIONAL NEURAL NETWORK*

Rahmatul Husna¹⁾, Lathifah Arief²⁾, Desta Yolanda³⁾

1)Jurusan Teknik Komputer Fakultas Teknologi Informasi Universitas Andalas,

2)Jurusan Teknik Komputer Fakultas Teknologi Informasi Universitas Andalas,

3)Jurusan Teknik Komputer Fakultas Teknologi Informasi Universitas Andalas

ABSTRAK

Retinopati Diabetik (RD) merupakan salah satu komplikasi diabetes mlitus yang terjadi pada retina mata. Kelainan retina (retinopati) yang terjadi berupa aneurismata, melebarnya vena, pendarahan dan eksudat lemak. Kelainan ini akan mempengaruhi penglihatan penderita dan dapat menyebabkan kebutaan permanen jika tidak ditindaklanjuti. Oleh sebab itu, diperlukan sistem pengolahan citra digital yang dapat memberikan diagnosa dini secara cepat, akurat dan objektif.

Penelitian ini bertujuan untuk membangun sistem yang dapat memberikan diagnosa dini Retinopati Diabetik (RD) dengan proses klasifikasi menggunakan algoritma *Convolutional Neural Network* (CNN). Sistem terdiri dari dua komponen utama yang dirancang portabel, yaitu *embedded system* dan *bot* berupa Bot Telegram. Sistem tertanam terdiri dari Raspberry Pi dengan luas bangun sistem yaitu 42,3 cm³. Data yang digunakan pada penelitian ini dari dataset Messidor dan IDRiD.

Pada proses klasifikasi citra 2D fundus menggunakan algoritma CNN, sistem memiliki persentase akurasi *training* dataset paling tinggi pada 4 kelas (Normal, Mild NPDR, Moderate NPDR, dan Severe NPDR) sebesar 100% dengan menerapkan segmentasi RAG Threshold dan persentase rata-rata nilai *error* paling rendah pada *testing* dengan dataset sebesar 66,67%. Pada pengiriman hasil diagnosa dini menggunakan Bot Telegram, proses pengiriman memiliki rata-rata waktu kirim selama 18,6 detik.

Kata kunci : RD, CNN, RAG Threshold, Bot Telegram.

**PORTABLE SYSTEM FOR EARLY DIAGNOSIS DIABETIC
RETINOPATHY USING CONVOLUTIONAL NEURAL NETWORK
ALGORITHM**

Rahmatul Husna¹⁾, Lathifah Arief²⁾, Desta Yolanda³⁾

1)Computer Engineering Faculty of Information Technology Andalas University,

2)Computer Engineering Faculty of Information Technology Andalas University,

3)Computer Engineering Faculty of Information Technology Andalas University

ABSTRACT

Diabetic Retinopathy (RD) is one of the complications of diabetes mlitus that occurs in the retina of the eye. Retinal disorders (retinopathy) that occur in the form of aneurysms, dilated veins, bleeding and fat exudate. This disorder will affect the patient's vision and can lead to permanent blindness if not followed up. Therefore, a digital image processing system is needed that can provide early diagnosis quickly, accurately and objectively.

This study aims to build a system that can provide early diagnosis of Diabetic Retinopathy (RD) with a classification process using the Convolutional Neural Network (CNN) algorithm. The system consists of two main components designed to be portable, namely an embedded system and a bot in the form of a Telegram Bot. The embedded system consists of a Raspberry Pi with a system building area of 42.3 cm³. The data used in this study were from the Messidor and IDRiD dataset.

In the fundus 2D image classification process using the CNN algorithm, the system has the highest training dataset accuracy percentage in 4 classes (Normal, Mild NPDR, Moderate NPDR, and Severe NPDR) by 100% by applying RAG Threshold segmentation and the average percentage of the most error values. low on testing with a dataset of 66.67%. In sending early diagnosis results using Telegram Bot, the sending process has an average send time of 18.6 seconds.

Keywords: *RD, CNN, RAG Threshold, Telegram Bot.*

DAFTAR ISI

HALAMAN PENGESAHAN.....	i
LEMBAR PERSETUJUAN TUGAS AKHIR.....	ii
HALAMAN PENGESAHAN TUGAS AKHIR.....	iii
HALAMAN PERNYATAAN.....	iv
HALAMAN PERSEMBAHAN.....	v
KATA PENGANTAR.....	ix
ABSTRAK.....	xi
<i>ABSTRACT</i>	xii
DAFTAR ISI.....	xiii
DAFTAR GAMBAR.....	xvii
DAFTAR TABEL.....	xix
BAB I.....	1
PENDAHULUAN.....	1
1.1. Latar Belakang.....	1
1.2. Rumusan Masalah.....	2
1.3. Batasan Masalah.....	2
1.4. Tujuan Penelitian.....	2
1.5. Manfaat Penelitian.....	2
1.6. Jenis dan Metodologi Penelitian.....	3
1.7. Sistematika Penulisan.....	6
BAB II.....	7
LANDASAN TEORI.....	7
2.1. Retinopati Diabetik.....	7
2.1.1. <i>Non-Proliferative Diabetic Retinopathy</i>	7
2.1.2. <i>Proliferative Diabetic Retinopathy</i>	8

2.2. IDRiD.....	8
2.3. Segmentasi <i>RAG Thresholding</i>	9
2.4. Citra Digital.....	9
2.5. Citra RGB.....	9
2.6. Citra <i>Grayscale</i>	10
2.7. <i>Deep Learning</i>	11
2.8. <i>Convolutional Neural Network</i>	12
2.8.1. <i>Convolutional Layer</i>	12
2.8.2. <i>Pooling Layer</i>	13
2.8.3. <i>Fully Connected Layer</i>	13
2.8.4. <i>Dropout</i>	13
2.9. <i>Single Board Computer (SBC)</i>	13
2.10. Google Colab.....	16
2.11. Python.....	16
2.12. Bot Telegram.....	17
2.13. Penelitian Terdahulu.....	28
BAB III.....	19
PERANCANGAN SISTEM.....	19
3.1 Analisis Kebutuhan Sistem.....	19
3.1.1 Kebutuhan Fungsional Sistem.....	19
3.1.2 Kebutuhan Non-Fungsional.....	19
3.1.3 Kebutuhan Perangkat Keras.....	20
3.1.4 Kebutuhan Perangkat Lunak.....	20
3.1.4.1 Kebutuhan Perangkat Lunak pada <i>Personal Computer</i> <i>(PC)/Cloud</i>	20
3.1.4.2 Kebutuhan Perangkat Lunak pada Sistem Tertanam.....	20

3.1.4.3	Kebutuhan Perangkat Lunak pada Bot Telegram.....	21
3.1.5	Kebutuhan Data.....	21
3.2	Rancangan Umum Sistem.....	23
3.3	Rencana Proses.....	24
3.3.1	Rencana Perangkat Keras.....	26
3.3.2	Rencana Perangkat Lunak.....	27
3.3.2.1	Program Perangkat Lunak pada PC/cloud.....	27
3.3.2.2	Program Perangkat Lunak pada Sistem Tertanam.....	27
3.3.2.3	Program <i>Pre-Processing</i>	29
3.3.2.3	Program Segmentasi <i>RAG Thresholding</i>	30
3.3.2.4	Program Klasifikasi.....	31
3.3.2.5	Program Pelatihan Jaringan CNN.....	31
3.3.2.6	Program Pengujian Jaringan CNN.....	32
3.3.2.7	Program Perangkat Lunak pada Bot Telegram.....	33
3.4	Rancangan Pengujian.....	35
3.4.2	Rencana Pengujian Perangkat Lunak.....	35
3.5	Pengujian Sistem.....	36
3.6	Analisis Kebutuhan Penelitian.....	37
BAB IV	38
IMPLEMENTASI DAN PENGUJIAN	38
4.1	Implementasi.....	38
4.1.2	Implementasi Perangkat Keras.....	38
4.1.2	Implementasi Perangkat Lunak.....	40
4.1.2.1	Implementasi Klasifikasi Citra Menggunakan CNN.....	41
4.1.2.1.1	Implementasi Proses Set-dataset.....	41
4.1.2.1.2	Implementasi Proses <i>Training</i>	42

4.1.2.1.3. Implementasi Program Segmentasi <i>RAG Thresholding</i>	45
4.1.2.2. Implementasi Program Pengiriman Hasil ke Telegram.....	46
4.2. Pengujian dan Analisis.....	48
4.2.1. Pengujian dan Analisis Perangkat Lunak.....	48
4.2.1.1. Pengujian Program <i>Pre-processing</i> Citra.....	48
4.2.1.2. Pengujian Program CNN.....	49
4.2.1.3. Pengujian Program Pengiriman Hasil ke Telegram.....	54
4.2.1.4. Pengujian Program Indikator Sistem.....	57
4.2.3. Pengujian dan Analisis Keseluruhan Sistem.....	57
BAB V.....	60
PENUTUP.....	60
5.1. Kesimpulan.....	60
5.2. Saran.....	60
DAFTAR PUSTAKA.....	61
LAMPIRAN	

DAFTAR GAMBAR

Gambar 1.1 Diagram Proses Sistem.....	4
Gambar 2.1. a) Mata Normal dan b) Mata penderita RD.....	7
Gambar 2.2. a) Mild NDPR, b) Moderate NDPR, c) Severe NDPR d) PDR.....	8
Gambar 2.3. Skala yang digunakan pada <i>Grayscale</i>	10
Gambar 2.4. Arsitektur Jaringan Syaraf Tiruan.....	11
Gambar 2.5. Proses CNN.....	12
Gambar 2.6. Raspberry Pi 4B.....	14
Gambar 3.1 Rancangan Umum Sistem.....	23
Gambar 3.2 <i>Flowchart</i> Rancangan Proses.....	25
Gambar 3.3 Rancangan Perangkat Keras.....	26
Gambar 3.4 <i>Flowchart</i> Program Perangkat Lunak pada PC/Cloud.....	27
Gambar 3.5 <i>Flowchart</i> Program Perangkat Lunak pada Sistem Tertanam.....	28
Gambar 3.6 <i>Flowchart Pre-Processing</i>	29
Gambar 3.7 <i>Flowchart</i> Segmentasi <i>RAG Thresholding</i>	30
Gambar 3.8 <i>Flowchart</i> Klasifikasi.....	31
Gambar 3.9 <i>Flowchart</i> Pelatihan Jaringan CNN.....	32
Gambar 3.10 <i>Flowchart</i> Pengujian Jaringan CNN.....	33
Gambar 3.11 <i>Flowchart</i> Rancangan Perangkat Lunak pada Bot Telegram.....	34
Gambar 4.1. a). Perangkat Keras.....	38
Gambar 4.1. b) Kotak A.....	39
Gambar 4.1. c) Kotak B.....	39
Gambar 4.2 a). <i>Labeling</i> Dataset menggunakan Microsoft Office.....	42
Gambar 4.2 b). Pemberian label dengan tiga digit angka pembeda.....	42

Gambar 4.3. <i>Model Summary</i> yang digunakan pada proses <i>training</i>	42
Gambar 4.4. <i>RAG Thresholding</i> Segmentasi 1.....	44
Gambar 4.5. <i>RAG Thresholding</i> Segmentasi 2.....	45
Gambar 4.6. Citra Hasil Segmentasi <i>RAG Thresholding</i>	45
Gambar 4.7 a). Tampilan hasil pesan /menu Bot Telegram RD Diagnosa.....	46
Gambar 4.7 b). Tampilan hasil pesan /hasil Bot Telegram RD Diagnosa.....	47
Gambar 4.7 c). Tampilan hasil pesan /foto Bot Telegram RD Diagnosa.....	47



DAFTAR TABEL

Tabel 2.1. Warna dan Nilai Penyusun Warna.....	10
Tabel 2.2. Spesifikasi dan Fitur Raspberry Pi.....	14
Tabel 2.4. Tabel Penelitian Terdahulu.....	18
Tabel 3.1. Sistematika Data.....	22
Tabel 3.2. Pembagian Jenis Dataset.....	22
Tabel 3.3. Rencana Pengujian Perangkat Lunak.....	35
Tabel 3.4. Pengujian Sistem.....	36
Tabel 3.5. Alat dan Bahan Penelitian.....	37
Tabel 4.1. Hasil <i>Training</i>	43
Tabel 4.2. Hasil <i>Pre-processing</i> Citra.....	48
Tabel 4.3. Hasil <i>Testing</i> Dataset A.....	49
Tabel 4.4. Rata-rata Hasil <i>Testing</i> Dataset A.....	50
Tabel 4.5. Hasil <i>Testing</i> Dataset B.....	51
Tabel 4.6. Rata-rata Hasil <i>Testing</i> Dataset B.....	52
Tabel 4.7. Hasil <i>Testing</i> Dataset C.....	52
Tabel 4.8. Rata-rata Hasil <i>Testing</i> Dataset C.....	53
Tabel 4.9. Pengiriman hasil <i>testing</i> ke Telegram Bot Diagnosa RD.....	54
Tabel 4.10. Pengujian keseluruhan indikator sistem.....	56
Tabel 4.11. Keluaran sistem secara keseluruhan.....	57

BAB I

PENDAHULUAN

1.1. Latar belakang

Retinopati Diabetik (RD) merupakan salah satu komplikasi diabetes militus yang terjadi pada retina mata. Kelainan retina (retinopati) yang terjadi berupa aneurismata, melebarnya vena, pendarahan dan eksudat lemak. Kelainan ini akan mempengaruhi penglihatan penderita dan dapat menyebabkan kebutaan permanen karena pecahnya pembuluh darah pada retina mata jika tidak ditindaklanjuti [1]. Salah satu penyebab terjadinya RD yaitu dislipidemia atau kelainan metabolisme lipid dalam darah karena meningkatnya insulin dalam tubuh [2]. RD diawali dengan tanda yang tidak terlihat dan baru dapat terindikasi jika sudah di tingkat DM kronis.

Salah satu cara untuk diagnosa penyakit RD yaitu dengan fotografi fundus. Pemeriksaan fundus retina mata merupakan komponen penting yang memberikan informasi diagnostik yang berharga untuk dokter mata dan nonoptamologi. Proses diagnosa hasil fotografi pada bidang kesehatan dengan menggunakan *computer vision* sudah mulai banyak dikembangkan. Diantaranya, pada penelitian oleh Chintya Dewi Regina Wulandari, Suryo Adhi Wibowo, S.T., M.T., Ph.D. dan Ledy Novamizanti, S.Si., M.T.[3], RD diklasifikasikan berdasarkan empat tingkat keparahan yaitu normal, mild NPDR, moderate NDPR dan serve NPDR. Penelitian ini memanfaatkan *computer vision* yaitu algoritma CNN dan *image processing* yaitu segmentasi SRM. Dengan hasil akurasi yang diperoleh 81,25%. Pada penelitian [3], penelitian dilakukan dengan dataset Messidor sebagai data latih serta data uji dengan jumlah total data yaitu 100 data citra.

Berdasarkan penelitian [3] maka diusulkan sebuah sistem portabel pada penelitian tersebut untuk melakukan pendiagnosaan dini retinopati diabetik menggunakan Raspberry Pi, dataset tambahan yaitu IDRiD, serta segmentasi RAG Threshold untuk menggantikan Segmentasi SRM dengan judul “**Sistem Portabel Diagnosa Dini Retinopati Diabetik Dengan Menerapkan Algoritma *Convolutional Neural Network***”.

1.2. Rumusan masalah

Berdasarkan latar belakang yang telah di uraikan, maka rumusan masalah yang diangkat adalah sebagai berikut:

1. Bagaimana caranya menerapkan diagnosa dini RD dengan meng-*input* dataset menggunakan algoritma CNN.
2. Bagaimana caranya meningkatkan nilai akurasi pada hasil diagnosa dini RD dengan menerapkan segmentasi RAG Threshold.
3. Bagaimana caranya mengirim hasil diagnosa dini RD ke *smartphone* pengguna.

1.3. Batasan masalah

Beberapa aspek permasalahan yang nantinya akan menjadi batasan dalam penelitian kedepannya adalah sebagai berikut:

1. Batas pengujian dilakukan pada dataset Messidor dan IDRiD.
2. Pengguna dapat menggunakan perangkat ini selama memiliki koneksi jaringan.
3. Aplikasi berupa aplikasi *mobile*.

1.4. Tujuan penelitian

Tujuan dari penelitian ini nantinya adalah sebagai berikut:

1. Membangun sistem yang mampu menerapkan diagnosa dini RD menggunakan algoritma CNN terhadap dataset.
2. Membangun sistem yang mampu meningkatkan nilai akurasi pada diagnosa dini RD dengan menerapkan segmentasi RAG Threshold.
3. Membangun Bot Telegram yang dapat menerima hasil diagnosa dini dan kondisi RD ke *smartphone* pengguna.

1.5. Manfaat Penelitian

Adapun manfaat dari penelitian ini adalah:

1. Hasil penelitian ini dapat memperkaya hazanah ilmu pengetahuan dalam bidang teknologi, khususnya biomedis.
2. Hasil penelitian ini dapat membantu klasifikasi retinopati diabetik melalui dataset fundus retina mata.

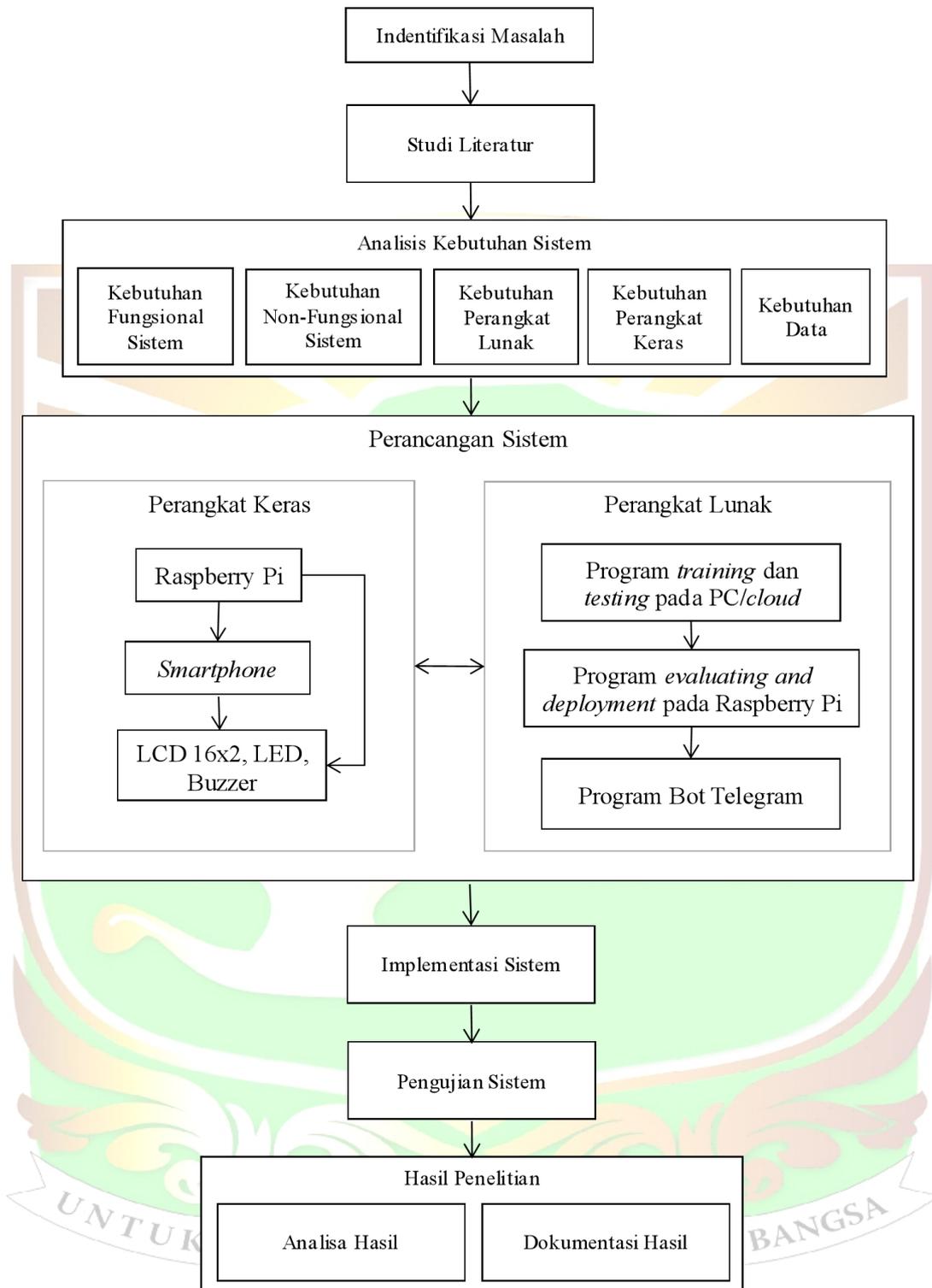
3. Hasil penelitian ini dapat memberi masukan kepada peneliti lain dalam bidang *computer vision* dan *image processing*.

1.6. Jenis dan Metodologi Penelitian

Jenis penelitian yang digunakan dalam pembuatan penelitian ini adalah penelitian eksperimental (*Experimental Research*). Penelitian ini digunakan untuk melihat sebab akibat dan bertujuan untuk membandingkan pengaruh suatu perlakuan atau tindakan terhadap tindakan lain. Percobaan yang dilakukan pada penelitian eksperimental dirancang secara khusus guna membandingkan data yang diperlukan untuk menyelesaikan penelitian. Pada penelitian ini, sistem diagnosa dini merupakan salah satu kebutuhan dari permasalahan yang dirasakan untuk mendapatkan diagnosa dini RD.

Rancangan penelitian merupakan dasar dari penelitian untuk mencapai tujuan penelitian secara terstruktur. Sistematika dalam melakukan penelitian dilakukan dengan langkah seperti diagram proses sistem pada gambar 1.1 berikut :





Gambar 1.1 Diagram Proses Sistem

Berdasarkan diagram 1.1 dapat di jelaskan tahapan tahapan dari rancangan sistem sebagai berikut :

A. Identifikasi Masalah

Pada tahap ini diidentifikasi masalah, yaitu:

Sistem yang sudah memanfaatkan *computer vision* yaitu CNN dan *image processing* yaitu segmentasi SRM, namun masih menggunakan data yang jumlahnya sedikit dalam melakukan proses klasifikasi sebagai data uji. Oleh karena itu, dibuatlah sistem yang dapat memberikan diagnosa dini RD pada dataset Messidor dan IDRiD dengan klasifikasi citra menggunakan algoritma CNN dan segmentasi RAG Threshold pada Raspberry Pi.

B. Studi Literatur

Studi literatur dilakukan untuk mencari dan memahami teori ilmiah yang dapat dijadikan landasan dalam perancangan sistem. Pada tahap ini, dilakukan pencarian dan pemahaman sumber literatur mengenai RD pada citra 2D fundus retina mata menggunakan algoritma CNN. Mencari dan memahami literatur tentang segmentasi *RAG Thresholding* serta literatur terkait pengiriman data hasil sebagai media untuk menampilkan hasil diagnosa dini RD ke *aplikasi mobile*.

C. Analisis Kebutuhan Sistem

Pada analisis kebutuhan sistem, terdapat 2 kebutuhan yang diperlukan agar sistem dapat dibangun dan bekerja dengan baik, diantaranya perancangan sistem yang terdiri dari perangkat lunak dan analisis kebutuhan data.

D. Implementasi Sistem

Tahap implementasi sistem menggambarkan tentang proses yang terjadi pada sistem. Sistem nantinya akan berupa perangkat lunak dan kebutuhan data

E. Pengujian Sistem

Pada pengujian sistem, dilakukan pengujian secara keseluruhan. Sistem melalui proses segmentasi dengan menerapkan segmentasi RAG Threshold, kemudian melalui proses klasifikasi hasil citra dataset dengan menerapkan algoritma CNN, selanjutnya, hasil diagnosa dini tersebut dikirim ke aplikasi *mobile* yaitu telegram sebagai hasil diagnosa dini RD dengan memanfaatkan bot telegram. Seluruh rangkaian proses kemudian diindikasikan dengan LED, LCD 16 x 2, dan *Buzzer*.

F. Analisis Hasil

Dari pengujian sistem akan dilakukan analisis kinerja sistem dan data yang didapatkan selama pengujian sistem.

G. Dokumentasi Hasil

Dokumentasi hasil sebagai pelaporan dari hasil yang didapatkan di penelitian Tugas Akhir.

1.7. Sistematika Penulisan

Secara garis besar, tugas akhir ini dibagi menjadi beberapa bab. Adapun bab tersebut adalah:

BAB I PENDAHULUAN

Memuat latar belakang, perumusan masalah, batasan masalah, tujuan penelitian dan sistematika penulisan.

BAB II LANDASAN TEORI

Memuat pemaparan teori dasar yang mendukung penelitian tugas akhir ini.

BAB III METODOLOGI PENELITIAN

Memuat proses perancangan pembangunan sistem, perancangan sistem, perancangan perangkat lunak dan rancangan pengujian sistem.

BAB IV IMPLEMENTASI DAN PENGUJIAN

Memuat tentang pengimplementasian sistem yang dibangun dan pengujian-pengujian yang dilakukan terhadap sistem seperti yang telah direncanakan pada

BAB III

BAB V PENUTUP

Memuat kesimpulan yang diperoleh dari hasil implementasi sistem dan pengujian sistem secara keseluruhan. Pada bab ini juga terdapat saran yang dapat dilakukan untuk pengembangan sistem selanjutnya.

DAFTAR PUSTAKA

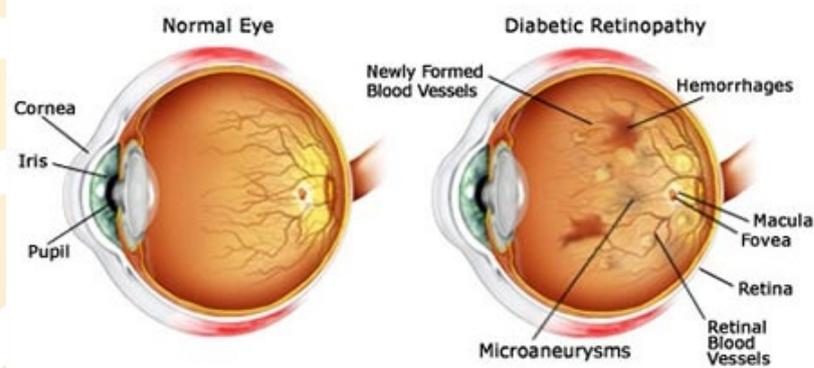
Berisi tentang sumber literatur yang digunakan dalam pembuatan tugas akhir.

BAB II

LANDASAN TEORI

2.1. Retinopati Diabetik

Retinopati diabetik (RD) adalah komplikasi retina yang umum terkait dengan diabetes. Ini adalah penyebab utama kebutaan pada kelompok umur menengah dan lebih tua. [4]. Berbagai aspek dan tahapan RD dianalisis dengan memeriksa gambar retina berwarna. Mikroaneurisma adalah kantong sakular kecil yang disebabkan oleh distensi lokal dinding kapiler dan muncul sebagai titik merah kecil. Dindingnya tipis dan mudah pecah menyebabkan pendarahan [4].



Gambar 2.1 a) Mata Normal dan b) Mata penderita RD [6]

Dari Gambar 2.1 a) dan b) dapat diamati perbedaan antara mata normal dan mata yang terindikasi RD. Pada gambar 2.1 a) yaitu gambar mata normal tampak bersih tanpa ada bercak merah. Sedangkan pada gambar 2.1 b) yaitu gambar mata dari penderita RD. Pada retina mata terdapat pembuluh darah yang baru terbentuk (*newly formed blood vessels*), pendarahan (*hemorrhages*) dan pembengkakan pembuluh darah yang berukuran mikro (*microaneurysms*). Secara umum RD memiliki tingkat keparahan yang terdiri dari dua jenis, yaitu *Non-Proliferative Diabetic Retinopathy* dan *Proliferative Diabetic Retinopathy* [5].

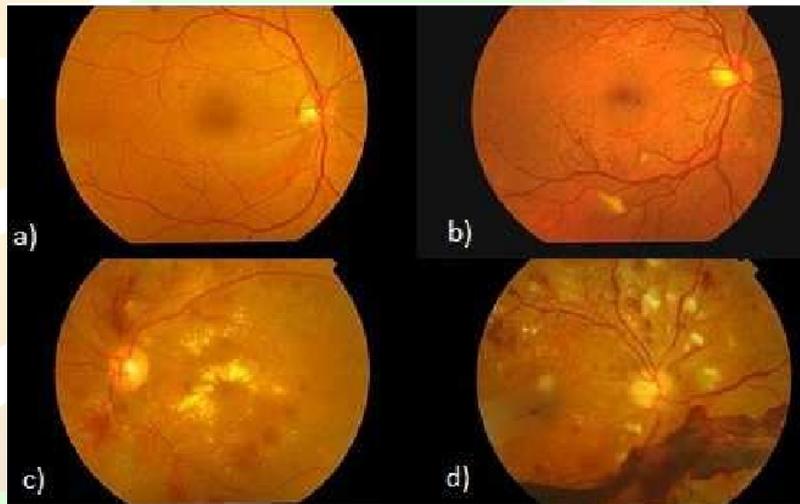
2.1.1. *Non-Proliferative Diabetic Retinopathy*

Non-Proliferative Diabetic Retinopathy (NPDR) merupakan tahap awal dari RD. NPDR diklasifikasikan lagi dengan beberapa tingkat keparahan, diantaranya terdapat *mild*, *moderate* dan *serve*. Pada tahap NPDR, pembuluh darah kecil retina mengalami pendarahan atau bocor. Cairan yang bocor ini menyebabkan

retina membengkak dan membentuk gumpalan-gumpalan atau yang disebut *exudate* [6]. Selain itu, NPDR ditandai juga dengan munculnya mikroaneurisma dan *hemorrhage* [7].

2.1.2. Proliferative Diabetic Retinopathy

Proliferative Diabetic Retinopathy (PDR) adalah tahap yang lebih parah setelah NPDR. Pada tahap ini, terdapat pembuluh darah baru atau jaringan *fibrous*, atau keduanya, muncul dari retina atau diskus optikus dan menyebar melalui permukaan dalam retina ke ruang *vitreous* [5]. Berikut merupakan gambar dari fundus retina *mild* NDPR, *moderate* NDPR, *severe* NDPR dan PDR.



Gambar 2.2. a) Mild NDPR, b) Moderate NDPR, c) Severe NDPR d) DPR [5]

2.2. IDRiD

IDRiD atau *Indian Diabetic Retinopathy Image Dataset* merupakan dataset yang terdiri dari kumpulan data citra fundus retina. IDRiD telah dijadikan sebagai dataset dari tantangan besar yaitu “Diabetic Retinopathy – Segmentation and Grading” yang diselenggarakan sehubungan dengan Simposium Internasional IEEE tentang *Biomedical Imaging* (ISBI - 2018) [8]. Ada tiga sub tantangan utama: segmentasi lesi, tingkat keparahan penyakit, dan lokalisasi landmark retinal dan segmentasi. Pada penelitian [8] telah didapatkan pembagian tingkat keparahan penyakit RD dari IDRiD berupa 5 kelas yaitu mata normal, mata yang terindikasi RD yaitu *mild*, *moderate*, *severe* dan PDR.

2.3. Segmentasi RAG Threshold

Region Adjacency Graphs (RAG) digunakan untuk merepresentasikan wilayah dan hubungan di antara mereka dalam sebuah gambar. Penekanannya adalah pada partisi image dalam bentuk *region* dan karakteristik tiap partisi. Properti wilayah yang berbeda dapat disimpan dalam struktur data *node*. RAG menekankan kedekatan wilayah dan memainkan peran penting dalam segmentasi. Setelah dilakukan segmentasi awal berdasarkan karakteristik primitif seperti nilai intensitas, hasilnya dapat direpresentasikan dalam RAG, kemudian daerah dapat digabungkan untuk mendapatkan segmentasi yang lebih baik. Metode untuk menghasilkan RAG diberikan dalam algoritma berikut [9].

1. Pindai array keanggotaan a dan lakukan langkah-langkah berikut pada setiap indeks piksel $[i, j]$.
2. Misalkan $r1 = a [i, j]$.
3. Kunjungi tetangga $[k, l]$ dari piksel di $[i, j]$. Untuk setiap tetangga, lakukan langkah berikut.
4. Misalkan $r2 = a[k, l]$. Jika $r1 \neq r2$, tambahkan busur antara node $r1$ dan $r2$ pada grafik ketetanggaan daerah.

2.4. Citra Digital

Citra digital adalah fungsi dua dimensi (x, y) dengan ukuran (M, N) . Nilai x dan y merupakan koordinat spasial, dimana nilai dari fungsi x, y dan amplitud f di titik koordinat x, y dinamakan intensitas. Jika nilai x, y dan nilai amplitud f secara keseluruhan memiliki nilai berhingga dan bernilai diskrit, maka bisa dikatakan bahwa citra tersebut adalah citra digital [10].

Citra digital terdiri dari kumpulan titik yang dinamakan *pixel* atau *picture element*. Pada setiap piksel digambarkan sebagai kotak-kotak kecil. Setiap piksel memiliki koordinat posisi seperti pada persamaan (2.1).

$$f(x,y) = \begin{bmatrix} f(0,0) & f(0,1) & \dots & f(0,N-1) \\ f(1,0) & f(1,1) & \dots & f(1,N-1) \\ \vdots & \vdots & & \vdots \\ f(M-1,0) & f(M-1,1) & \dots & f(M-1,N-1) \end{bmatrix} \dots (2.1)$$

Di mana,

M = jumlah baris

N = jumlah kolom indeks baris x dan indeks kolom y menyatakan suatu koordinat piksel citra

$f(x, y)$ = intensitas derajat keabuan

Citra yang didapatkan dari kamera raspberry pi nantinya berupa citra digital. Di mana komposisi yang dimiliki citra digital seperti persamaan 2.1 diatas.

2.5. Citra RGB

Citra RGB merupakan jenis citra dengan warna penyusunnya mempunyai warna dengan bentuk komponen R (*Red*), G (*Green*) dan B (*Blue*). Setiap komponen terdiri dari delapan bit yang nilainya berkisar antara 0 sampai dengan 255. Dengan demikian, memungkinkan warna yang disajikan mencapai $255 \times 255 \times 255$ atau 16.581.375 warna. Tabel 2.2 di bawah menunjukkan contoh warna dan nilai R, G, dan B [10].

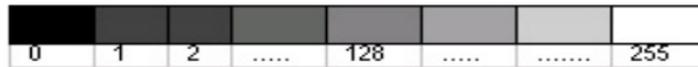
Tabel 2.1. Warna dan Nilai Penyusun Warna

Warna	R	G	B
Merah	255	0	0
Hijau	0	255	0
Biru	0	0	255
Hitam	0	0	0
Putih	255	255	255
Kuning	0	255	255

Di mana citra digital fundus retina mata nantinya memiliki warna RGB. Sebagaimana yang sudah dijabarkan pada tabel 2.1, setiap warna merepresentasikan sebuah angka dari nilai RGB. Angka-angka dari warna tersebut akan membantu proses klasifikasi.

2.6. Citra Grayscale

Citra grayscale adalah citra dengan skala keabuan mulai dari nilai intensitas paling besar 255 yaitu berwarna putih hingga nilai intensitas paling kecil bernilai 0 dengan warna hitam [10] seperti pada gambar 2.3.



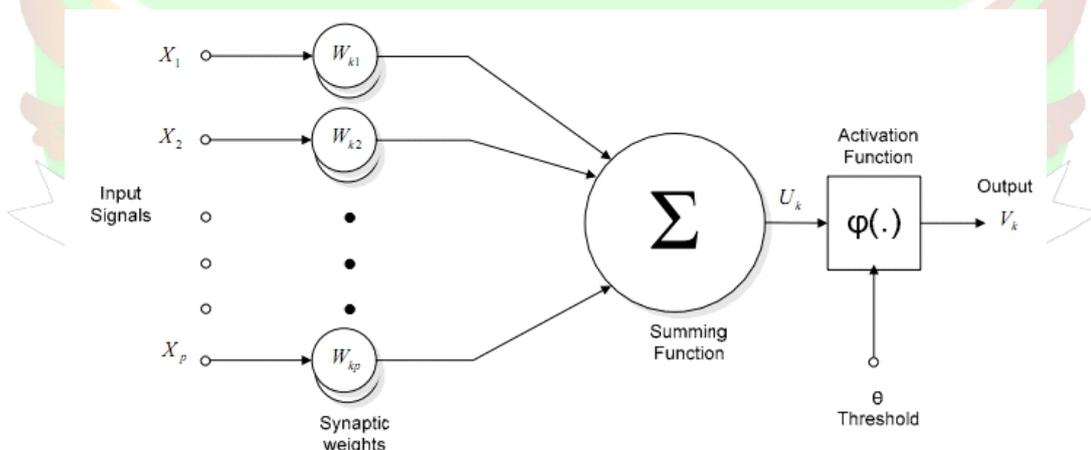
Gambar 2.3. Skala yang digunakan pada *Grayscale* [10]

Pada gambar 2.3 dapat dilihat bahwa citra digital dengan skala keabuan 8-bit memiliki 256 kemungkinan warna, yaitu nilai minimal yaitu 0 hingga nilai maksimal yaitu 255 [10].

2.7. *Deep Learning*

Deep Learning adalah cabang ilmu pembelajaran mesin, dimana model komputasinya memungkinkan terdiri dari beberapa lapisan proses untuk mempelajari representasi dari data dengan berbagai tingkatan abstraksi [11]. Pembelajaran mesin telah menjadi salah satu bidang paling penting dalam organisasi pengembangan yang mencari cara guna memahami aset data untuk membantu bisnis hingga mencapai tingkat pemahaman yang baru [12], di dalam pemrosesan pembelajaran mesin terdapat prinsip dari jaringan syaraf tiruan (JST).

Jaringan Syaraf Tiruan (JST) atau disebut juga dengan *artificial neural network* (ANN) merupakan sistem komputasi yang memiliki arsitektur serta pengoperasiannya berasal dari ilmu tentang sel saraf biologis di dalam otak yang bisa digambarkan sebagai model matematis untuk fungsi aproksimasi tidak linier, klasifikasi data kluster dan regresi yang bukan parametrik atau sebuah simulasi dari koleksi model jaringan syaraf biologi [13].

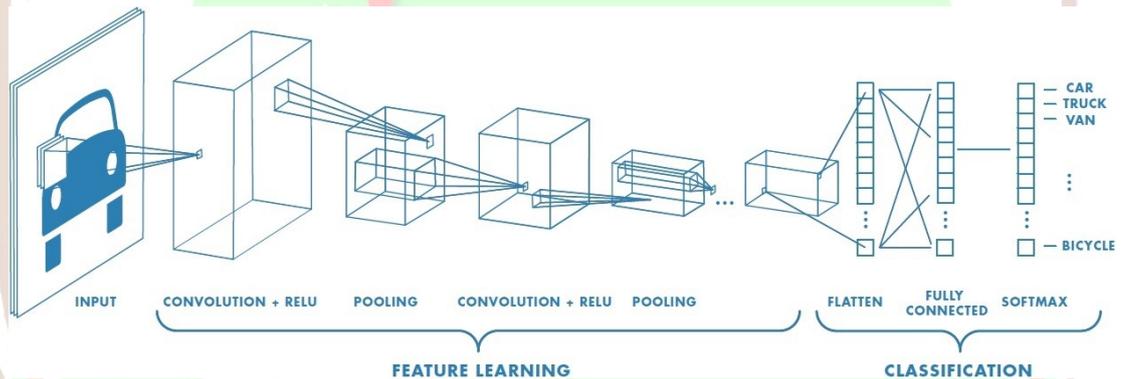


Gambar 2.4. Arsitektur Jaringan Syaraf Tiruan [14]

Pada gambar 2.3 dapat dilihat bahwa JST memiliki arsitektur yang mirip jaringan syaraf manusia. Dimana sinyal masukan pada JST berperan sebagai dendrit pada syaraf manusia, beban sebagai sinapsis, keluaran sebagai akson, dan *node* sebagai badan sel atau soma [15].

2.8. Convolutional Neural Network

Convolutional Neural Network atau disebut juga dengan CNN merupakan pengembangan ilmu dari *Multilayer Perceptron* atau MLP yang di rancang untuk mengolah data dua dimensi. CNN termasuk ke dalam jenis *Deep Neural Network* karena dalamnya jaringan yang tinggi dan banyak diimplementasikan pada data citra. Pada contoh pengelompokan citra, MLP kurang sesuai untuk digunakan, dikarenakan tidak bisa menyimpan informasi spasial dari data citra dan menganggap setiap piksel merupakan fitur yang berdiri sendiri sehingga menghasilkan keluaran yang kurang baik [16].



Gambar 2.5. Proses CNN [17]

Terdapat beberapa layer pada CNN yang berfungsi untuk melakukan filter pada setiap tahapan. Proses tersebut disebut dengan proses pelatihan yang dapat diamati pada gambar 2.4 diatas. Pada proses training terdapat beberapa tahapan yaitu Lapisan Konvolusi, Lapisan Pooling, dan Fully connected layer [18].

2.8.1. Convolutional Layer

Proses pertama setelah memasukkan gambar yaitu Convolutional Layer atau lapisan konvolusional. Semua bagian data yang di masukan dan akan menghasilkan *activation map* atau *feature map 2D*. Filter pada lapisan konvolusi memiliki panjang, tinggi dan tebal sesuai dengan channel data masukan. Setiap

filter akan mengalami pergeseran dan operasi “.” antara data masukan dan nilai dari filter. Lapisan konvolusi secara pesat mengalami model yang kompleks melalui proses pengoptimalan keluarannya. Proses optimalisasi dilakukan dengan melalui tiga parameter, *depth*, *stride* dan pengaturan *zero padding* [18].

2.8.2. Pooling Layer

Pooling Layer atau lapisan penyatuan adalah tahapan yang dilakukan setelah lapisan konvolusi. Lapisan penyatuan terdiri dari filter dengan ukuran dan *stride* atau langkah tertentu. Setiap perubahan disebabkan oleh jumlah langkah yang akan digeser pada seluruh area di *feature map* atau *activation map*. Dalam pengaplikasiannya, lapisan penyatuan yang biasa digunakan adalah *Max Pooling* serta *Average Pooling*. Sebagai contoh, jika digunakan *Max Pooling 2x2* dengan Langkah 2, maka di setiap pergeseran filter, nilai yang diambil yaitu nilai yang terbesar di area 2x2 tersebut, Sedangkan *Average Pooling* diambil dari nilai rata-rata [18].

2.8.3. Fully Connected Layer

Feature map yang merupakan hasil dari tahap sebelumnya berbentuk array berdimensi banyak. Sehingga, sebelum memasuki tahap Fully Connected Layer, Feature Map tersebut akan melalui tahap “flatten” atau perubahan ukuran. Tahap flatten akan menghasilkan vektor yang digunakan sebagai masukan dari Fully Connected Layer atau lapisan terkoneksi sempurna. Fully Connected Layer memiliki beberapa lapisan tersembunyi, fungsi aktivasi, lapisan keluaran dan Loss Function [18].

2.8.4. Dropout

Dropout adalah salah satu usaha pencegahan agar tidak terjadi overfitting dan juga menambah kecepatan proses pembelajaran. Overfitting merupakan kondisi yang hampir seluruh datanya telah melalui proses pelatihan dengan mencapai persentase yang baik, namun terjadi ketidaksesuaian pada tahap prediksi. Dalam sistem kerjanya, Dropout menghilangkan sementara suatu neuron yang berupa lapisan tersembunyi maupun lapisan tampak yang berada didalam jaringan [18]

2.9. Single Board Computer (SBC)

Raspberry Pi yang sering disingkat dengan Raspi merupakan komputer papan tunggal atau single-board circuit (SBC) dengan ukuran sebesar kartu kredit yang

biasa digunakan untuk menjalankan program di perkantoran, permainan komputer, atau sebagai pemutar media hingga video dengan resolusi tinggi [19]. Raspberry Pi memiliki system on a chip (SoC) yaitu Broadcom BCM835. SoC adalah sebuah IC dimana semua komponennya diintegrasikan dari sebuah komputer seperti CPU, GPU, RAM menjadi sebuah IC. SoC Broadcom BCM2835 memiliki prosesor ARM1176JZF-S dengan kecepatan sebesar 700 MHz dan dapat ditingkatkan (overclock) menjadi 1 GHz. Juga dilengkapi Video Core IV GPU dan RAM sebesar 512 MB untuk model B serta 256 MB untuk model A [20].



Gambar 2.6. Raspberry Pi 4B[21]

SBC Raspberry Pi terdiri dari banyak model, diantaranya model 4B, 3B+, A+, 3, 2, dan B. Untuk masing-masing model Raspberry pi, memiliki spesifikasi dan fitur yang disediakan seperti pada tabel 2.1 berikut.

Tabel 2.2 Spesifikasi dan Fitur Raspberry Pi

Name product	Raspberry Pi 4 B	Raspberry Pi 3 B+	Raspberry Pi A+	Raspberry Pi 3	Raspberry Pi 2	Raspberry Pi B
Tanggal keluaran	24 Juni 2019	14 Maret 2018	10 November 2014	29 Februari 2016	1 Februari 2015	15 Februari 2012
Tipe SOC	Broadcom BCM2711	Broadcom BCM2837B0	Broadcom BCM2835	Broadcom BCM2837	Broadcom BCM2836	Broadcom BCM2835
Tipe Core	Cortex-A72 (ARM v8) 64-bit	Cortex-A53 64-bit	ARM1176JZ F-S	Cortex-A53 64-bit	Cortex-A7	ARM1176JZ F-S
Nomor	4	4	1	4	4	1

Cores						
GPU		VideoCore IV	VideoCore IV	VideoCore IV 1080p@30	VideoCore IV	VideoCore IV 1080p@30
CPU Clock	1.5 GigaHz	1.4 GigaHz	700 MegaHz	1.2 GigaHz	900 MegaHz	700 MegaHz
RAM	1 GB , 2 GB, 4 GB	1 GB DDR2	256 MB	1 GB DDR2	1 GB	512 MB
USB	Ya 2x USB3.0 + 2x USB2.0	Ya 4x USB2.0	Ya 1	Ya 4x USB2.0 + micro OTG	Ya 4 + OTG	Ya 2x USB 2.0
Ethernet	Ya Gigabit	Ya Gigabit - Over USB 2.0	Tidak	Ya 10/100M	Ya 10/100M	Ya
SATA Ports	Tidak	Tidak	Tidak	Tidak	Tidak	Tidak
Port HDMI	Ya 2x micro HDMI	Ya	Ya	Ya	Ya	Ya
Keluaran Audio Analog	Ya 3.5mili meter jack	Ya 3.5mili meter jack	Ya 3.5mili meter jack	Ya 3.5mili meter jack	Ya 3.5mili meter jack	Ya 3.5mili meter jack
Masukan Audio Analog	Tidak	Tidak	Tidak	Tidak	Tidak	- via GPIO, USB Mic atau USB Sound Card
SPI	Ya	Ya	Ya	Ya	Ya	Ya
I2C	Ya	Ya	Ya	Ya	Ya	Ya
GPIO	Ya	Ya	Ya	Ya 40-pin	Ya	- 26-pins
Panel LCD	Ya	Ya	Ya	Ya	Ya	Ya DSI
Kamera	Ya	Ya	Ya	Ya	Ya	Ya DSI
SD/MMC	Ya microSD	Ya microSD	Ya microSD	Ya microSD	Ya microSD	Ya SD, SDHC dan SDXC hingga 2TB
Seri	-	- RX/TX UART	-	-	-	- Through Expansion

						Connector, Membutuhkan perpindahan level
Wi-Fi	Ya 2.4GHz and 5GHz 802.11 b/g/n/ac	Ya 2.4GHz and 5GHz 802.11 b/g/n/ac	Tidak	Ya 802.11n	Tidak	Tidak
Bluetooth ®	Ya 5.0	Ya 4.2, BLE	Tidak	Ya 4.1 LE	Tidak	Tidak
Dimensi						
Tinggi	3.37 in (85.6 mili meter)	3.37 in (85.6 mili meter)	2.55 in (65 mili meter)	3.37 in (85.6 mili meter)	3.37 in (85.6 mili meter)	2.12 in (53.98 mili meter)
Lebar	2.22 in (56.5 mili meter)	2.22 in (56.5 mili meter)	2.22 in (56.5 mili meter)	2.22 in (56.5 mili meter)	2.22 in (56.5 mili meter)	3.37 in (85.6 mili meter)
Tingkatan Daya		1.13 Amper @5V	200 mili Amper	1.34 A @5V	800 mili Amper	700 mili Amper @5V
Sumber Daya	USB-C	microUSB, GPIO	microUSB atau GPIO	microUSB atau GPIO	microUSB atau GPIO	microUSB atau GPIO

Pada penelitian ini nantinya akan menggunakan SBC Raspberry Pi model 4B karena sangat cocok digunakan untuk sistem yang menerapkan *deep learning*.

2.10. Google Colab

Google Colaboratory atau Google Colab merupakan tools baru yang dikeluarkan oleh Google Internal Research untuk membantu para Researcher untuk mengelola data dalam keperluan belajar atau pun melakukan eksperimen pada pengolahan data. Google Colab secara penggunaan serupa dengan Jupyter Notebook dan dibuat atas lingkungan Jupyter yang tidak membutuhkan pengaturan terlebih dahulu sebelum digunakan sepenuhnya pada Cloud yaitu dengan memanfaatkan media penyimpanan Google Drive [22].

2.11. Python

Python merupakan bahasa pemrograman yang bermodelkan skrip (scripting language) dengan orientasi objek. Python juga dapat digunakan dalam berbagai keperluan pengembangan software serta dapat berjalan dalam bermacam platform sistem operasi. Python sendiri merupakan bahasa pemrograman yang freeware atau dalam arti sebenarnya yaitu perangkat bebas, tidak ada batas dalam menyalin atau pendistribusian. Lengkap dengan sumber kode, debugger dan profiler, interface yang ada di dalamnya untuk pelayanan interface, fungsi sistem, GUI (antarmuka pengguna grafis), dan basis datanya. (Triasanti, 2010) [23]

Python adalah bahasa pemrograman yang mendukung pemrograman dengan orientasi objek. Beberapa fitur yang ditawarkan Python seperti Pustaka yang luas dimana terdapat modul-modul untuk bermacam keperluan, mendukung pemrograman dengan orientasi objek, terdapat tata bahasa yang mudah dipelajari, terdapat sistem pengelolaan memori secara otomatis, arsitektur dapat dikembangkan dan ditanam dalam bahasa pemrograman lain, misal object oriented Python juga dapat digabungkan dengan modul yang dibuat dengan bahasa pemrograman C++ [23].

2.12. Bot Telegram

Telegram adalah satu-satunya aplikasi pesan singkat yang menyediakan API untuk pengguna agar dapat membangun bot yang bisa dimanfaatkan untuk sistem informasi. Bot Telegram adalah aplikasi pihak ketiga yang bisa dijalankan di dalam Telegram. Penggunaanya juga dapat mengontrol bot menggunakan HTTPS ke API telegram [24].

Berikut merupakan kegunaan bot telegram yaitu dapat digunakan sebagai koran pintar yang bisa mengirimkan berita kepada pengguna bot tersebut, dapat digunakan sebagai penyambung layanan lain (seperti Gmail, Gambar, GIF, IMDB, Wiki, Musik, Youtube, GitHub), juga dapat digunakan untuk menerima pembayaran dari pengguna telegram dimana Bot menawarkan layanan kerja sebagai etalase virtual, dapat pula digunakan sebagai alat khusus (seperti pemberian peringatan, informasi ramalan cuaca, terjemahan bahasa, pembuatan format, atau layanan lainnya), juga digunakan sebagai permainan, baik pemain

tunggal atau pun pemain bersama, dan terakhir dapat digunakan sebagai layanan sosial dengan menghubungkan orang yang mencari mitra percakapan berdasarkan keinginan atau minat yang sama.

2.13. Penelitian Terdahulu

Pada tabel 2.5 akan dideskripsikan penelitian-penelitian yang telah dilakukan tentang Retinopati Diabetik, Raspberry Pi, dan *Deep Learning*.

Tabel 2.4 Tabel Penelitian Terdahulu

No	Nama Peneliti	Bentuk Publikasi	Abstrak
1.	Wulandari C.D. R., Suryo A.W., Ledya N.	Skripsi dan Jurnal e- <i>Proceeding of Engineering</i>	Pada jurnal dengan judul “Klasifikasi Diabetes Retinopati menggunakan Metode <i>Statistical Region Merging</i> dan <i>Convolutional Neural Network</i> ” ini, penelitian dilakukan menggunakan <i>deep learning</i> untuk mendapatkan klasifikasi RD dengan 4 tingkat keparahan yaitu <i>non-RD</i> , <i>mild</i> , <i>moderate</i> , dan <i>serve Proliferatif RD</i> . <i>Deep learning</i> yang digunakan yaitu algoritma CNN dan segmentasi SRM. Di mana objek dari penelitian ini adalah citra fundus retina mata yang di- <i>input</i> -kan menggunakan dataset citra yang sudah ada. Untuk mempermudah proses klasifikasi dari RD pada penelitian ini, digunakan <i>pre-processing</i> yaitu <i>crop</i> dan <i>resize</i> . Penelitian ini hanya berfokus pada keakurasian dari klasifikasi 4 tingkat keparahan RD. Hasil yang didapat pada penelitian ini yaitu dengan menggunakan algoritma CNN dan segmentasi SRM, tingkat keakurasian klasifikasi dari 4 tingkat keparahan RD mencapai 81,25%.

Pada jurnal [3], memiliki metode klasifikasi 4 tingkat keparahan RD menggunakan *deep learning*, yaitu klasifikasi CNN dan segmentasi SRM. Penelitian pada jurnal [3] menggunakan *deep learning* untuk melakukan klasifikasi, hanya saja masih menggunakan dataset terbatas. Penelitian ini akan mengadaptasi metode dari jurnal [3] dengan penambahan yaitu melakukan

diagnosa dini RD menggunakan *deep learning* dengan dataset yang lebih banyak dari penelitian sebelumnya dan pengiriman hasil diagnosa RD ke *smartphone* pengguna.



BAB III PERANCANGAN SISTEM

3.1. Analisis Kebutuhan Sistem

Pada bagian ini akan dijelaskan kebutuhan sistem secara keseluruhan. Agar dapat memudahkan dalam memahami kebutuhan sistem, digunakan beberapa analisis kebutuhan, diantaranya yaitu :

3.1.1. Kebutuhan Fungsional Sistem

Kebutuhan fungsional adalah kebutuhan yang berisi fungsi-fungsi pada sistem agar mampu mencapai tujuan yang telah ditetapkan. Fungsi-fungsi yang dibutuhkan adalah sebagai berikut:

- 1) Sistem dapat melakukan *pre-processing* pada citra fundus retina mata berupa *cropping*, *resize*, *masking* pada dataset IDRiD agar memiliki spesifikasi yang sama dengan dataset Messidor.
- 2) Sistem dapat melakukan segmentasi pada dataset dengan menggunakan segmentasi RAG Threshold
- 3) Sistem dapat memberikan klasifikasi berupa terhadap dataset yang sesuai dengan pengklasifikasian objek menggunakan algoritma CNN.
- 4) Sistem dapat memberikan informasi dan indikasi proses pada sistem yang sedang berlangsung melalui *buzzer*, LED merah, LED kuning, LED hijau dan LCD 16×2 .
- 5) Sistem dapat mengirimkan hasil diagnosa dini RD ke aplikasi *mobile* pengguna.

3.1.2. Kebutuhan Non-Fungsional Sistem

Kebutuhan non-fungsional meliputi kebutuhan pendukung atau kebutuhan yang menyangkut kepada sifat dari sistem agar dapat mencapai tujuan yang telah ditetapkan. Kebutuhan non-fungsional pada sistem ini, yaitu:

- 1) *Performance*, meliputi akurasi, presisi, dan respon waktu.
- 2) *Portability*, yaitu kemampuan sistem agar mampu dibawa kemana pun dengan ukuran yang bisa disimpan dan diletakkan di mana pun.

- 3) *Usability*, yaitu kemampuan sistem untuk memberikan indikasi sistem dan pengiriman hasil diagnosa dini RD ke *smartphone* pengguna.

3.1.3. Kebutuhan Perangkat Keras

Pada sistem yang akan dibangun, piranti perangkat keras yang dibutuhkan untuk mencapai tujuan yang telah ditetapkan adalah sebagai berikut:

- 1) Raspberry Pi 4 *model* 4B 4 GB yang digunakan sebagai kontroler untuk memberikan dan memproses segala jenis instruksi dengan *output* berupa diagnosa dini RD.
- 2) SD Card *Extreme* 64 GB *Speed* 150 MB yang digunakan untuk penyimpanan pada sistem.
- 3) *Smartphone* yang akan digunakan untuk menampilkan hasil diagnosa dini RD melalui aplikasi *mobile*.
- 4) Lampu LED merah, kuning, hijau, *buzzer* dan LCD 16 × 2 yang akan digunakan sebagai indikator.

3.1.4. Kebutuhan Perangkat Lunak

Pada sistem yang akan dibangun, perangkat lunak yang dibutuhkan untuk mencapai tujuan yang telah ditetapkan adalah sebagai berikut:

3.1.4.1. Kebutuhan Perangkat Lunak pada *Personal Computer (PC)/Cloud*

Perancangan perangkat lunak pada PC/cloud berupa program *training* dan *testing*. Program dirancang menggunakan bahasa pemrograman python pada Google Colaboratory (Google Colab). Program akan menggunakan 176 data sampel citra 2D fundus retina mata sebagai dataset *training* dengan pembagian yaitu 100 data *train* dan 52 data validasi.

3.1.4.2. Kebutuhan Perangkat Lunak pada Sistem Tertanam

Perancangan perangkat lunak pada sistem tertanam terdiri dari beberapa tahap, diantaranya sebagai berikut:

- 1) *Pre-processing*, pada tahap ini citra 2D fundus retina mata didapat dari *input* kamera NoIR v2 Raspberry Pi dan dilakukan proses *cropping* dan *resize* resolusi untuk menyamakan ukuran citra.

- 2) Klasifikasi menggunakan algoritma CNN, pada tahap ini citra 2D diklasifikasikan yaitu mata Normal atau mata dengan prediksi RD yang dijadikan sebagai hasil diagnosa dini RD.
- 3) Pengiriman hasil diagnosa dini ke Bot Telegram dengan menggunakan *library* telebot.

3.1.4.3. Kebutuhan Perangkat Lunak pada Bot Telegram

Perancangan perangkat lunak pada bot telegram berupa pesan singkat yang akan mengirimkan dan menampilkan hasil dari diagnosa dini RD pada sistem. Pada bot telegram akan digunakan *interface* yang sudah disediakan pada aplikasi telegram melalui fitur Bot Father pada telegram. Pada program untuk membangun Bot Telegram, digunakan tiga menu, yaitu /mulai, /hasil, dan /foto. Penjelasan menu pada aplikasi Telegram adalah sebagai berikut:

- /mulai : ketika mengirim pesan /mulai pada Telegram, maka Bot Telegram akan mengirimkan pesan pembuka dan penjelasan mengenai sistem serta penjelasan mengenai menu yang dapat dijalankan pada Bot Telegram RD Diagnosis.
- /hasil : ketika mengirim pesan /hasil pada Telegram, maka Bot Telegram akan mengirimkan pesan berupa hasil yang didapat dari klasifikasi keseluruhan citra 2D.
- /foto : ketika mengirim pesan /foto pada Telegram, maka Bot Telegram akan mengirimkan citra 2D, yaitu tiga citra 2D beserta pesan berupa masing-masing nama foto.

3.1.5. Kebutuhan Data

Kebutuhan data pada sistem yaitu data berupa citra 2D fundus retina mata dengan label data dibagi menjadi Normal, *Mild NDPR*, *Moderate NDPR*, atau *Severe NDPR*. Pada masing-masing dataset memiliki jumlah yang berbeda-beda seperti pada tabel 3.1 sistematika data berikut.

Tabel 3.1 Sistematika Data

Kelas	Jumlah Data					
	Messidor			IDRiD		
	Train	Validation	Test	Train	Validation	Test
Normal	16	3	1	9	10	5
Mild NPDR	16	3	1	9	10	5
Moderate NPDR	16	3	1	9	10	5
Severe NPDR	16	3	1	9	10	5
Jumlah	64	12	4	27	40	20
	80			87		

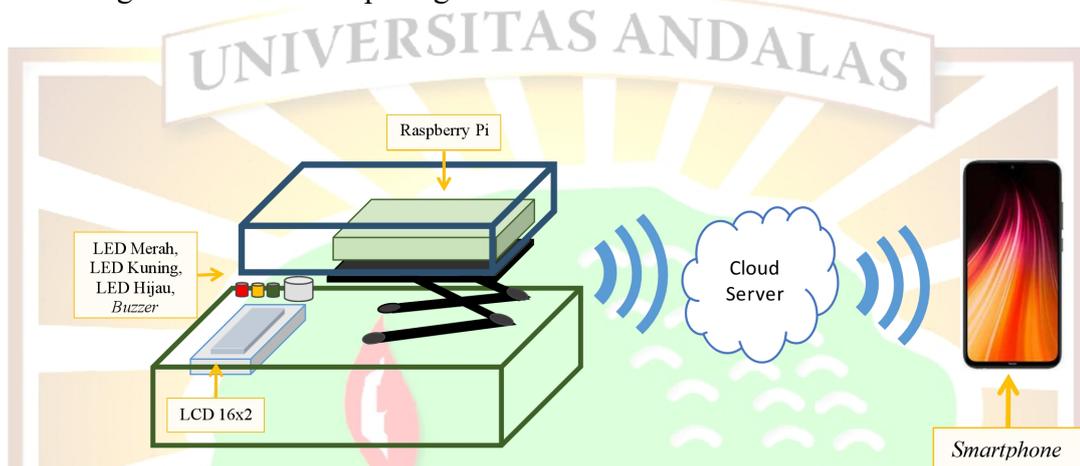
Klasifikasi yang menggunakan CNN terbagi menjadi dua proses yaitu *training* dan *testing*. Pengkategorian awal data pada dataset berupa data Normal, Mild NPDR, Moderate NPDR, dan Severe NPDR dikategorikan menggunakan nama seperti ‘Normal.’, ‘Mild.’, ‘Moderate.’, dan ‘Severe.’. Dataset yang digunakan pada proses *training* dibagi menjadi tiga jenis dataset dengan penjelasan pada tabel 3.2 Pembagian Jenis Dataset sebagai berikut.

Tabel 3.2. Pembagian Jenis Dataset

Dataset	Penjelasan
Dataset A	Dataset Messidor dan IDRiD yang telah melalui proses <i>pre-processing</i> .
Dataset B	Dataset Messidor dan IDRiD yang telah melalui proses <i>pre-processing</i> , <i>masking</i> dan segmentasi label 1.
Dataset C	Dataset Messidor dan IDRiD yang telah melalui proses <i>pre-processing</i> , <i>masking</i> dan segmentasi label 2.

3.2. Rancangan Umum Sistem

Untuk mencapai tujuan yang di inginkan dari sistem, maka dibangun sistem dengan kebutuhan fungsional dan non-fungsional yang telah di tetapkan sebelumnya. Adapun rancangan umum sistem yang akan dibangun terdapat pada Rancangan Umum Sistem pada gambar 3.2 berikut.



Gambar 3.1. Rancangan Umum Sistem

Perancangan umum sistem dilakukan dengan menggambarkan rancangan perangkat keras sistem secara keseluruhan, yaitu dengan susunan sebagai berikut:

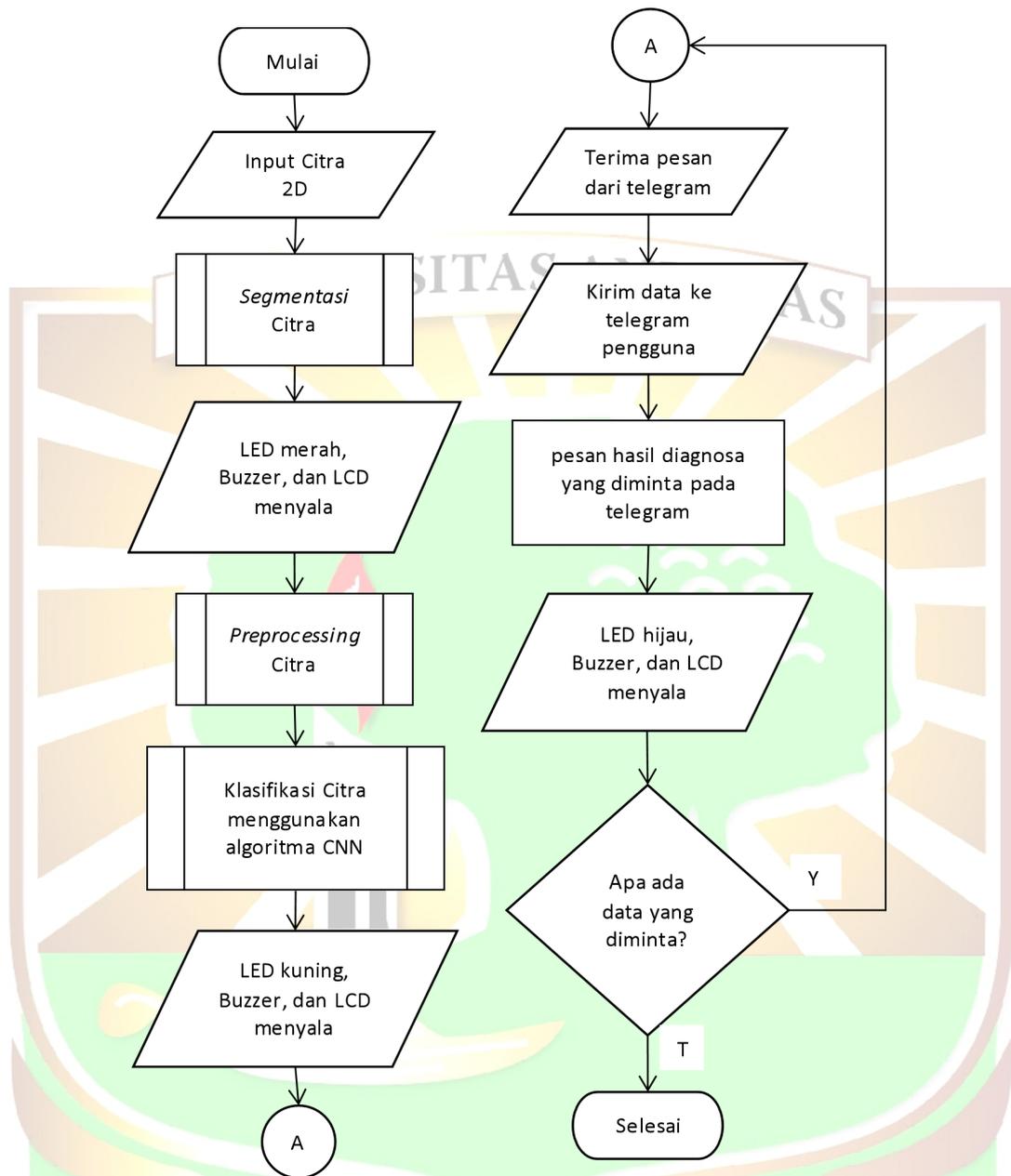
- 1) Input dataset kedalam sistem sebagai data yang akan didiagnosa
- 2) Setelah itu, Raspberry Pi akan mengolah citra melalui proses *pre-processing* citra.
- 3) LED merah akan menyala dan *buzzer* akan berbunyi satu kali sesaat setelah proses *pre-processing* citra selesai.
- 4) Kemudian, LCD menampilkan pernyataan bahwa citra telah selesai melalui tahap *pre-processing* citra.
- 5) Setelah itu, Raspberry Pi akan melalui proses klasifikasi citra hingga mendapatkan diagnosa dini RD.
- 6) LED kuning akan menyala dan *buzzer* akan berbunyi dua kali apabila proses klasifikasi citra telah selesai dan didapat hasil diagnosa dini RD.
- 7) Kemudian, LCD menampilkan pernyataan bahwa klasifikasi telah selesai dan didapat hasil diagnosa dini RD.
- 8) Hasil diagnosa RD akan dikirimkan ke aplikasi *mobile* apabila terdapat pesan yang didapat oleh sistem dari aplikasi *mobile*.

- 9) LED hijau akan menyala dan *buzzer* akan berbunyi tiga kali apabila pesan *feedback* telah diterima oleh aplikasi *mobile*.
- 10) Kemudian, LCD menampilkan pernyataan bahwa hasil diagnosa dini RD telah dikirimkan ke aplikasi *mobile*.

3.3. Rancangan Proses

Pada sistem yang dibangun, terdapat lima tahap dalam diagnosa dini RD menggunakan IDRiD. Pertama, proses input citra yang sudah ada pada Raspberry Pi ke dalam program. Selanjutnya, citra yang telah diinput akan melalui tahap *pre-processing*. Kemudian, citra yang telah selesai melalui tahap *pre-processing* akan melalui proses segmentasi menggunakan segmentasi *RAG Thresholding*. Selanjutnya, citra hasil segmentasi akan melalui proses klasifikasi dengan algoritma CNN. Terakhir, hasil klasifikasi yang merupakan hasil diagnosa dini dikirim ke aplikasi *mobile*. Alur fungsi sistem dapat dilihat pada *flowchart* Rancangan Proses berikut:





Gambar 3.2. Flowchart Rancangan Proses

Gambar 3.2 merupakan alur kerja sistem secara keseluruhan. Proses dimulai dengan melalui *input* data citra ke program. Kemudian, data citra akan melalui *pre-processing* citra. Setelah itu LED merah, *buzzer* dan LCD akan memberitahukan bahwa data citra telah diambil dan disimpan pada penyimpanan internal sistem. Kemudian, data citra hasil *pre-processing* akan melalui proses segmentasi menggunakan segmentasi *RAG Thresholding* dan proses klasifikasi menggunakan algoritma CNN, setelah itu LED kuning, *buzzer* dan LCD akan memberitahukan bahwa proses segmentasi dan klasifikasi telah selesai. Sistem

kemudian akan menerima pesan dari aplikasi *mobile* untuk mengirimkan hasil klasifikasi yang merupakan diagnosa dini RD. Setelah pesan terkirim, LED hijau, *buzzer* dan LCD akan memberi indikasi bahwa hasil diagnosa dini RD telah dikirim ke Telegram pengguna.

3.3.1. Rancangan Perangkat Keras

Rancangan perangkat keras secara keseluruhan dapat dilihat pada gambar 3.4 berikut:



Gambar 3.3. Rancangan Perangkat Keras

Pada gambar 3.3. diatas perlu diketahui bahwa fungsi dari perangkat keras yang digunakan adalah sebagai berikut:

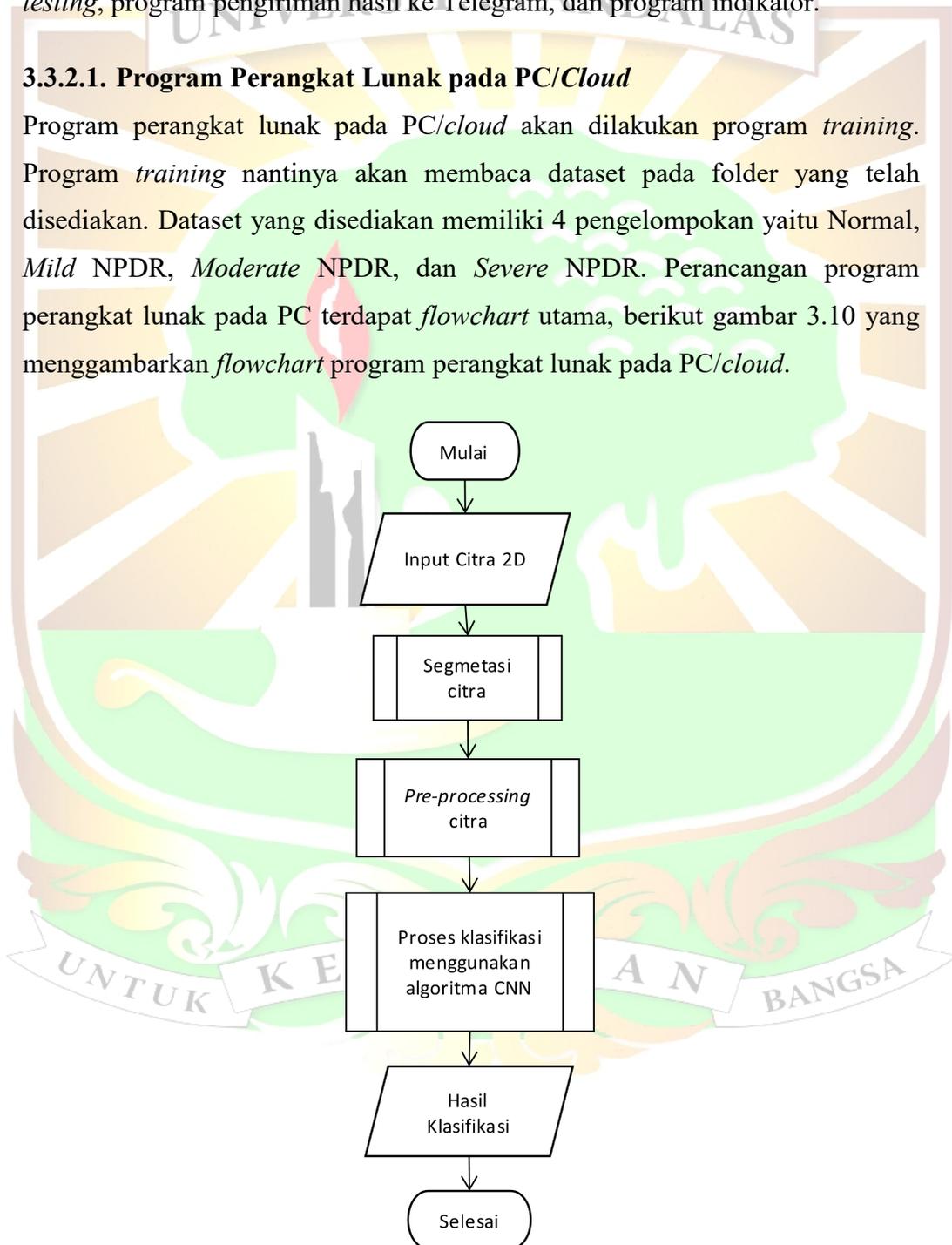
1. Raspberry Pi 4 B, berperan sebagai kontroler dan pemrosesan sistem untuk mendapatkan klasifikasi berupa hasil diagnosa dini RD dengan memanfaatkan algoritma CNN dan segmentasi RAG Threshold.
2. *Smartphone*, berperan sebagai media penampilan *output* dari sistem yaitu data hasil diagnosa RD serta citra 2D fundus retina mata.
3. LCD 16×2 , berperan sebagai perangkat keluaran berupa kalimat yang mewakili sistem pada setiap proses sistem.
4. LED Merah, Kuning, Hijau, dan *Buzzer* berperan sebagai keluaran berupa indikator dari berjalannya sebuah proses tertentu.

3.3.2. Rancangan Perangkat Lunak

Perancangan perangkat lunak pada penelitian ini meliputi program perangkat lunak pada *Personal Computer (PC)/cloud* dan pada sistem tertanam. Pada *PC/cloud* meliputi program *pre-processing*, segmentasi, *training* dan *testing*. Pada sistem tertanam meliputi program *pre-processing*, program segmentasi, program *testing*, program pengiriman hasil ke Telegram, dan program indikator.

3.3.2.1. Program Perangkat Lunak pada *PC/Cloud*

Program perangkat lunak pada *PC/cloud* akan dilakukan program *training*. Program *training* nantinya akan membaca dataset pada folder yang telah disediakan. Dataset yang disediakan memiliki 4 pengelompokan yaitu Normal, *Mild NPDR*, *Moderate NPDR*, dan *Severe NPDR*. Perancangan program perangkat lunak pada *PC* terdapat *flowchart* utama, berikut gambar 3.10 yang menggambarkan *flowchart* program perangkat lunak pada *PC/cloud*.

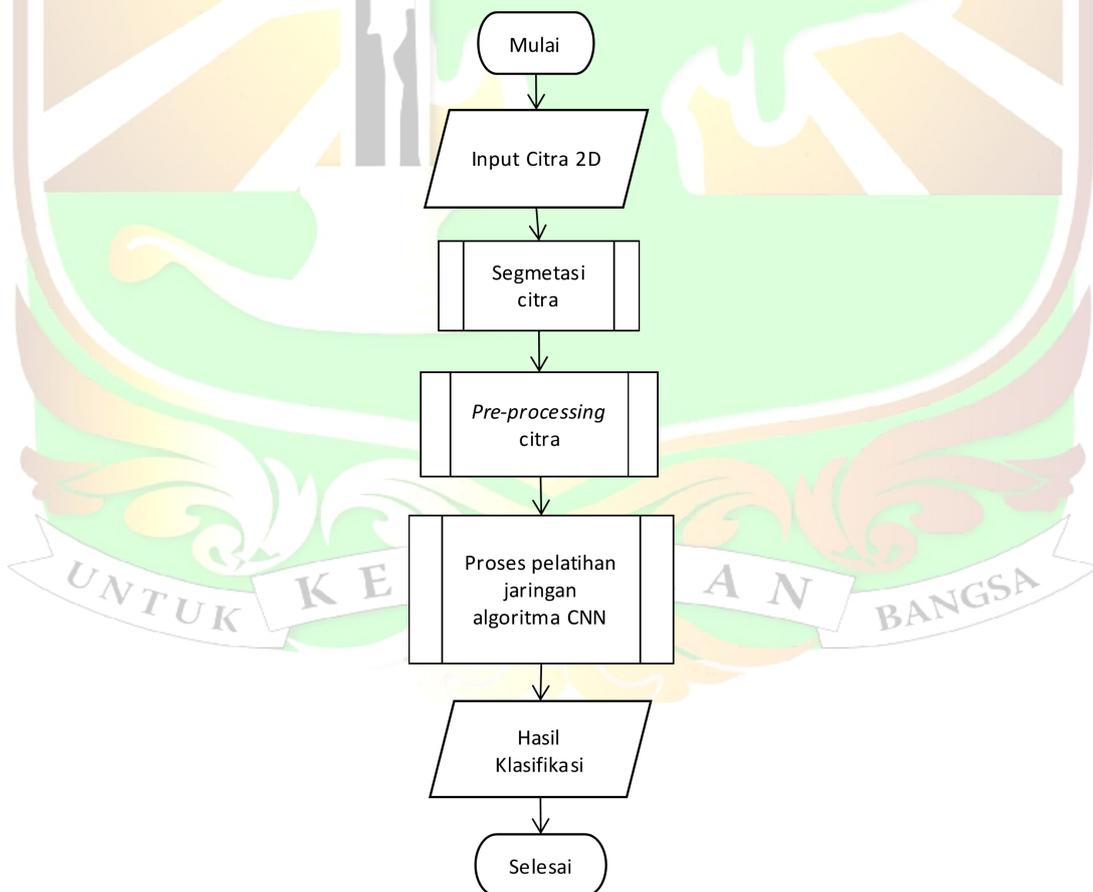


Gambar 3.4. Flowchart Program Perangkat Lunak pada *PC/Cloud*

Pada Gambar 3.4. merupakan alur program yang akan berjalan pada PC. Tahap awal setelah meng-*input*-kan citra 2D adalah tahap *pre-processing*. Tahap *Pre-processing* dilakukan agar mempermudah mendapatkan hasil diagnosa yang presisi dengan input citra berbeda. Proses selanjutnya pada citra yaitu tahap segmentasi yang terdiri dari dua jenis segmentasi dengan menerapkan segmentasi *RAG Thresholding*. Proses berikutnya yaitu tahap klasifikasikan yang terdiri dari proses *training* dan proses *testing* dengan menggunakan algoritma CNN untuk mendapatkan hasil diagnosa dini RD. Hasil klasifikasi RD akan menunjukkan pada tingkatan mana data citra 2D dikelompokkan.

3.3.2.2. Program Perangkat Lunak pada Sistem Tertanam

Program perangkat lunak pada sistem tertanam akan dilakukan program *testing*. Program *testing* nantinya akan dilakukan langsung oleh pengguna sistem. Data input yang akan di program berupa citra 2D yang telah dipisahkan dari data untuk proses *testing* pada PC/Cloud.

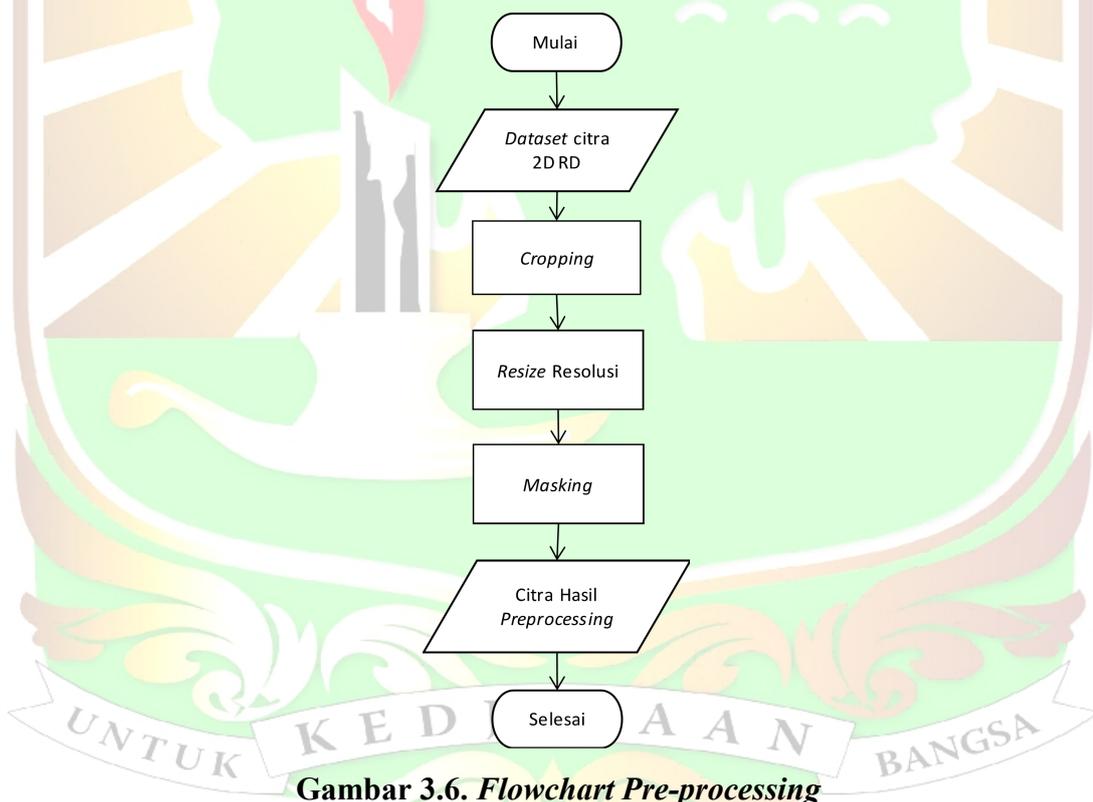


Gambar 3.5. Flowchart Program Perangkat Lunak pada Sistem Tertanam

Pada program perangkat lunak yang dilakukan pada sistem tertanam, data di-*input*-kan dari data citra 2D, kemudian sistem akan melakukan proses *testing* menggunakan data citra 2D. Proses *training* hanya dilakukan di PC/Cloud dengan menyimpan model dari proses *training*, kemudian proses *testing* yang dilakukan di sistem tertanam hanya menggunakan model yang telah didapat dari proses *training* pada PC/Cloud.

3.3.2.3. Program *Pre-processing*

Pada awal program dilakukan *pre-processing* citra yang telah disediakan. *Pre-processing* dilakukan untuk meningkatkan kualitas citra sehingga citra yang diolah mendapatkan hasil yang maksimal saat melalui tahap klasifikasi CNN. Berikut *flowchart* tahap *pre-processing* pada gambar 3.6.



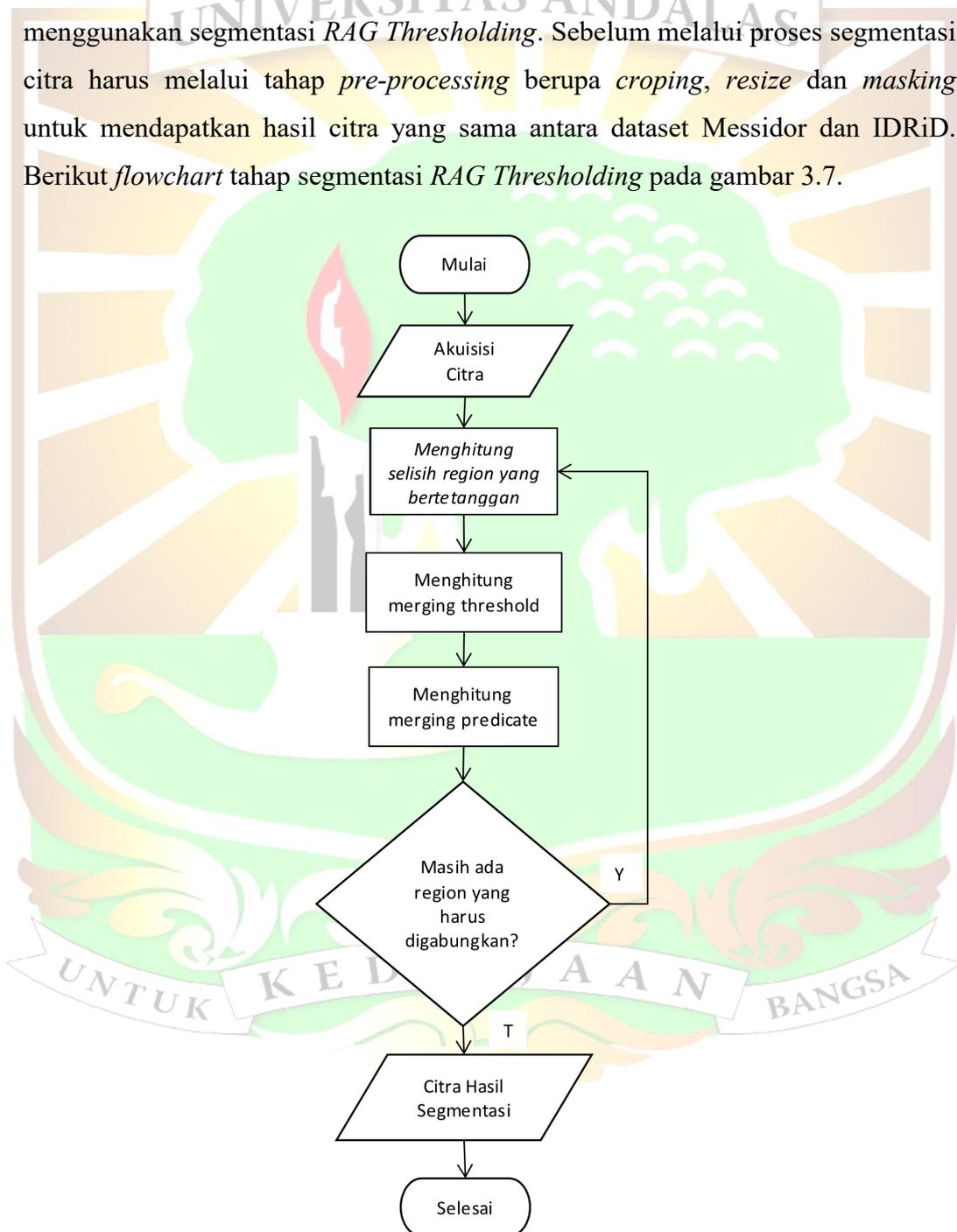
Gambar 3.6. Flowchart *Pre-processing*

Pada program *pre-processing*, dilakukan pemotongan gambar dan pengaturan ukuran citra menjadi 227×227 piksel. Selanjutnya, citra yang telah disesuaikan ukuran dan besarnya kemudian dinaikkan kualitas resolusinya menjadi 300×300 dpi melalui tahap *pre-processing*. Tahap terakhir, citra 2D akan melalui tahap

masking dengan memanfaatkan citra berwarna hitam menjadi *mask ellipse* agar citra 2D memiliki yang sama antar data pada dataset.

3.3.2.4. Program Segmentasi Citra

Proses segmentasi bertujuan untuk membagi citra menjadi region yang homogen atau memiliki kriteria yang sama. Segmentasi citra pada penelitian ini menggunakan segmentasi *RAG Thresholding*. Sebelum melalui proses segmentasi, citra harus melalui tahap *pre-processing* berupa *cropping*, *resize* dan *masking* untuk mendapatkan hasil citra yang sama antara dataset Messidor dan IDRiD. Berikut *flowchart* tahap segmentasi *RAG Thresholding* pada gambar 3.7.

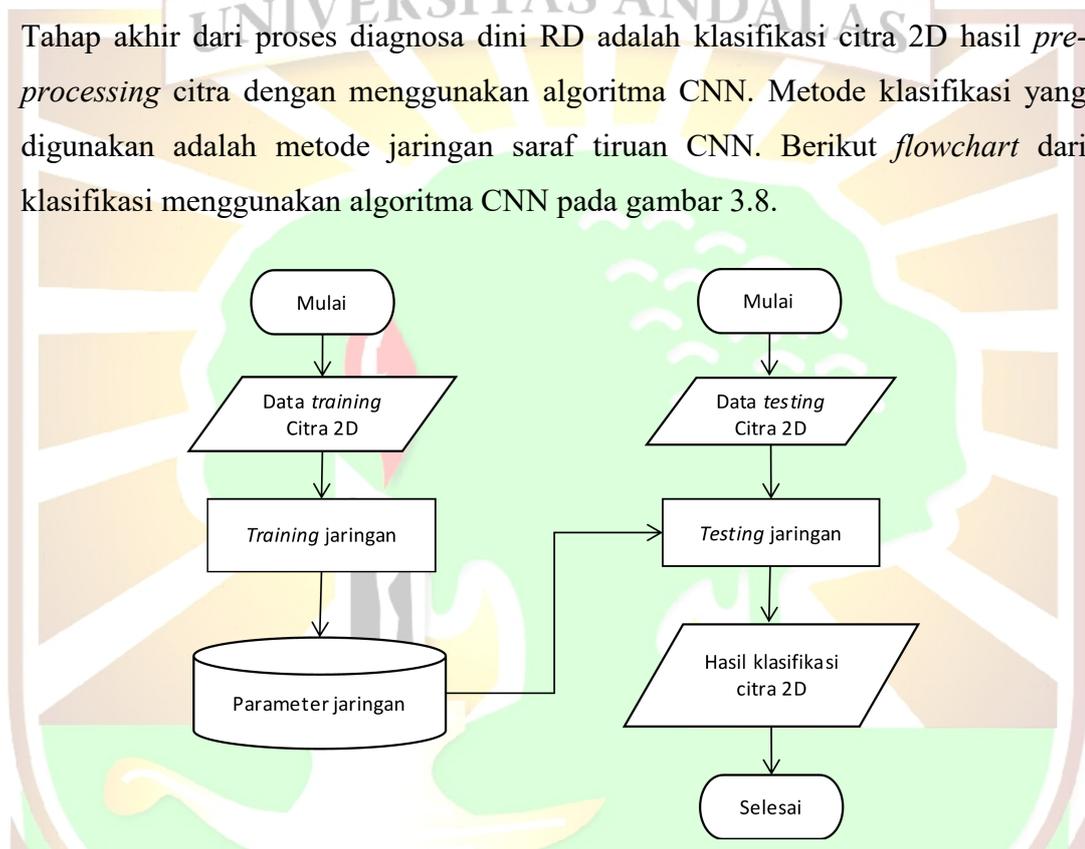


Gambar 3.7. Flowchart Segmentasi *RAG Thresholding*

Pada program segmentasi, dilakukan akuisisi citra, kemudian dilakukan selisih region yang bertetangga. Selanjutnya, dihitung *merging threshold* dan *merging predicate* untuk proses segmentasi nilai piksel yang bertetangga dan didapatkanlah hasil citra segmentasi.

3.3.2.5. Program Klasifikasi

Tahap akhir dari proses diagnosa dini RD adalah klasifikasi citra 2D hasil *pre-processing* citra dengan menggunakan algoritma CNN. Metode klasifikasi yang digunakan adalah metode jaringan saraf tiruan CNN. Berikut *flowchart* dari klasifikasi menggunakan algoritma CNN pada gambar 3.8.



Gambar 3.8. Flowchart Klasifikasi [3]

Pada program klasifikasi, dimulai dengan data masukan dari dataset, kemudian dibangun jaringan *training* agar mendapat parameter pembeda. Selanjutnya, setelah parameter jaringan *training* telah terkumpul, dimulailah proses *testing* dengan memasukkan data *testing* dari dataset. Kemudian, digunakan parameter jaringan *training* untuk membangun jaringan *testing*. Setelah dibentuk jaringan *testing*, didapatkanlah hasil klasifikasi dari masukan data *testing*.

3.3.2.6. Program Pelatihan Jaringan CNN

Tahap pelatihan merupakan tahapan di mana jaringan dilatih menggunakan data latih untuk memperoleh akurasi yang tinggi dari klasifikasi yang

dilakukan. Tahap ini terdiri dari *transfer learning*, *feedforward* dan *backpropagation*. Berikut *flowchart* tahap pelatihan jaringan CNN pada gambar 3.9.

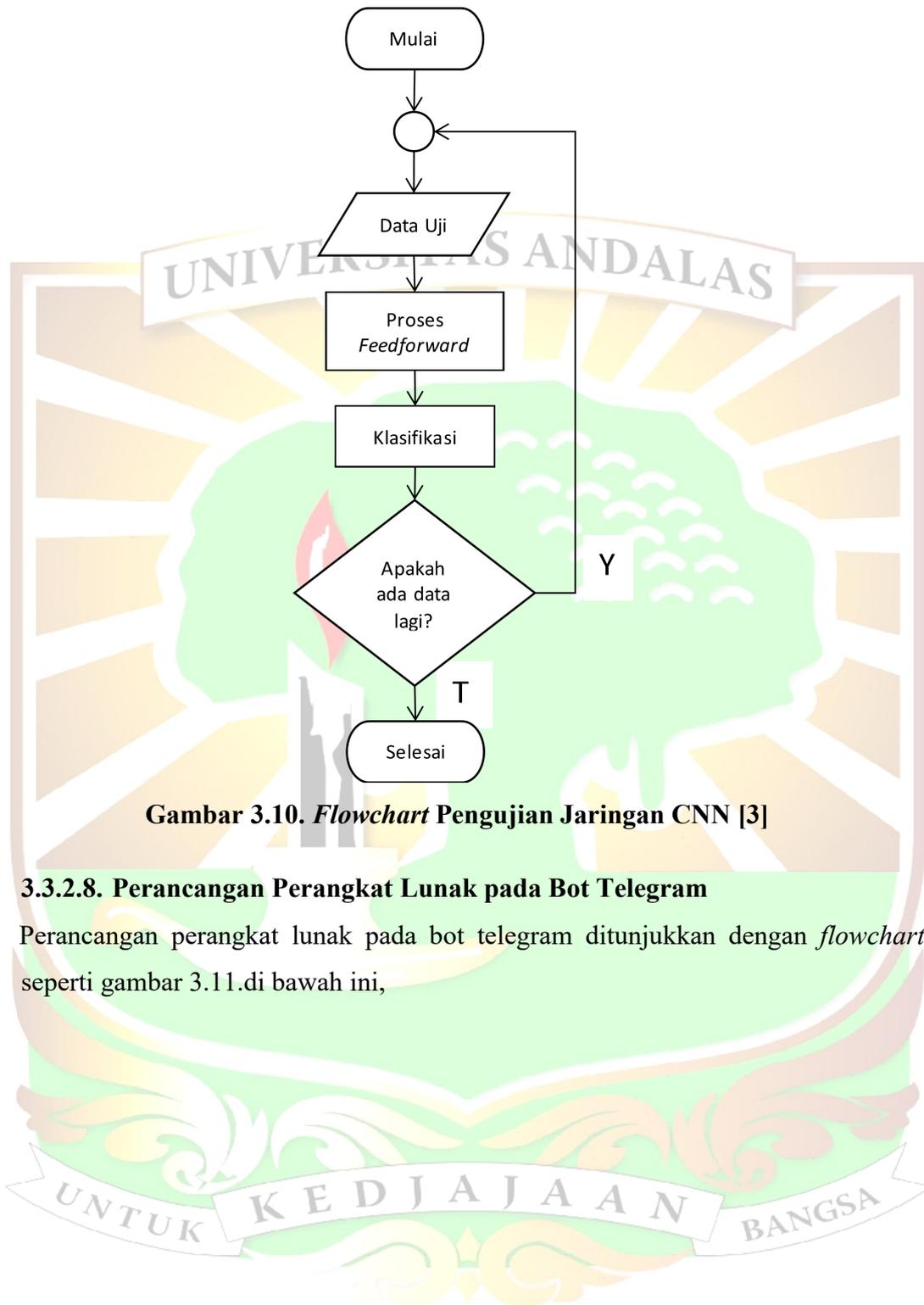


Gambar 3.9. Flowchart Pelatihan Jaringan CNN [3]

Pada program pelatihan jaringan, dilakukan proses transfer learning dari nilai yang diinputkan berupa nilai 0.001 atau 0.0001 atau yang lainnya, kemudian dilakukan proses *feedforward* yaitu dengan melalui seluruh tahapan model sesuai susunan dan data masukan. Setelah itu, dilakukan proses *backpropagation* yaitu dengan melakukan seluruh tahapan model kembali untuk mengetahui data dan nilai yang salah. Proses terakhir yaitu menghitung bobot dan bias yang didapat dari data dan nilai error selama proses *backpropagation*.

3.3.2.7. Program Pengujian Jaringan CNN

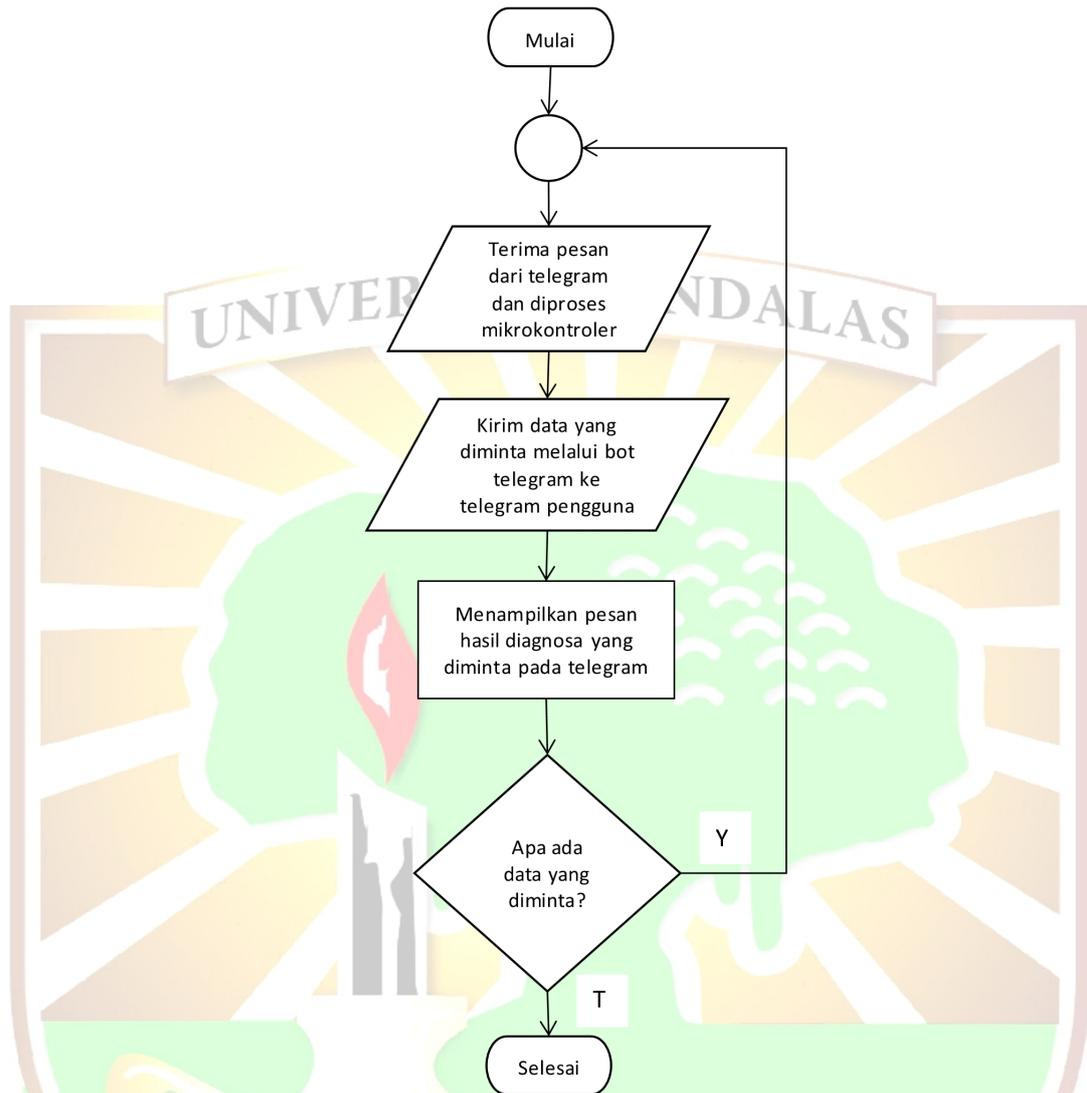
Tahap pengujian jaringan CNN menggunakan bobot dan bias dari hasil proses pelatihan. Tahap ini memiliki kesamaan dengan tahap pelatihan, namun tidak memiliki proses *feedforward* dan *backpropagation*. Berikut *flowchart* tahap pengujian jaringan CNN pada gambar 3.10.



Gambar 3.10. Flowchart Pengujian Jaringan CNN [3]

3.3.2.8. Perancangan Perangkat Lunak pada Bot Telegram

Perancangan perangkat lunak pada bot telegram ditunjukkan dengan *flowchart* seperti gambar 3.11. di bawah ini,



Gambar 3.11. Flowchart Rancangan Perangkat Lunak pada Bot Telegram

Pada Gambar 3.11. merupakan alur kerja dari Bot Telegram yang berfungsi sebagai pengiriman hasil diagnosa dini RD dari Raspberry Pi ke *smartphone* pengguna. Bot telegram akan mengirimkan pesan data yang akan dikirimkan ke *smartphone* pengguna ke mikrokontroler, kemudian mikrokontroler mengirimkan data yang diminta oleh pengguna melalui bot telegram. Kemudian data yang telah di terima akan disimpan sebagai pesan singkat dan dokumen berupa foto pada telegram pengguna.

3.4. Rencana Pengujian

Rencana pengujian merupakan rencana yang akan dilakukan untuk memperoleh data dari perangkat keras dan perangkat lunak, serta fungsional sistem secara keseluruhan pada saat sistem telah selesai di implemantasikan.

3.4.1. Pengujian Perangkat Lunak

Program perangkat lunak yang dibutuhkan dalam pembuatan sistem adalah bahasa pemrograman Python pada Raspberry Pi dan Google Colab pada PC/Cloud. Rancangan pengujian yang akan dilakukan untuk menguji perangkat lunak pada sistem terdapat pada Tabel 3.2. berikut:

Tabel 3.3. Rencana Pengujian Perangkat Lunak

No	Komponen pengujian	Rencana pengujian	Tujuan
1.	Program indikator sistem	Melakukan pengujian pada sistem terkait indikator yang akan diberikan oleh LED merah, LED kuning, LED hijau, <i>buzzer</i> dan LCD 16 × 2 berdasarkan penafsiran dari indikator tersebut.	Untuk menguji ketepatan dalam mengindikasi proses yang telah berlangsung sebagai hasil tampak.
2.	Program <i>pre-processing</i> citra 2D	Melakukan pengujian terhadap dataset yang telah melalui proses <i>cropping</i> , <i>resize</i> dan <i>masking</i> dataset	Untuk menentukan besar <i>cropping</i> , <i>resize</i> dan <i>masking</i> yang akan dilakukan terhadap citra 2D fundus retina mata.
3.	Program Segmentasi <i>RAG Thresholding</i>	Melakukan pengujian terhadap dataset yang telah melalui proses segmentasi <i>RAG Thresholding</i>	Untuk menentukan besar nilai <i>compactness</i> , <i>segment</i> dan <i>label</i> pada program segmentasi <i>RAG Thresholding</i>
4.	Program <i>Training</i>	Menguji nilai akurasi dari program CNN dalam melakukan <i>training</i> dataset	Untuk menguji akurasi program CNN yang digunakan

5.	Program <i>Testing</i>	Menguji keakuratan dari sistem dengan menggunakan <i>input</i> citra 2D fundus retina mata	Untuk menguji keakuratan hasil dari diagnosa citra 2D fundus retina mata
6.	Program pengiriman dan penampilan hasil diagnosa dini RD	Melakukan pengiriman dan penampilan hasil diagnosa dini RD pada aplikasi <i>mobile</i> berupa bot telegram	Untuk menguji bahwa sistem dapat mengirim hasil diagnosa dini RD dengan tepat sesuai hasil diagnosa dini RD yang diberikan oleh sistem

3.5. Pengujian Sistem

Pada tahap pengujian sistem, yang menjadi pembahasan utama adalah tindakan yang akan dilakukan dan data yang harus diperoleh dari perangkat keras dan perangkat lunak, serta bagaimana sistem secara keseluruhan sesuai dengan kebutuhan fungsional dan non-fungsional. Pengujian sistem secara keseluruhan akan menguji apakah sistem dapat menampilkan hasil diagnosa RD pada aplikasi *mobile* dengan klasifikasi citra 2D fundus menggunakan algoritma CNN. Untuk rencana pengujian sistem dapat dilihat pada tabel 3.3.

Tabel 3.4. Pengujian Sistem

No	Butir Uji	Indikator Keberhasilan
1.	Sistem memberikan diagnosa dini RD melalui citra 2D	Sistem melakukan klasifikasi RD terhadap citra 2D
2.	Sistem mengirimkan hasil diagnosa dini RD pada bot telegram	Bot telegram mengirimkan pesan terkait hasil diagnosa dini RD pada bot telegram pengguna
3.	Sistem memberikan tanda pada pengguna pada proses yang sedang berjalan pada sistem menggunakan indikator LED, <i>Buzzer</i> , dan LCD 16 × 2	LED, <i>buzzer</i> dan LCD 16 × 2 aktif pada waktu yang telah ditentukan untuk memberi informasi terkait proses pada sistem yang sedang berjalan

3.6. Analisis Kebutuhan Penelitian

Analisis kebutuhan penelitian, pada kebutuhan sistem terdapat dua bagian, yaitu kebutuhan perangkat keras dan kebutuhan perangkat lunak. Untuk kebutuhan yang dibutuhkan pada tiap-tiap perangkat diantaranya:

Tabel 3.5. Alat dan Bahan Penelitian

No.	Perangkat Keras	Perangkat Lunak	Data
1.	Raspberry Pi 4B	Bahasa pemrograman Python	Data <i>Training</i>
2.	<i>Smartphone</i>	Sistem operasi Raspberry Pi	Data <i>Validation</i>
3.	<i>SD Card</i>	Microsoft office	Data <i>Testing</i>
4.	<i>Buzzer</i>	Google Colab	
5.	LCD 16 × 2	Google Script	
6.	LED merah, kuning, hijau		
7.	Kabel jumper		



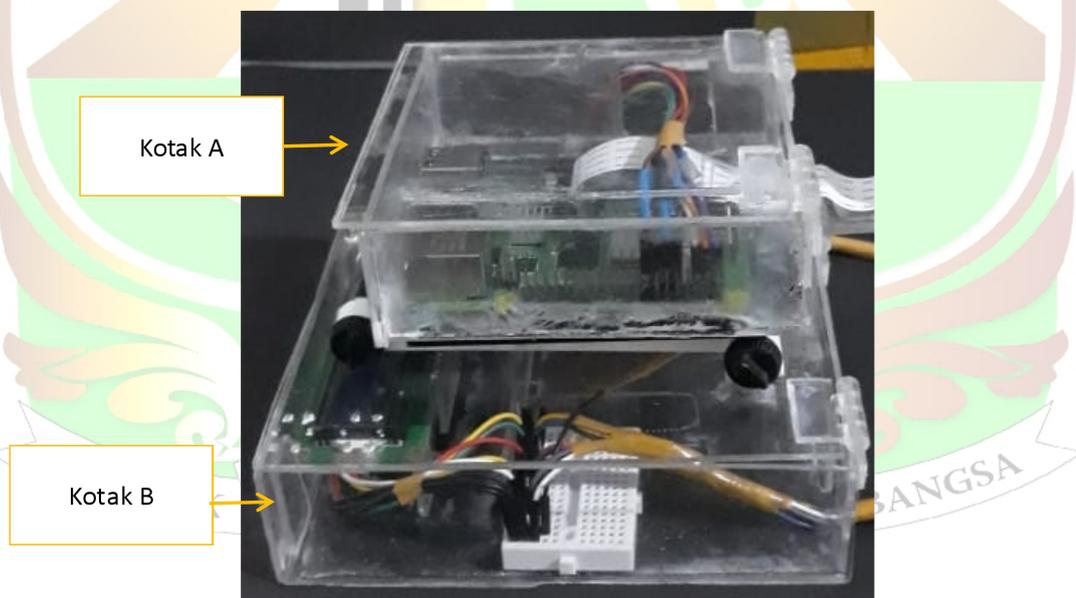
BAB IV IMPLEMENTASI DAN PENGUJIAN SISTEM

4.1. Implementasi

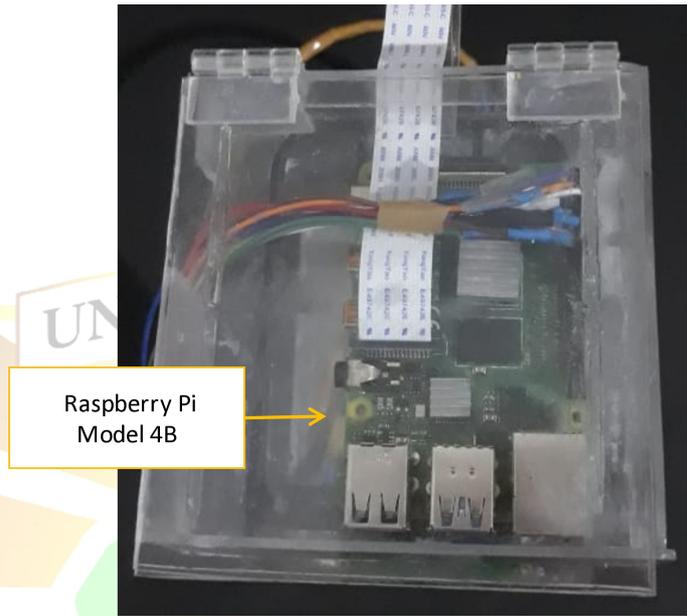
Tahap implementasi dan pengujian sistem adalah tahap untuk mengkaji dan menganalisis apakah sistem yang dibangun sudah sesuai dengan kriteria yang diinginkan atau sudah berjalan dengan baik sesuai dengan tujuan dari sistem ini dibuat. Agar proses pengkajian implementasi dan pengujian terstruktur, maka implementasi dan pengujian akan dibahas dengan membagi ke dalam tiga bagian yaitu, implementasi perangkat keras, implementasi perangkat lunak dan implementasi sistem.

4.1.1. Implementasi Perangkat Keras

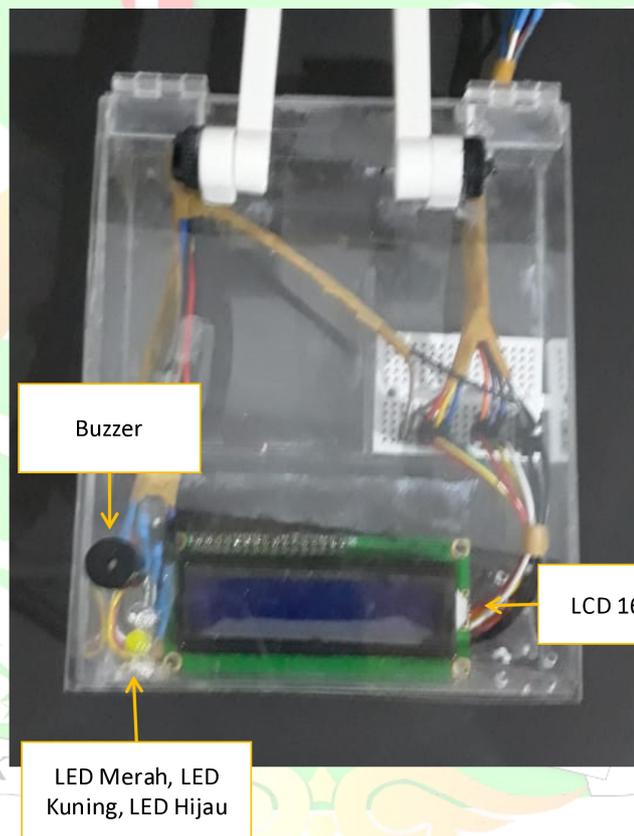
Implementasi perangkat keras adalah proses di mana proses perangkaian atau proses *wiring* dari komponen-komponen perangkat keras yang telah disesuaikan dengan kebutuhan perangkat keras yang akan membangun sistem.



Gambar 4.1 a). Perangkat Keras



Gambar 4.1 b). Kotak A



Gambar 4.1 c). Kotak B

Pada sistem yang dibangun memiliki panjang 10,6 cm lebar 10,3 cm dan tinggi 3,3 cm pada kotak A, dan panjang 15,3 cm, lebar 12,1 cm dan tinggi 4,1 cm pada kotak B. Sistem memiliki tinggi 10,9 cm ketika dirapatkan, tinggi 32,4 cm ketika

diregangkan, serta tempat modul kamera dengan tinggi 4 cm dari kotak A. Sehingga luas bangun sistem secara keseluruhan dengan kotak B sebagai luas alas yaitu $42,3 \text{ cm}^3$.

Perangkat yang diperlukan pada sistem adalah Raspberry Pi 4 model B, LCD 16×2 , *buzzer* dan LED. Untuk gambar dari perangkat keras yang digunakan dapat dilihat pada gambar 4.1. pada Gambar 4.1 komponen - komponen perangkat keras tersebut memiliki fungsi sebagai berikut:

1. Raspberry Pi 4 Model B, berfungsi sebagai kontroler yang mengatur semua proses dari sistem, Raspberry Pi juga tempat untuk melakukan proses klasifikasi citra menggunakan CNN. Spesifikasi utama Raspberry Pi 4 model B adalah sebagai berikut:

Processor : Quad Core Cortex-A72 (ARM v8)

RAM : 4GB

2. LCD 16×2 , berfungsi sebagai perangkat keluaran untuk menampilkan kegiatan yang telah dilakukan sistem.
3. LED, berfungsi sebagai perangkat keluaran yang memberikan notifikasi berupa nyala lampu dengan pembagian sebagai berikut:

LED Merah : Pengambilan citra

LED Kuning : Klasifikasi selesai

LED Hijau : Pengiriman Hasil

4. *Buzzer*, berfungsi sebagai perangkat keluaran yang memberikan notifikasi berupa bunyi dengan pembagian sebagai berikut:

Bunyi 1 kali : Pengambilan citra

Bunyi 2 kali : Klasifikasi selesai

Bunyi 3 kali : Hasil telah dikirim ke Telegram

4.1.2. Implementasi Perangkat Lunak

Implementasi perangkat lunak dan *dependency* diperlukan agar perangkat keras dapat bekerja dengan kebutuhan sistem. Adapun perangkat lunak yang diimplementasikan ke dalam sistem adalah sebagai berikut:

1. Untuk proses klasifikasi citra 2D fundus retina mata menggunakan algoritma CNN yang diperlukan pada pembuatan sistem ini adalah

pustaka Tensorflow dan modul Keras. Tensorflow merupakan pustaka yang diperlukan untuk mendukung membangun model *machine learning* [25]. Tensorflow yang digunakan untuk membangun model klasifikasi citra menggunakan CNN adalah Tensorflow dengan versi 2.3.0. Contoh penerapan Tensorflow pada pembuatan model klasifikasi citra adalah penggunaan modul Keras, pada penelitian ini penulis menggunakan modul *keras_application* dengan versi 1.0.8 dan *keras_preprocessing* dengan versi 1.1.0.

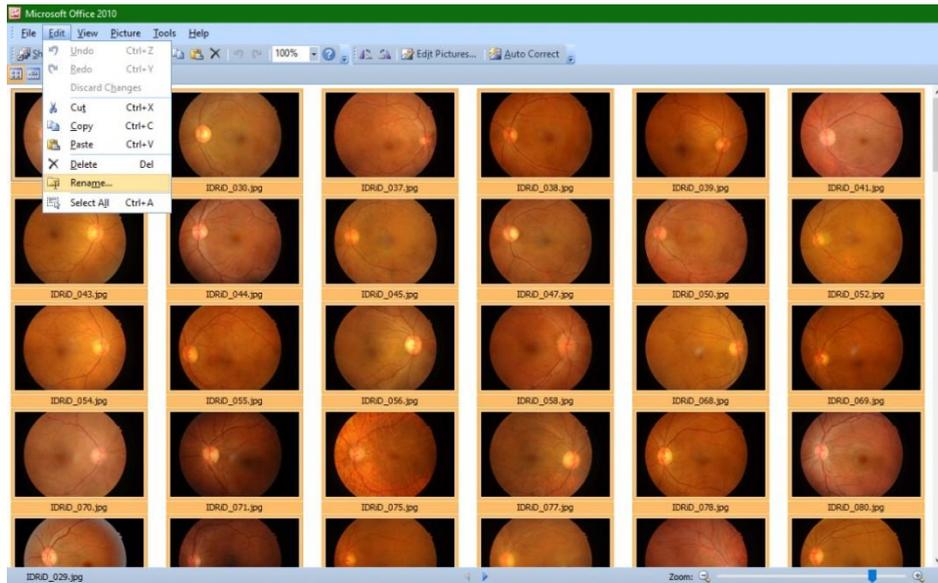
2. Untuk bahasa pemrograman yang digunakan adalah Bahasa pemrograman Python, keseluruhan sistem dimulai dari pengambilan citra, *pre-processing* citra menggunakan *library* PIL, segmentasi citra menggunakan *library* skimage, klasifikasi citra menggunakan CNN, pengiriman pesan ke Telegram dengan menggunakan *library* telebot, pembunyian *buzzer*, penampilan LCD dan LED di program dengan Bahasa pemrograman Python. Untuk keseluruhan sistem digunakan juga *library* scipy.misc, *library* scipy.misc digunakan untuk membuka file gambar ke lingkungan pemrograman python, juga mengonversi gambar menjadi *array* [26].

4.1.2.1. Implementasi Program Klasifikasi Citra Menggunakan CNN

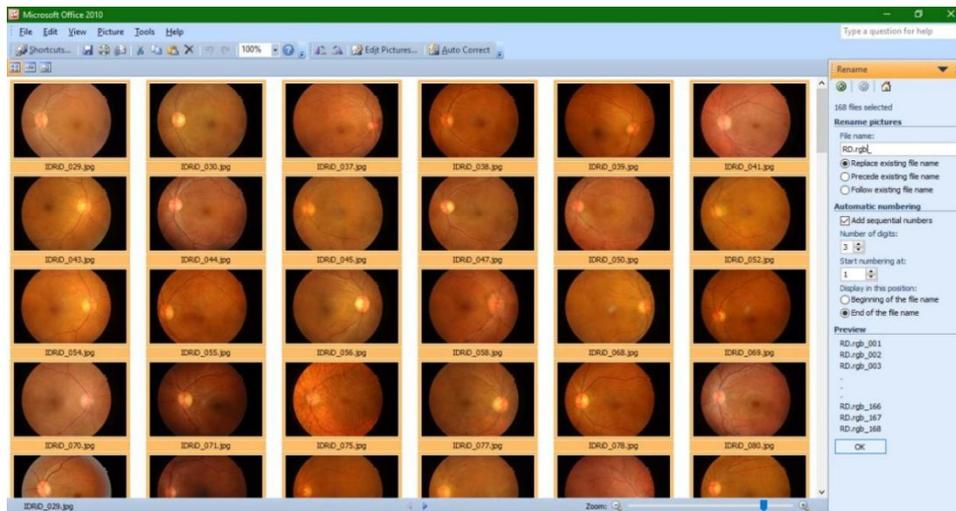
Klasifikasi citra 2D menggunakan CNN digunakan untuk mengklasifikasikan citra 2D fundus retina dengan pembagian menjadi 4 kelas, yaitu Normal, Mild NPDR, Moderate NPDR, dan Severe NPDR. Agar CNN dapat membedakan antara citra 2D fundus retina mata Normal, *Mild NPDR*, *Moderate NPDR*, dan *Severe NPDR* dari data citra 2D, maka sistem harus melalui proses *training* dan *testing* terlebih dahulu.

4.1.2.1.1. Implementasi Proses Set-dataset

Proses set-dataset CNN dilakukan pada Microsoft Office dengan menggunakan sumber data citra 2D fundus retina yaitu dataset IDRiD. Data citra memiliki total jumlah dataset yang akan digunakan yaitu 167 data citra 2D. Proses pemberian label pada 187 data dilakukan dengan menggunakan Microsoft Office seperti pada gambar 4.4.a dan 4.4.b berikut.



Gambar 4.2.a). *Labeling Dataset menggunakan Microsoft Office*



Gambar 4.2.b). *Pemberian label dengan tiga digit angka pembeda*

4.1.2.1.2. Implementasi Program *Training*

Proses *training* sistem dilakukan pada Google Colaboratory atau Google Colab. Proses *training* dilakukan dengan mencoba nilai epoch 100, 150 dan 300. Proses *training* yang dilakukan pada nilai epoch dan pembagian dataset yang berbeda memiliki hasil akurasi yang berbeda. Pada proses training, nilai didapat terdiri dari nilai akurasi, nilai akurasi dari proses *validation*, nilai *loss*, dan nilai *loss* dari proses *validation*. Berikut merupakan nilai hasil dari proses *training* ketiga pembagian dataset pada tabel 4.1. Hasil *Training*.

Tabel 4.1. Hasil *Training*

Epoch100					
Dataset	<i>Accuracy (%)</i>	<i>Loss</i>	<i>Validation Accuracy (%)</i>	<i>Validation Loss</i>	<i>Waktu (S)</i>
Dataset A	86,16	0,5268	68	1,0348	6,5052
Dataset B	86,34	0,4464	68	0,8860	6,4763
Dataset C	84,92	0,4934	72	0,9066	7,8518
Epoch 150					
Dataset	<i>Accuracy (%)</i>	<i>Loss</i>	<i>Validation Accuracy (%)</i>	<i>Validation Loss</i>	<i>Waktu (S)</i>
Dataset A	93,51	0,3269	64	1,0379	11,9651
Dataset B	92,41	0,2925	72	0,6167	9,6580
Dataset C	90,43	0,3288	72	0,9735	10,5181
Epoch 300					
Dataset	<i>Accuracy (%)</i>	<i>Loss</i>	<i>Validation Accuracy (%)</i>	<i>Validation Loss</i>	<i>Waktu (S)</i>
Dataset A	99,54	0,0491	72	0,9804	20,1477
Dataset B	100	0,0434	80	0,7559	19,8330
Dataset C	97,40	0,0894	64	0,9859	20,9337

Dari tabel hasil training, dapat disimpulkan bahwa nilai epoch 300 memiliki akurasi yang paling tinggi dari penggunaan epoch 100 dan epoch 150. Nilai akurasi tertinggi, nilai akurasi proses validasi tertinggi, nilai *loss* terendah dan nilai *loss* pada proses validasi terendah didapat ketika proses training menggunakan Dataset B, yaitu dataset dengan menerapkan segmentasi 1. Serta, penggunaan segmentasi pada dataset dapat meningkatkan nilai akurasi dan menurunkan nilai *loss* pada proses *training*.

Proses *training* CNN pada penelitian ini menggunakan model 'Sequential'. Model Sequential merupakan tumpukan layer yaitu terdiri dari layer Conv2D, Activation, MaxPooling2D, Flatten, Dense dan Dropout. Berikut *layer*, *shape* serta *parameter* penggunaan model Sequential pada gambar 4.3.

Model: "sequential_3"

Layer (type)	Output Shape	Param #
conv2d_6 (Conv2D)	(None, 85, 85, 32)	896
activation_9 (Activation)	(None, 85, 85, 32)	0
max_pooling2d_6 (MaxPooling2D)	(None, 42, 42, 32)	0
conv2d_7 (Conv2D)	(None, 21, 21, 64)	8256
activation_10 (Activation)	(None, 21, 21, 64)	0
max_pooling2d_7 (MaxPooling2D)	(None, 10, 10, 64)	0
flatten_3 (Flatten)	(None, 6400)	0
dense_6 (Dense)	(None, 256)	1638656
activation_11 (Activation)	(None, 256)	0
dropout_3 (Dropout)	(None, 256)	0
dense_7 (Dense)	(None, 4)	1028
Total params: 1,648,836		
Trainable params: 1,648,836		
Non-trainable params: 0		

Gambar 4.3 Model Summary yang digunakan pada proses *training*

Pada program *training* sistem dimasukkan nilai *batch size* sebesar 5, nilai *sample per epoch* 20, nilai *validation step* 5, *learning rate* 0,0004, dan *classes number* 5.

Berikut penjelasan model yang digunakan pada penelitian ini:

1. Pada tahapan pertama yaitu proses *convolution* pertama digunakan kernel dengan ukuran 3×3 serta jumlah filter sebanyak 32. Setelah proses *convolution*, ditambahkan sebuah aktivasi yaitu aktivasi RELU (Rectified Linear Unit) yang memiliki fungsi untuk mengubah nilai (-) dari sebuah matriks hasil *convolution* menjadi 0. Hasil *convolution* ini memiliki ukuran baru sebesar 85×85 .
2. Pada tahap selanjutnya yaitu proses *pooling* yang pertama. Proses *pooling* adalah pengurangan ukuran dari matriks dengan penggunaan operasi *pooling*. Pada penelitian ini digunakan *max pooling* agar mendapatkan nilai matriks yang baru. Berdasarkan hasil *pooling* yang pertama ini didapatkan matriks baru dengan ukuran 42×42 dari masukan hasil *convolution*

pertama dengan ukuran 85×85 . Proses ini digunakan kernel pooling sebesar 2×2 . Cara kerja dari *max pooling* yaitu dengan mengambil nilai paling besar dari pergeseran kernel sebanyak nilai stride yaitu 2.

3. Tahap ketiga yaitu proses *convolution* kedua dengan meneruskan dari hasil dari proses *pooling* pertama. Hasil kemudian dimasukan matriks gambar yang berukuran 42×42 dan jumlah filter sebanyak 64 dan kernel berukuran 3×3 . Proses *convolution* kedua ini juga menggunakan fungsi aktivasi yaitu aktivasi RELU. Proses ini menghasilkan citra dengan ukuran sebesar 21×21 .

4. Tahap keempat adalah proses *pooling* kedua dengan proses yang sama dengan proses *pooling* pertama, tetapi memiliki ukuran keluaran yang berbeda. Proses ini menghasilkan citra dengan ukuran sebesar 10×10 .

5. Tahap kelima yaitu proses *flatten* dan *fully connected* dengan hanya menggunakan satu lapisan tersembunyi dalam jaringan MLP. *Flatten* digunakan untuk mengubah keluaran lapisan *pooling* menjadi sebuah vektor. Sebelum melalui tahap klasifikasi atau memprediksi citra, pada tahap ini harus menambahkan nilai *dropout*. Tujuan dari *dropout* ini adalah mengurangi tingkat *overfitting* pada proses pelatihan. Nilai *dropout* yang digunakan yaitu 0,5 dengan menghasilkan nilai 256.

6. Tahap akhir yaitu proses penggunaan aktivasi fungsi Softmax. Fungsi ini biasanya digunakan pada proses yang menggunakan metode klasifikasi *multinomial logistic regression* dan *multiclass linear discriminant analysis*.

4.1.2.1.3. Implementasi Program Segmentasi RAG Threshold

Segmentasi RAG Threshold yang digunakan pada penelitian ini, menggunakan 2 keluaran citra berbeda yang dibedakan berdasarkan segmentasi dan program yang dibangun. Berikut ini adalah segmentasi 1 menggunakan RAG Threshold pada gambar 4.4 dan segmentasi 2 pada gambar 4.5.

```

labels1 = segmentation.slic(img, compactness=30, n_segments=100)
out1 = color.label2rgb(labels1, img, kind='avg', bg_label=0)

```

Gambar 4.4 RAG Thresholding Segmentasi 1

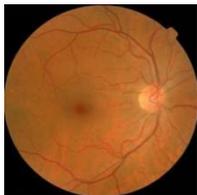
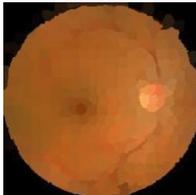
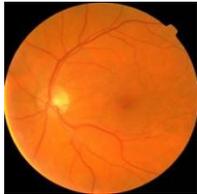
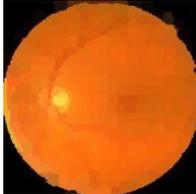
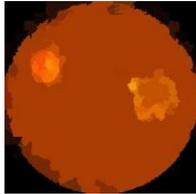
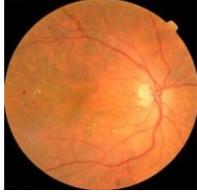
```

g = graph.rag_mean_color(img, labels1)
labels2 = graph.cut_threshold(labels1, g, 29)
out2 = color.label2rgb(labels2, img, kind='avg', bg_label=0)

```

Gambar 4.5 RAG Thresholding Segmentasi 2

Hasil segmentasi citra menggunakan RAG Threshold dengan keluaran segmentasi 1 dan keluaran segmentasi 2 memiliki hasil yang berbeda. Berikut tabel perbandingan citra dataset tanpa segmentasi dengan hasil segmentasi RAG Thresholding segmentasi 1 dan segmentasi 2 pada gambar 4.6.

	Tanpa Segmentasi	Segmentasi 1	Segmentasi 2
Normal			
Mild NPDR			
Moderate NPDR			
Severe NPDR			

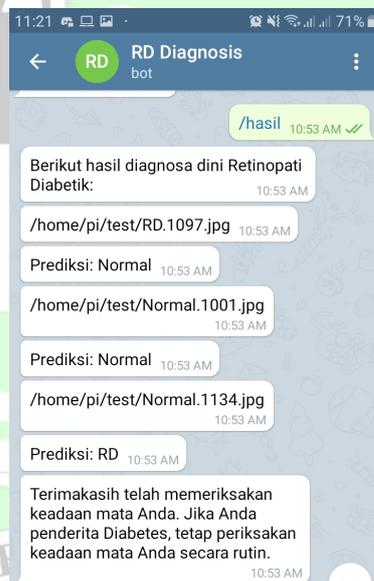
Gambar 4.6 Citra Hasil Segmentasi RAG Thresholding

4.1.2.2. Implementasi Program Pengiriman Hasil ke Telegram

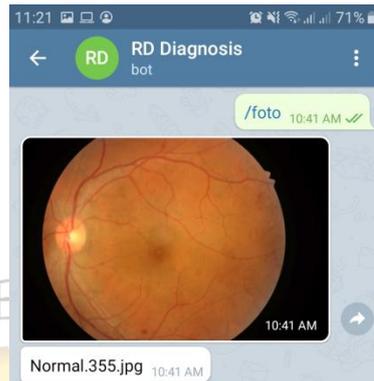
Program pengiriman hasil klasifikasi ke aplikasi sosial media Telegram, menggunakan *library* telebot. Pada penelitian ini, hasil dari klasifikasi akan dikirim melalui pesan pada aplikasi Telegram. Hasil pengiriman pesan dari Bot Telegram untuk masing-masing menu, dapat dilihat pada gambar 4.7 a), 4.7 b) dan 4.7 c) berikut.



Gambar 4.7. a) Tampilan hasil pesan /menu Bot Telegram RD Diagnosa



Gambar 4.7 b) Tampilan hasil pesan /hasil Bot Telegram RD Diagnosa



Gambar 4.7 c) Tampilan hasil pesan /foto Bot Telegram RD Diagnosa

4.2. Pengujian dan Analisis

Pengujian dan analisis dilakukan untuk mengkaji dan menganalisis hasil dari pengimplementasian perangkat lunak pada sistem.

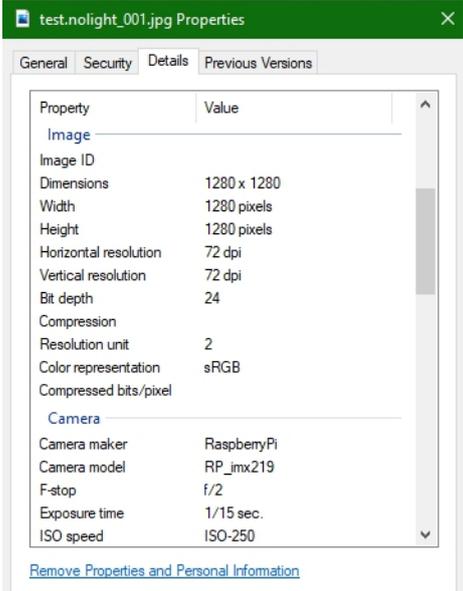
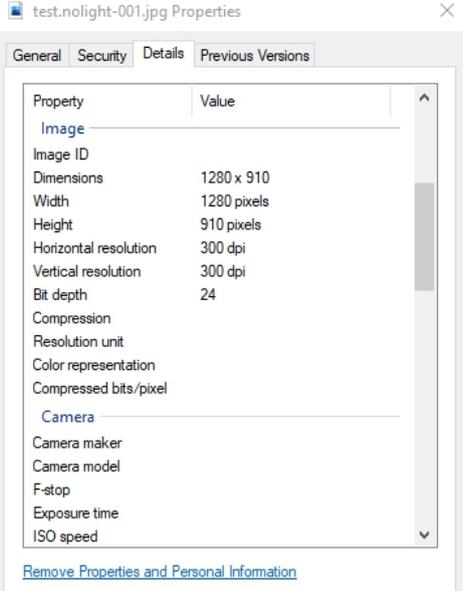
4.2.1. Pengujian dan Analisis Perangkat Lunak

Pengujian perangkat lunak dilakukan pada program *Pre-processing* citra dengan menggunakan *library* PIL, program klasifikasi citra retina dengan menggunakan CNN, program pengiriman hasil klasifikasi CNN ke aplikasi Telegram menggunakan *library* telebot, dan program indikator sistem berupa *Buzzer*, LED Merah, LED Kuning, LED Hijau, dan LCD 16×2 .

4.2.1.1. Pengujian Program *Pre-processing* Citra

Pengujian perangkat lunak *Pre-processing* citra dilakukan menggunakan *library* PIL. *Pre-processing* citra dilakukan untuk mengubah resolusi pada citra. Sintaks yang digunakan untuk menaikkan resolusi yaitu $\text{dpi} = (300, 300)$. Berikut merupakan hasil dari *pre-processing* citra pada tabel 4.2.

Tabel 4.2. Hasil *Pre-processing* Citra

Jenis	Sebelum <i>Pre-processing</i>	Setelah <i>Pre-processing</i>
Format citra 2D		

Pada tabel pengujian perangkat lunak *Pre-processing* citra dapat diamati bahwa data citra melalui proses *resize* resolusi untuk mengubah kualitas resolusi citra yang digunakan pada proses *testing* agar memiliki resolusi sama untuk semua data yaitu 300×300 dpi.

4.2.1.2. Pengujian Program Testing Menggunakan Algoritma CNN

Proses *testing* dilakukan pada Google Colaboratory atau Google Colab dengan menggunakan masukan data *testing* berupa 24 data citra dari dataset yang berbeda yaitu dataset A, dataset B dan dataset C, serta nilai epoch 100, 150 dan 300. Berikut hasil *testing* pada dataset A pada tabel 4.3.

Tabel 4.3 Hasil *Testing* Dataset A

Label	Nomor Citra	Epoch 100	Epoch 150	Epoch 300
		Hasil Klasifikasi	Hasil Klasifikasi	Hasil Klasifikasi
Normal	038	Mild	Mild	Mild
Normal	039	Mild	Mild	Mild

Normal	040	Mild	Mild	Mild
Normal	041	Mild	Mild	Mild
Normal	042	Moderate	Mild	Mild
Normal	043	Mild	Mild	Mild
Mild	038	Mild	Mild	Mild
Mild	039	Mild	Mild	Mild
Mild	040	Moderate	Mild	Mild
Mild	041	Mild	Mild	Mild
Mild	042	Mild	Mild	Mild
Mild	043	Mild	Mild	Mild
Moderate	032	Mild	Severe	Severe
Moderate	033	Mild	Mild	Mild
Moderate	034	Mild	Mild	Mild
Moderate	035	Mild	Mild	Severe
Moderate	042	Mild	Mild	Mild
Moderate	043	Mild	Mild	Mild
Severe	038	Mild	Mild	Mild
Severe	039	Moderate	Mild	Severe
Severe	040	Severe	Severe	Severe
Severe	041	Mild	Mild	Mild
Severe	042	Mild	Mild	Mild
Severe	043	Mild	Mild	Mild

Tabel 4.3 di atas merupakan hasil pengujian *testing* sistem yang dilakukan pada dataset A dengan model menggunakan epoch 100, 150 dan 300. Pada perhitungan rata-rata nilai *error* data hasil *testing* sistem, dilakukan perhitungan di bawah dan didapat hasil seperti pada tabel 4.4 rata-rata hasil testing dataset A.

$$\text{Rata - Rata Nilai Error} = \frac{\text{Jumlah Data Salah}}{\text{Jumlah Data Keseluruhan}}$$

Tabel 4.4 Rata-rata Hasil *Testing* Dataset A

Parameter	Epoch 100	Epoch 150	Epoch 300
Jumlah data salah	17	17	16
Rata – Rata Nilai <i>Error</i>	0,7083	0,7083	0,6667
Persentase	70,83%	70,83%	66,67%

Proses *testing* pada dataset A memiliki nilai persentase nilai *error* paling rendah yaitu 66,67%. Nilai persentase berkurang dari 70,83% hingga 66,67% dipengaruhi oleh menggunakan epoch 300 pada model ketika proses training. Proses *testing* berikutnya dilakukan pada dataset B pada tabel 4.5.

Tabel 4.5 Hasil *Testing* Dataset B

Label	Nomor Citra	Epoch 100	Epoch 150	Epoch 300
		Hasil Klasifikasi	Hasil Klasifikasi	Hasil Klasifikasi
Normal	038	Mild	Mild	Mild
Normal	039	Severe	Severe	Severe
Normal	040	Mild	Mild	Mild
Normal	041	Mild	Mild	Mild
Normal	042	Moderate	Mild	Mild
Normal	043	Mild	Mild	Moderate
Mild	038	Mild	Mild	Mild
Mild	039	Mild	Mild	Mild
Mild	040	Mild	Mild	Mild
Mild	041	Mild	Severe	Mild
Mild	042	Mild	Mild	Mild
Mild	043	Mild	Mild	Mild
Moderate	039	Mild	Severe	Moderate
Moderate	040	Mild	Mild	Mild
Moderate	041	Mild	Mild	Mild

Moderate	042	Mild	Severe	Severe
Moderate	043	Mild	Mild	Mild
Moderate	044	Mild	Mild	Mild
Severe	038	Mild	Mild	Mild
Severe	039	Mild	Mild	Mild
Severe	040	Mild	Mild	Mild
Severe	041	Mild	Mild	Mild
Severe	042	Mild	Mild	Mild
Severe	043	Mild	Mild	Mild

Tabel 4.5 di atas merupakan hasil pengujian *testing* sistem yang dilakukan pada dataset A dengan model menggunakan epoch 100, 150 dan 300. Pada perhitungan rata-rata nilai *error* data hasil *testing* sistem, dilakukan perhitungan di bawah dan didapat hasil seperti pada tabel 4.6 rata-rata hasil testing dataset B.

Tabel 4.6 Rata-rata Hasil Testing Dataset B

Parameter	Epoch 100	Epoch 150	Epoch 300
Jumlah data salah	18	19	17
Rata – Rata Nilai Error	0,75	0,7916	0,7083
Persentase	75%	79,16%	70,83%

Proses *testing* pada dataset B memiliki nilai persentase nilai *error* paling rendah yaitu 70,83%. Nilai persentase berkurang dari 75% hingga 70,83% dipengaruhi oleh menggunakan epoch 300 pada model ketika proses *training*. Proses *testing* berikutnya dilakukan pada dataset C pada tabel 4.7.

Tabel 4.7 Hasil Testing Dataset C

Label	Nomor Citra	Epoch 100	Epoch 150	Epoch 300
		Hasil Klasifikasi	Hasil Klasifikasi	Hasil Klasifikasi
Normal	038	Mild	Mild	Mild

Normal	039	Severe	Severe	Severe
Normal	040	Mild	Mild	Mild
Normal	041	Mild	Mild	Mild
Normal	042	Mild	Mild	Mild
Normal	043	Mild	Mild	Mild
Mild	038	Mild	Mild	Mild
Mild	039	Mild	Severe	Mild
Mild	040	Mild	Severe	Mild
Mild	041	Severe	Severe	Severe
Mild	042	Mild	Mild	Mild
Mild	043	Mild	Mild	Mild
Moderate	039	Moderate	Moderate	Moderate
Moderate	040	Severe	Severe	Mild
Moderate	041	Mild	Mild	Mild
Moderate	042	Severe	Severe	Severe
Moderate	043	Mild	Mild	Mild
Moderate	044	Mild	Mild	Mild
Severe	038	Mild	Mild	Mild
Severe	039	Mild	Mild	Mild
Severe	040	Mild	Severe	Mild
Severe	041	Mild	Mild	Mild
Severe	042	Severe	Severe	Severe
Severe	043	Severe	Severe	Severe

Tabel 4.7 di atas merupakan hasil pengujian *testing* sistem yang dilakukan pada dataset A dengan model menggunakan epoch 100, 150 dan 300. Pada perhitungan rata-rata nilai *error* data hasil *testing* sistem, dilakukan perhitungan di bawah dan didapat hasil seperti pada tabel 4.8 rata-rata hasil testing dataset C.

Tabel 4.8 Rata-rata Hasil Testing Dataset C

Parameter	Epoch 100	Epoch 150	Epoch 300
Jumlah data salah	16	17	16
Rata – Rata Nilai <i>Error</i>	0,6667	0,7083	0,6667
Persentase	66,67%	70,83%	66,67%

Proses *testing* pada dataset C memiliki nilai persentase nilai *error* paling rendah yaitu 66,67%. Nilai persentase berkurang dari 70,83% hingga 66,67% dipengaruhi oleh penggunaan epoch 300 pada model ketika proses *training*.

Persentase yang didapat pada proses *testing* menggunakan dataset B dengan epoch 300 memiliki hasil lebih tinggi dari persentase proses *testing* menggunakan dataset A dan C dengan epoch 300. Persentase nilai *error* dengan menggunakan dataset C dengan epoch 300 lebih rendah dipengaruhi oleh citra yang dihasilkan setelah melalui tahap segmentasi label 2 dan mampu menghasilkan persentase sama dengan menggunakan dataset A dengan epoch 300. Dengan demikian, dapat disimpulkan bahwa dengan menggunakan epoch 300 dan segmentasi *RAG Thresholding* dapat mengurangi persentase nilai *error* pada klasifikasi menggunakan algoritma CNN.

4.2.1.3. Pengujian Program Pengiriman Hasil ke Telegram

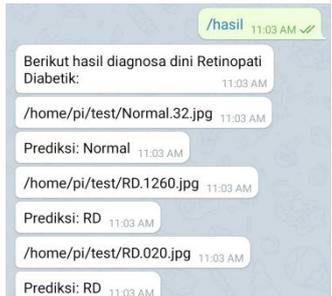
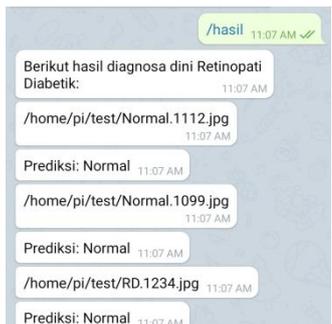
Pengujian program pengiriman hasil ke Telegram dilakukan untuk menguji kesesuaian pesan yang dikirimkan dan lama waktu pengiriman dari sistem ke aplikasi Telegram. Program dibangun melalui Bot Telegram dengan *username* @RD_Diagnosa. Pengiriman pesan hasil ketika pengguna mengirimkan pesan berupa /menu, /hasil, dan /test. Pengiriman pesan hasil diagnosa dilakukan dengan menggunakan pesan *feedback* untuk mencegah pesan terkirim langsung ke pengguna lain yang tidak sedang menggunakan alat.

Program pengiriman pesan hasil ke Telegram menggunakan pendekatan kondisi *'if'* dan *'else'*. Dimana hasil diagnosa mata normal diberikan nilai 0 sedangkan

hasil diagnosa mata terindikasi RD diberi nilai 1. Berikut merupakan tabel pengiriman hasil *testing* ke Bot Telegram Diagnosa RD pada tabel 4.9.

Tabel 4.9 Pengiriman hasil *testing* ke Telegram Bot Diagnosa RD

No.	Hasil <i>Testing</i> Sistem	Hasil <i>Testing</i> pada Telegram	Waktu per pesan	Total Waktu
1	<p>/home/pi/test/Normal.373.jpg Prediksi: RD</p> <p>/home/pi/test/Normal.355.jpg Prediksi: Normal</p> <p>/home/pi/test/RD.1229.jpg Prediksi: RD</p>		<p>/menu : 1s /hasil : 5s /foto : 11s</p>	17s
2	<p>/home/pi/test/RD.308.jpg Prediksi: RD</p> <p>/home/pi/test/Normal.400.jpg Prediksi: RD</p> <p>/home/pi/test/RD.122.jpg Prediksi: RD</p>		<p>/menu : 1s /hasil : 5s /foto : 14s</p>	20s
3	<p>/home/pi/test/RD.1097.jpg Prediksi: Normal</p> <p>/home/pi/test/Normal.1001.jpg Prediksi: Normal</p> <p>/home/pi/test/Normal.1134.jpg Prediksi: RD</p>		<p>/menu : 1s /hasil : 5s /foto : 14s</p>	20s
4	<p>/home/pi/test/RD.326.jpg Prediksi: Normal</p> <p>/home/pi/test/RD.1229.jpg Prediksi: RD</p> <p>/home/pi/test/Normal.386.jpg Prediksi: Normal</p>		<p>/menu : 1s /hasil : 5s /foto : 12s</p>	18s

5	<pre> /home/pi/test/Normal.32.jpg Prediksi: Normal /home/pi/test/RD.1260.jpg Prediksi: RD /home/pi/test/RD.020.jpg Prediksi: RD </pre>		<pre> /menu : 1s /hasil : 5s /foto : 12s </pre>	18s
6	<pre> /home/pi/test/Normal.1112.jpg Prediksi: Normal /home/pi/test/Normal.1099.jpg Prediksi: Normal /home/pi/test/RD.1234.jpg Prediksi: Normal </pre>		<pre> /menu : 1s /hasil : 5s /foto : 13s </pre>	19s

Pada tabel 4.9, didapatkan kesimpulan bahwa hasil diagnosa pada sistem dan hasil yang ditampilkan pada Telegram sama. Rata-rata waktu pengiriman /menu dari sistem ke Telegram adalah 1 detik, rata-rata waktu pengiriman /hasil dari sistem ke Telegram adalah 5 detik, dan rata-rata waktu pengiriman /foto dari sistem ke Telegram adalah 12,6 detik,. Sehingga, total waktu rata-rata yang dibutuhkan untuk melakukan pengiriman semua jenis pesan dari sistem ke Telegram adalah 18,6 detik.

4.2.1.4. Pengujian Program Indikator Sistem

Pengujian program indikator sistem bertujuan untuk menguji ketepatan berjalannya indikator sesuai kondisi yang telah ditentukan. Pada pengujian indikator sistem, pengujian dibagi menjadi tiga program indikator. Pertama, terdapat LED dengan tiga warna yang bertujuan mengindikasikan kondisi yang berbeda pada masing-masing warna. Kedua, terdapat *buzzer* yang akan mengeluarkan bunyi yang berbeda yang bertujuan mengindikasikan kondisi yang berbeda pada masing-masing suara *buzzer* yang dihasilkan. Ketiga, terdapat LCD 16×2 yang akan menampilkan kondisi sistem berupa tampilan tulisan sesuai program yang sedang berjalan. Berikut tabel pengujian indikator sistem.

Tabel 4.10 Pengujian keseluruhan indikator sistem

No	Kondisi Sistem	LED			Buzzer	LCD 16 × 2	Hasil
		Merah	Kuning	Hijau			
1.	Proses Pre-processing	ON	-	-	-	“Proses Pre-processing”	
2.	Proses Klasifikasi	ON	-	-	-	“Proses Klasifikasi”	
3.	Didapatkan hasil diagnosa dini RD	-	ON	-	2×	“Lihat hasil pada Telegram”	
4.	Pengiriman pesan hasil ke Telegram	-	-	ON	3×	“Pesan berhasil terkirim”	

4.2.3. Pengujian dan Analisis Keseluruhan Sistem

Pengujian sistem secara keseluruhan berarti menguji sistem secara keseluruhan yang mana hasil dari sistem adalah sistem dapat menampilkan hasil diagnosa dini RD dari klasifikasi citra 2D menggunakan CNN dengan tepat pada aplikasi Telegram. Keluaran dari sistem secara keseluruhan adalah memberikan indikator pada penggunaan alat, sehingga kondisi aktivitas sistem dapat diketahui dari suara *buzzer*, dapat dibedakan dari warna LED, serta dapat melihat hasil secara detail dari aplikasi Telegram. Pengujian keseluruhan sistem dilakukan dengan memasukkan data pada citra 2D dari dataset *testing*. Berikut hasil dari pengujian sistem secara keseluruhan pada tabel 4.11.

Tabel 4.11 Keluaran sistem secara keseluruhan

Percoba-an ke-	Label Citra	Nomor Citra	Hasil Klasifikasi	Pengiriman ke Telegram	Hasil Indikator Sistem
1.	Normal	038	Mild	Terkirim	LED = Menyala, Buzzer = Berbunyi, LCD = Sesuai kondisi sistem

2.	Normal	039	Severe	Terkirim	LED = Menyala, Buzzer = Berbunyi, LCD = Sesuai kondisi sistem
3.	Normal	040	Mild	Terkirim	LED = Menyala, Buzzer = Berbunyi, LCD = Sesuai kondisi sistem
4.	Mild	038	Mild	Terkirim	LED = Menyala, Buzzer = Berbunyi, LCD = Sesuai kondisi sistem
5.	Mild	039	Mild	Terkirim	LED = Menyala, Buzzer = Berbunyi, LCD = Sesuai kondisi sistem
6.	Mild	040	Mild	Terkirim	LED = Menyala, Buzzer = Berbunyi, LCD = Sesuai kondisi sistem
7.	Moderate	039	Moderate	Terkirim	LED = Menyala, Buzzer = Berbunyi, LCD = Sesuai kondisi sistem
8.	Moderate	040	Mild	Terkirim	LED = Menyala, Buzzer = Berbunyi, LCD = Sesuai kondisi sistem
9.	Moderate	041	Mild	Terkirim	LED = Menyala, Buzzer = Berbunyi, LCD = Sesuai kondisi sistem
10.	Severe	041	Mild	Terkirim	LED = Menyala, Buzzer = Berbunyi, LCD = Sesuai kondisi sistem
11.	Severe	042	Severe	Terkirim	LED = Menyala, Buzzer = Berbunyi, LCD = Sesuai kondisi sistem
12.	Severe	043	Severe	Terkirim	LED = Menyala, Buzzer = Berbunyi, LCD = Sesuai kondisi sistem

Pada tabel pengujian keseluruhan sistem diatas, terdapat enam hasil klasifikasi yang tidak tepat dan enam hasil klasifikasi yang tepat dengan menggunakan

dataset C dan epoch 300. Pada perhitungan persentase rata-rata nilai *error* data hasil klasifikasi data citra pada sistem, dilakukan perhitungan sebagai berikut.

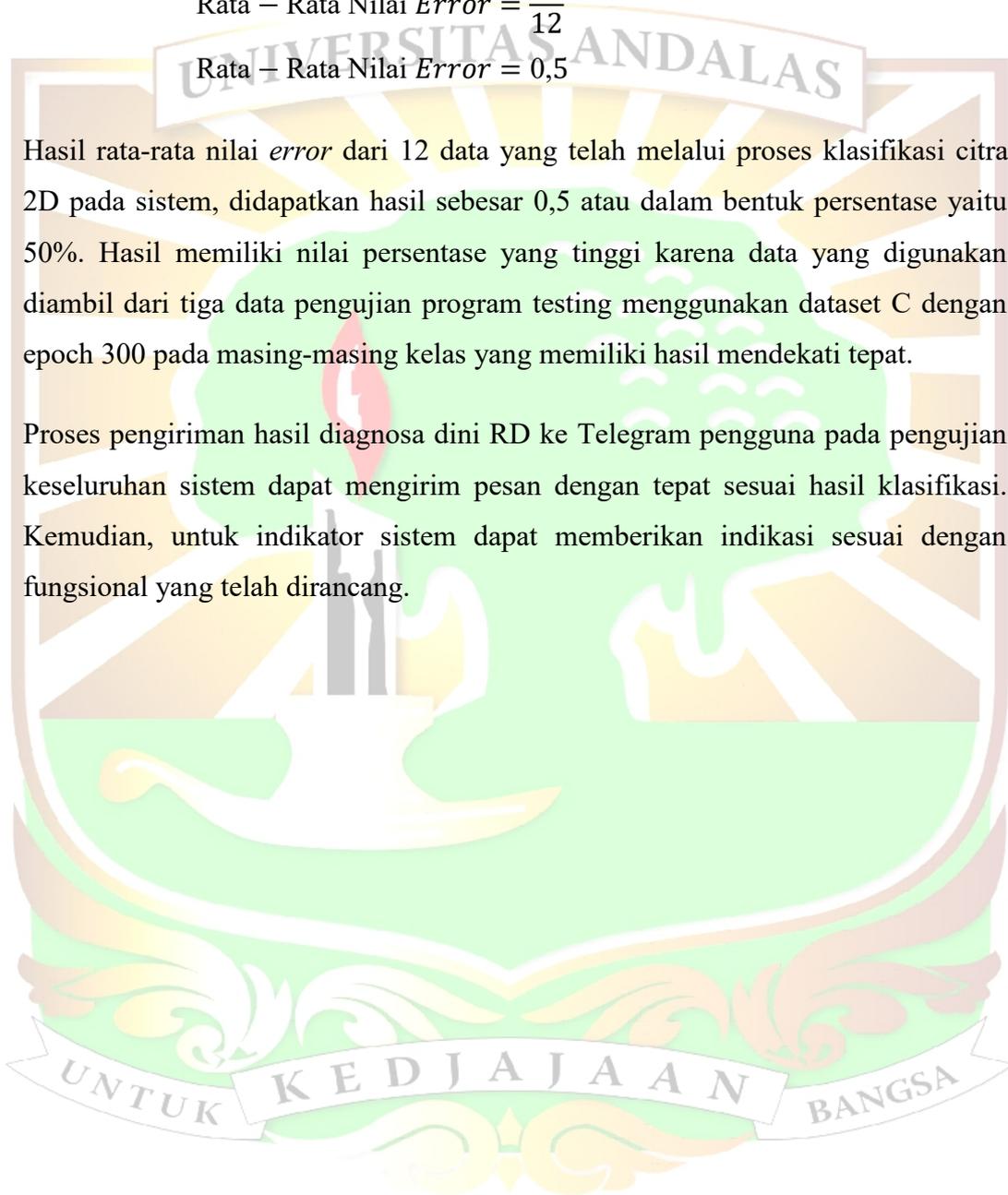
$$\text{Rata - Rata Nilai } Error = \frac{\text{Jumlah Data Salah}}{\text{Jumlah Data Keseluruhan}}$$

$$\text{Rata - Rata Nilai } Error = \frac{6}{12}$$

$$\text{Rata - Rata Nilai } Error = 0,5$$

Hasil rata-rata nilai *error* dari 12 data yang telah melalui proses klasifikasi citra 2D pada sistem, didapatkan hasil sebesar 0,5 atau dalam bentuk persentase yaitu 50%. Hasil memiliki nilai persentase yang tinggi karena data yang digunakan diambil dari tiga data pengujian program testing menggunakan dataset C dengan epoch 300 pada masing-masing kelas yang memiliki hasil mendekati tepat.

Proses pengiriman hasil diagnosa dini RD ke Telegram pengguna pada pengujian keseluruhan sistem dapat mengirim pesan dengan tepat sesuai hasil klasifikasi. Kemudian, untuk indikator sistem dapat memberikan indikasi sesuai dengan fungsional yang telah dirancang.



BAB V PENUTUP

5.1. Kesimpulan

Berdasarkan pengujian dan analisis sistem keseluruhan pada sistem Portabel Diagnosa Dini Retinopati Diabetik dengan Menggunakan Algoritma *Convolutional Neural Network*, diperoleh kesimpulan berupa:

1. Proses *training* pada klasifikasi algoritma CNN yang dilakukan dengan menggunakan segmentasi *RAG Thresholding* dan epoch 300 dapat meningkatkan nilai akurasi hingga didapat persentase akurasi sebesar 100%.
2. Proses *testing* pada klasifikasi algoritma CNN yang dilakukan dengan menggunakan segmentasi *RAG Thresholding* dan epoch 300 dapat mengurangi nilai *error* hingga didapat persentase nilai *error* sebesar 66,67%
3. Bot telegram dapat mengirim hasil diagnosa dini retinopati diabetik dari sistem menuju aplikasi telegram berupa pesan singkat dan citra 2D.

5.2. Saran

Untuk melakukan pengembangan pada penelitian ini berdasarkan pengujian dan analisis sistem secara keseluruhan, oleh karena itu untuk penelitian selanjutnya adapun beberapa saran yang akan meningkatkan kinerja dari sistem ini, yaitu:

1. Menggunakan algoritma lain yang lebih spesifik berfokus pada perbedaan antara retina normal atau retina yang terindikasi Retinopati Diabetik. Misalnya, perbedaan pada bercak pecahnya pembuluh darah, pembengkakan pembuluh darah dan indikasi retinopati diabetik lainnya yang dapat dibedakan antar data.
2. Membangun *database online* yang dapat menyimpan hasil citra *realtime* beserta hasil diagnosa dini yang dipisahkan berdasarkan nama pengguna dan waktu penggunaan alat.
3. Menambahkan fitur autentifikasi pada alat, sehingga sistem dapat menyimpan data hasil penggunaan alat berdasarkan identitas pengguna dan waktu penggunaan alat.
4. Membangun sistem agar alat serta program dapat berjalan dengan menggunakan pemicu dari komponen tambahan pada alat, misalnya *push button* atau fitur login.

DAFTAR PUSTAKA

- [1]. Holland K & Tyler W. (2016). “*What You Should Know About Diabetes and Eye Exams*”. diakses pada halaman *web* www.healthline.com.
- [2]. Soelistijo S.A., DKK. (2015). “Konsensus Pengelolaan Dan Pencegahan Diabetes Melitus Tipe 2 Di Indonesia 2015”. Penerbit PB. PERKENI. 978-979-19388-6-0.
- [3]. Wulandari C.D.R., Suryo A.W., & Ledyo N. (2019). “Klasifikasi Diabetes Retinopati Menggunakan Metode *Statistical Region Merging* Dan *Convolutional Neural Network*”. Teknik Telekomunikasi, Fakultas Teknik Elektro, Universitas Telkom. Bandung.
- [4]. Singh N., Remesh C. T. (2010). “*Automated Early Detection of Diabetic Retinopathy Using Image Analysis Techniques*”. *Int. J. Comput. Appl.*, vol. 8, no. 2.
- [5]. Kusuma B. (2017). “Apa yang Dimaksud dengan Retinopati Diabetik?”. diakses pada halaman *web* dictio.id.
- [6]. R. Sitompul. (2011). “*Diabetic Retinopathy*”. Jakarta: *Department of Ophthalmology*, Universitas Indonesia.
- [7]. A.P. Shingade & A.R. Kasetwar. (2011). “*A Review on Implementation of Algorithm for Detection of Diabetic Retinopathy*”. *Int. J. Res Eng. Technol.*, Vol.2016.PP.8794.
- [8]. P. Porwal, S. Pachade and M. Kokare et al. (2019). “*Medical Image Analysis*”. Elsevier, 59 (2020) 101561
- [9]. Jain R, Rangachar K, Brian .G.S. (1995). “*Machine Vision*”. *Published by McGraw-Hil, Inc.* ISBN 0-07-0320 18-7.
- [10].D. Putra. (2010). “*Pengolahan Citra Digital*“. Yogyakarta: Penerbit Andi.

- [11]. LeCun, Y., Bengio, Y., & Hinton, G. (2015). "Deep Learning". *Nature*, 521 (7533), 436-444.
- [12]. Nofriani. (2020). "Machine Learning Application for Classification Prediction of Household's Welfare Status". *NOFRIANI /JITCE - VOL. 04 NO. 02 (2020) 72-82*.
- [13]. Estari N., Lucky LVFC. (2017). "Implementasi Jaringan Syaraf Tiruan untuk Menilai kelayakan Tugas Akhir Mahasiswa (Studi Kasus di Amik Bukittinggi)". Volume 8, Nomor 1.
- [14]. Prasetyo A.P., Hanggar P.K., Setiya T.H., Puput F. (2018). "Model Penduga Produktifitas Perikanan Pukat Cincin di Laut Jawa". *J. Lit. Perikan . Ind. Vol. 18, No. 3*.
- [15]. Mayasari D., Lia L. (2014). "Jaringan Syaraf Tiruan". Fakultas Matematika dan Ilmu Pengetahuan Alam. Universitas Negeri Yogyakarta: Yogyakarta.
- [16]. Suartika W.E.P., Arya Y.W., dan Rully S. (2016). "Klasifikasi Citra Menggunakan CNN pada Caltech 101". *Jurnal Teknik ITS Vol. 5, No. 1*.
- [17]. _____. "Jaringan Saraf Konvolusional". diakses pada halaman *web* www.mathworks.com. pada tanggal 10 Januari 2020
- [18]. Danukusumo, Kevin Pudi. (2017). "Implementasi Deep Learning Menggunakan *Convolutional Neural Network* untuk Klasifikasi Citra Candi Berbasis GPU". Tugas Akhir. Program Studi Teknik Informatika Fakultas Teknologi Industri Universitas Atma Jaya Yogyakarta
- [19]. Jones R.C. (2011). "BBC - dot.Rory: *A 15 pound computer to inspire young programmers*". diakses pada halaman *web* bbc.co.uk.
- [20]. Yendri.D & Rahmi E.P. (2018). "Sistem Pengontrolan dan Keamanan Rumah Pintar (*Smart Home*) Berbasis Android". *JITCE - VOL. 02 NO. 01 (2018) 1-6*

- [21]. Dharmawan H.A. 2017. “Mikrokontroller: Konsep Dasar dan Praktis”. Penerbit UB Press: Jl. Veteran 10-11, Malang, 65145, Indonesia.
- [22]. Digmi I. (2018). “Google Colab Gratis untuk Belajar *Deep Learning*”. diakses pada halaman *web* imam,digmi.id. pada tanggal 6 Februari 2020.
- [23]. Jimmy. P. (2018). “Implementasi *Deep Learning* Menggunakan *Convolutional Neural Network* Untuk Klasifikasi Alat Tulis”. Jurusan Statistika, Fakultas Matematika Dan Ilmu Pengetahuan Alam, Universitas Islam Indonesia, Yogyakarta.
- [24]. Mulyanto. A. D. (2020). “Pemanfaatan Bot Telegram Untuk Media Informasi Penelitian”. Jurnal Ilmu Komputer dan Teknologi Informasi Volume 12, No. 1 (2020), pp 49-54, ISSN : 1978-161X(p); 2477-2550©
- [25]._____. “Mengapa TensorFlow”. diakses pada halaman *web* www.tensorflow.org. pada tanggal 24 Desember 2020.
- [26]. Pahwa. Y. (2020). “*Dog-Vs-Cats*”. diakses pada halaman *web* www.kaggle.com. pada tanggal 24 Desember 2020.
- [27]. Duhan20. “Convolution Neural Network using Keras in Python”. diakses pada halaman *web* github.com.
- [28]. _____. “*Python Pillow : Tutorials Point Simply Easy Learning*”. diakses pada halaman *web* www.tutorialspoint.com.
- [29]._____. “*RAG Thresholding*”. diakses pada halaman *web* scikit-image.org.
- [30]. NickL17. “*Set Up Telegram Bot on Raspberry Pi*”. diakses pada halaman *web* www.instructables.com

Lampiran 1

Source Code

Source Code Training [27]

```
import sys
import os
import matplotlib.pyplot as plt
from keras.preprocessing.image import ImageDataGenerator
from keras import optimizers
from keras.models import Sequential
from keras.layers import Dropout, Flatten, Dense, Activation
from keras.layers.convolutional import Convolution2D, MaxPooling2D
from keras import callbacks
import time

start = time.time()

DEV = False
argvs = sys.argv
argc = len(argvs)

if argc > 1 and (argvs[1] == "--development" or argvs[1] == "-d"):
    DEV = True

if DEV:
    epochs = 2
else:
    epochs = 300

train_data_path = './Training'
validation_data_path = './Validation'

img_width, img_height = 255, 255
batch_size = 5
samples_per_epoch = 20
validation_steps = 5
nb_filters1 = 32
nb_filters2 = 64
conv1_size = 3
conv2_size = 2
pool_size = 2
classes_num = 4
lr = 0.0004

model = Sequential()
model.add(Convolution2D(nb_filters1, conv1_size, conv1_size, padding = "same", input_shape=(img_width, img_height, 3)))
model.add(Activation("relu"))
model.add(MaxPooling2D(pool_size=(pool_size, pool_size)))

model.add(Convolution2D(nb_filters2, conv2_size, conv2_size, padding = "same"))
model.add(Activation("relu"))
model.add(MaxPooling2D(pool_size=(pool_size, pool_size), padding = 'valid'))

model.add(Flatten())
model.add(Dense(256))
model.add(Activation("relu"))
model.add(Dropout(0.5))
model.add(Dense(classes_num, activation='softmax'))

model.summary()

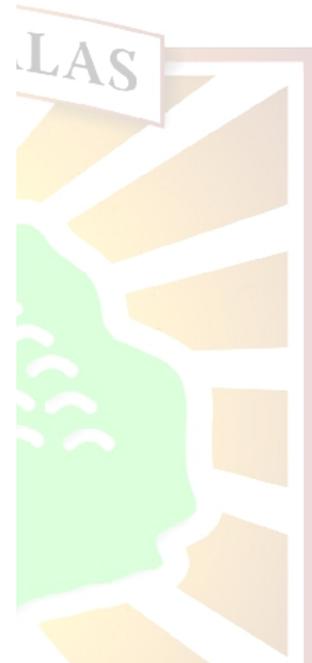
model.compile(loss='categorical_crossentropy',
              optimizer=optimizers.RMSprop(lr=lr),
              metrics=['accuracy'])

train_datagen = ImageDataGenerator(
    rescale=1. / 255,
    shear_range=0.2,
    zoom_range=0.2,
    horizontal_flip=True)

test_datagen = ImageDataGenerator(rescale=1. / 255)

train_generator = train_datagen.flow_from_directory(
    train_data_path,
    target_size=(img_height, img_width),
    batch_size=batch_size,
    class_mode='categorical')

validation_generator = test_datagen.flow_from_directory(
    validation_data_path,
    target_size=(img_height, img_width),
    batch_size=batch_size,
    class_mode='categorical')
```



```

log_dir = './models/tf-log/'
tb_cb = callbacks.TensorBoard(log_dir=log_dir, histogram_freq=0)
cbks = [tb_cb]

model.fit_generator(
    train_generator,
    steps_per_epoch=samples_per_epoch,
    epochs=epochs,
    validation_data=validation_generator,
    validation_steps=validation_steps,
    callbacks=cbks)

target_dir = './models/'
if not os.path.exists(target_dir):
    os.mkdir(target_dir)
model.save('./models/model.h5')
model.save_weights('./models/weights.h5')

end = time.time()
dur = end-start

if dur<60:
    print("Execution Time:",dur,"seconds")
elif dur>60 and dur<3600:
    dur=dur/60
    print("Execution Time:",dur,"minutes")
else:
    dur=dur/(60*60)
    print("Execution Time:",dur,"hours")

```

Source Code Testing [27]

```

import os
import numpy as np
from keras.preprocessing.image import ImageDataGenerator, load_img, img_to_array
from keras.models import Sequential, load_model
import time

#Define Path
model_path = './models/model.h5'
model_weights_path = './models/weights.h5'
test_path = './Test'

#Load the pre-trained models
model = load_model(model_path)
model.load_weights(model_weights_path)

#Define image parameters
img_width, img_height = 255, 255

import time
start = time.time()
#Prediction Function
def predict(file):
    x = load_img(file, target_size=(img_width,img_height))
    x = img_to_array(x)
    x = np.expand_dims(x, axis=0)
    array = model.predict(x)
    result = array[0]
    #print(result)
    answer = np.argmax(result)
    if answer == 0:
        print("Prediksi: Normal")
    elif answer == 1:
        print("Prediksi: Mild")
    elif answer == 2:
        print("Prediksi: Moderate")
    elif answer == 3:
        print("Prediksi: Severe")

    return answer

```



```

#Walk the directory for every image
for i, ret in enumerate(os.walk(test_path)):
    for i, filename in enumerate(ret[2]):
        if filename.startswith("."):
            continue

        print(ret[0] + '/' + filename)
        result = predict(ret[0] + '/' + filename)
        print(" ")

#Calculate execution time
end = time.time()
dur = end-start

if dur<60:
    print("Execution Time:",dur,"seconds")
elif dur>60 and dur<3600:
    dur=dur/60
    print("Execution Time:",dur,"minutes")
else:
    dur=dur/(60*60)
    print("Execution Time:",dur,"hours")

```

Source Code i2c_lcd_driver [28]

```

# -*- coding: utf-8 -*-
# Original code found at:
# https://gist.github.com/DenisFromHR/cc863375a6e19dce359d
"""
Compiled, mashed and generally mutilated 2014-2015 by Denis Pleic
Made available under GNU GENERAL PUBLIC LICENSE

# Modified Python I2C library for Raspberry Pi
# as found on http://www.recantha.co.uk/blog/?p=4849
# Joined existing 'i2c_lib.py' and 'lcddriver.py' into a single library
# added bits and pieces from various sources
# By DenisFromHR (Denis Pleic)
# 2015-02-10, ver 0.1
"""

# i2c bus (0 -- original Pi, 1 -- Rev 2 Pi)
I2CBUS = 1

# LCD Address
ADDRESS = 0x27

import smbus
from time import sleep

class i2c_device:
    def __init__(self, addr, port=I2CBUS):
        self.addr = addr
        self.bus = smbus.SMBus(port)

# Write a single command
    def write_cmd(self, cmd):
        self.bus.write_byte(self.addr, cmd)
        sleep(0.0001)

# Write a command and argument
    def write_cmd_arg(self, cmd, data):
        self.bus.write_byte_data(self.addr, cmd, data)
        sleep(0.0001)

```

```

# Write a block of data
def write_block_data(self, cmd, data):
    self.bus.write_block_data(self.addr, cmd, data)
    sleep(0.0001)

# Read a single byte
def read(self):
    return self.bus.read_byte(self.addr)

# Read
def read_data(self, cmd):
    return self.bus.read_byte_data(self.addr, cmd)

# Read a block of data
def read_block_data(self, cmd):
    return self.bus.read_block_data(self.addr, cmd)

# commands
LCD_CLEARDISPLAY = 0x01
LCD_RETURNHOME = 0x02
LCD_ENTRYMODESET = 0x04
LCD_DISPLAYCONTROL = 0x08
LCD_CURSORSHIFT = 0x10
LCD_FUNCTIONSET = 0x20
LCD_SETCGRAMADDR = 0x40
LCD_SETDRAMADDR = 0x80

# flags for display entry mode
LCD_ENTRYRIGHT = 0x00
LCD_ENTRYLEFT = 0x02
LCD_ENTRYSHIFTINCREMENT = 0x01
LCD_ENTRYSHIFTDECREMENT = 0x00

# flags for display on/off control
LCD_DISPLAYON = 0x04
LCD_DISPLAYOFF = 0x00
LCD_CURSORON = 0x02
LCD_CURSOROFF = 0x00
LCD_BLINKON = 0x01
LCD_BLINKOFF = 0x00

# flags for display/cursor shift
LCD_DISPLAYMOVE = 0x08
LCD_CURSORMOVE = 0x00
LCD_MOVERIGHT = 0x04
LCD_MOVELEFT = 0x00

# flags for function set
LCD_8BITMODE = 0x10
LCD_4BITMODE = 0x00
LCD_2LINE = 0x08
LCD_1LINE = 0x00
LCD_5x10DOTS = 0x04
LCD_5x8DOTS = 0x00

# flags for backlight control
LCD_BACKLIGHT = 0x08
LCD_NOBACKLIGHT = 0x00

En = 0b00000100 # Enable bit
Rw = 0b00000010 # Read/Write bit
Rs = 0b00000001 # Register select bit

class lcd:
    #initializes objects and lcd
    def __init__(self):
        self.lcd_device = i2c_device(ADDRESS)

        self.lcd_write(0x03)
        self.lcd_write(0x03)
        self.lcd_write(0x03)
        self.lcd_write(0x02)

        self.lcd_write(LCD_FUNCTIONSET | LCD_2LINE | LCD_5x8DOTS | LCD_4BITMODE)
        self.lcd_write(LCD_DISPLAYCONTROL | LCD_DISPLAYON)
        self.lcd_write(LCD_CLEARDISPLAY)
        self.lcd_write(LCD_ENTRYMODESET | LCD_ENTRYLEFT)
        sleep(0.2)

```



```

# clocks EN to latch command
def lcd_strobe(self, data):
    self.lcd_device.write_cmd(data | En | LCD_BACKLIGHT)
    sleep(.0005)
    self.lcd_device.write_cmd(((data & ~En) | LCD_BACKLIGHT))
    sleep(.0001)

def lcd_write_four_bits(self, data):
    self.lcd_device.write_cmd(data | LCD_BACKLIGHT)
    self.lcd_strobe(data)

# write a command to lcd
def lcd_write(self, cmd, mode=0):
    self.lcd_write_four_bits(mode | (cmd & 0xF0))
    self.lcd_write_four_bits(mode | ((cmd << 4) & 0xF0))

# write a character to lcd (or character rom) 0x09: backlight | RS=DR<
# works!
def lcd_write_char(self, charvalue, mode=1):
    self.lcd_write_four_bits(mode | (charvalue & 0xF0))
    self.lcd_write_four_bits(mode | ((charvalue << 4) & 0xF0))

# put string function with optional char positioning
def lcd_display_string(self, string, line=1, pos=0):
    if line == 1:
        pos_new = pos
    elif line == 2:
        pos_new = 0x40 + pos
    elif line == 3:
        pos_new = 0x14 + pos
    elif line == 4:
        pos_new = 0x54 + pos

    self.lcd_write(0x80 + pos_new)

    for char in string:
        self.lcd_write(ord(char), Rs)

# clear lcd and set to home
def lcd_clear(self):
    self.lcd_write(LCD_CLEARDISPLAY)
    self.lcd_write(LCD_RETURNHOME)

# clear lcd and set to home
def lcd_clear(self):
    self.lcd_write(LCD_CLEARDISPLAY)
    self.lcd_write(LCD_RETURNHOME)

# define backlight on/off (lcd.backlight(1); off= lcd.backlight(0)
def backlight(self, state): # for state, 1 = on, 0 = off
    if state == 1:
        self.lcd_device.write_cmd(LCD_BACKLIGHT)
    elif state == 0:
        self.lcd_device.write_cmd(LCD_NOBACKLIGHT)

# add custom characters (0 - 7)
def lcd_load_custom_chars(self, fontdata):
    self.lcd_write(0x40);
    for char in fontdata:
        for line in char:
            self.lcd_write_char(line)

```

Source Code Masking [28]

```

from PIL import Image, ImageDraw, ImageFilter

im1 = Image.open("./Masking/01-IDRiD/1_2.jpg")
im2 = Image.open("./black.jpg")

< (left, upper, right, lower) = (498, 651, 725, 878)
im_2 = im2.crop((left, upper, right, lower))

mask = Image.new("L", im1.size, 0)
draw = ImageDraw.Draw(mask)
#(kiri, atas, kanan, bawah)
draw.ellipse((0, 0, 227, 227), fill=400)
im = Image.composite(im1, im_2, mask)
im.save("./Masking/IDRiD - Masking/1_2.jpg")

```



Source Code RAG Threshold [29]

```
from skimage import data, segmentation, color
from skimage.future import graph
from matplotlib import pyplot as plt
from skimage import io

img = io.imread('./dataset/train/Normal/Normal.001.jpg')

labels1 = segmentation.slic(img, compactness=30, n_segments=100)
out1 = color.label2rgb(labels1, img, kind='avg', bg_label=0)

g = graph.rag_mean_color(img, labels1)
labels2 = graph.cut_threshold(labels1, g, 29)
out2 = color.label2rgb(labels2, img, kind='avg', bg_label=0)

io.imsave('./RAG Thresholding/IDRiD/out 1/0_10.jpg',out1)
io.imsave('./RAG Thresholding/IDRiD/out 2/0_10.jpg',out2)
```

Source Code Sistem

```
#CNN Package#
import os
import numpy as np
from keras.preprocessing.image import ImageDataGenerator, load_img, img_to_array
from keras.models import Sequential, load_model
import time

#Skimage Package#
from skimage import data, segmentation, color
from skimage.future import graph
from skimage import io

#Base Package#
from gpiozero import LED, Button, Buzzer
from picamera import PiCamera
from datetime import datetime
from PIL import Image, ImageDraw, ImageFilter
from time import sleep, time
from subprocess import check_call
from signal import pause
from time import *
import pyrebase
import matplotlib.pyplot as plt
import i2c_lcd_driver

#indikator
ledputih = LED(6)
ledmerah = LED(17)
ledkuning = LED(27)
ledhijau = LED(22)
buzz = Buzzer(23)
mylcd = i2c_lcd_driver.lcd()
timestamp = datetime.now().isoformat()
```

```

## Pre-processing ##

#open file gambar citra tes
im1 = Image.open("../Masking/01-IDRiD/1_2.jpg")
im2 = Image.open("../black.jpg")

#crop
(left, upper, right, lower) = (498, 651, 725, 878)
im_2 = im2.crop((left, upper, right, lower))

#Masking
mask = Image.new("L", im1.size, 0)
draw = ImageDraw.Draw(mask)
#(kiri, atas, kanan, bawah)
draw.ellipse((0, 0, 227, 227), fill=400)
im = Image.composite(im1, im_2, mask)

#simpan citra Pre-processing
im.save("../Masking/IDRiD - Masking/1_2.jpg")

##Masking##

#open file gambar citra tes
img = io.imread('../dataset/train/Normal/Normal.001.jpg')

#label1
labels1 = segmentation.slic(img, compactness=30, n_segments=100)
out1 = color.label2rgb(labels1, img, kind='avg', bg_label=0)

#label2
g = graph.rag_mean_color(img, labels1)
labels2 = graph.cut_threshold(labels1, g, 29)
out2 = color.label2rgb(labels2, img, kind='avg', bg_label=0)

#simpan citra
io.imsave('/home/pi/Desktop/pic/Foto citra fundus.jpg',out2)

#indikator segmentasi citra
sleep(0.2)
ledmerah.on()
buzz.on()
sleep(0.1)
buzz.off()
ledmerah.off()
sleep(0.1)

## CNN ##

#Define Path
model_path = '/home/pi/Desktop/program/cnn/model_fix_full/model.h5'
model_weights_path = '/home/pi/Desktop/program/cnn/model_fix_full/weights.h5'
test_path = '/home/pi/Desktop/pic'

#Load the pre-trained models
model = load_model(model_path)
model.load_weights(model_weights_path)

#Define image parameters
img_width, img_height = 255, 255

import time
start = time.time()
#Prediction Function
def predict(file):
    x = load_img(file, target_size=(img_width,img_height))
    x = img_to_array(x)
    x = np.expand_dims(x, axis=0)
    array = model.predict(x)
    result = array[0]
    #print(result)
    answer = np.argmax(result)
    if answer == 0:
        print("Prediksi: Normal")
    elif answer == 1:
        print("Prediksi: Mild")
    elif answer == 2:
        print("Prediksi: Moderate")
    elif answer == 3:
        print("Prediksi: Severe")

```



```

#Walk the directory for every image
for i, ret in enumerate(os.walk(test_path)):
    for j, filename in enumerate(ret[2]):
        if filename.startswith("."):
            continue

            print(ret[0] + '/' + filename)
            result = predict(ret[0] + '/' + filename)
            print(" ")

#Calculate execution time
end = time.time()
dur = end-start

if dur<60:
    print("Execution Time:",dur,"seconds")
elif dur>60 and dur<3600:
    dur=dur/60
    print("Execution Time:",dur,"minutes")
else:
    dur=dur/(60*60)
    print("Execution Time:",dur,"hours")

#indikator selesai
ledkuning.on()
mylcd.lcd_display_string(" ")
mylcd.lcd_display_string("_Hasil DIAGNOSA_")
sleep(0.5)
mylcd.lcd_display_string(" ")
mylcd.lcd_display_string("_Pada Telegram_")
sleep(1)
buzz.on()
sleep(0.2)
buzz.off()
sleep(0.1)
buzz.on()
sleep(0.1)
ledkuning.off()
buzz.off()
sleep(0.1)

## TELEGRAM ##
import time, datetime
import telepot
from telepot.loop import MessageLoop

now = datetime.datetime.now()

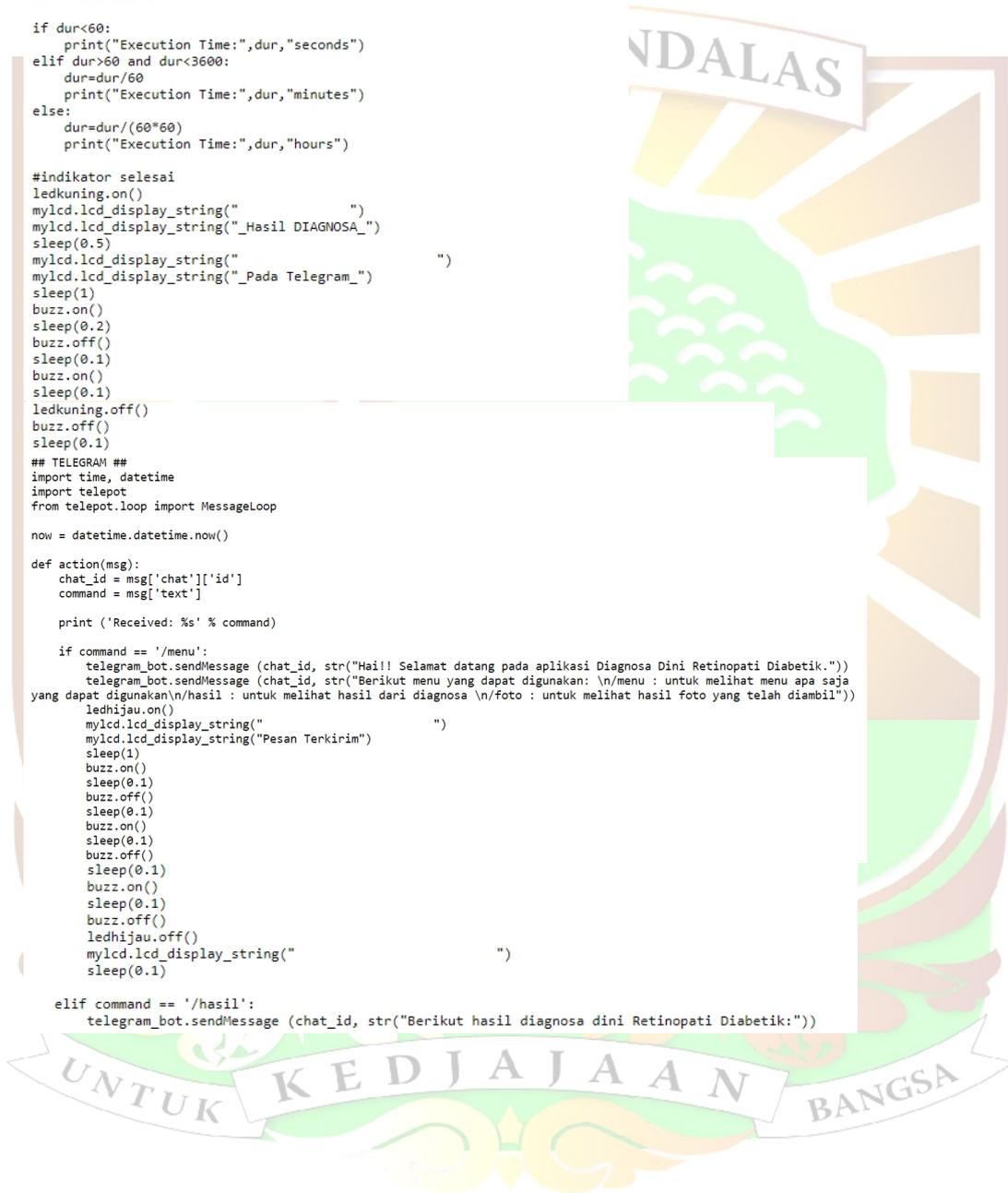
def action(msg):
    chat_id = msg['chat']['id']
    command = msg['text']

    print ('Received: %s' % command)

    if command == '/menu':
        telegram_bot.sendMessage (chat_id, str("Hai!! Selamat datang pada aplikasi Diagnosa Dini Retinopati Diabetik."))
        telegram_bot.sendMessage (chat_id, str("Berikut menu yang dapat digunakan: \n/menu : untuk melihat menu apa saja
yang dapat digunakan\n/hasil : untuk melihat hasil dari diagnosa \n/foto : untuk melihat hasil foto yang telah diambil"))
        ledhijau.on()
        mylcd.lcd_display_string(" ")
        mylcd.lcd_display_string("Pesan Terkirim")
        sleep(1)
        buzz.on()
        sleep(0.1)
        buzz.off()
        sleep(0.1)
        buzz.on()
        sleep(0.1)
        buzz.off()
        sleep(0.1)
        buzz.on()
        sleep(0.1)
        buzz.off()
        ledhijau.off()
        mylcd.lcd_display_string(" ")
        sleep(0.1)

    elif command == '/hasil':
        telegram_bot.sendMessage (chat_id, str("Berikut hasil diagnosa dini Retinopati Diabetik:"))

```



```

elif command == '/hasil':
    telegram_bot.sendMessage(chat_id, str("Berikut hasil diagnosa dini Retinopati Diabetik:"))

    for i, ret in enumerate(os.walk(test_path)):
        for i, filename in enumerate(ret[2]):
            if filename.startswith("."):
                continue

            print(ret[0] + '/' + filename)
            telegram_bot.sendMessage(chat_id, str(ret[0] + '/' + filename))
            result = predict(ret[0] + '/' + filename)
            answer = result
            if answer == 0:
                telegram_bot.sendMessage(chat_id, str("Prediksi: Normal"))
            elif answer == 1:
                telegram_bot.sendMessage(chat_id, str("Prediksi: RD"))

    telegram_bot.sendMessage(chat_id, str("Terimakasih telah memeriksakan keadaan mata Anda. Jika Anda penderita
    Diabetes, tetap periksakan keadaan mata Anda secara rutin.))
    ledhijau.on()
    mylcd lcd_display_string(" ")
    mylcd lcd_display_string("Pesan Terkirim")
    sleep(1)
    buzz.on()
    sleep(0.1)
    buzz.off()
    sleep(0.1)
    buzz.on()
    sleep(0.1)
    buzz.off()
    ledhijau.off()
    mylcd lcd_display_string(" ")
    sleep(0.1)

elif command == '/foto':
    telegram_bot.sendPhoto(chat_id, photo=open('/home/pi/Desktop/pic/Foto citra fundus.jpg', 'rb'))
    telegram_bot.sendMessage(chat_id, str("Foto Citra Fundus"))
    ledhijau.on()
    mylcd lcd_display_string(" ")
    mylcd lcd_display_string("Pesan Terkirim")
    sleep(1)
    buzz.on()
    sleep(0.1)
    buzz.off()
    sleep(0.1)
    buzz.on()
    sleep(0.1)
    buzz.off()
    sleep(0.1)
    buzz.on()
    sleep(0.1)
    buzz.off()
    ledhijau.off()
    mylcd lcd_display_string(" ")
    sleep(0.1)

telegram_bot = telepot.Bot('1451558209:AAE2U06FdxZQPguR7cRz7_HA4ZnVxHp7TdG')

MessageLoop(telegram_bot, action).run_as_thread()
print('Lets have a chat session...')

while 1:
    time.sleep(5)

```

UNTUK KEDJAJAAN BANGSA

Lampiran 2

HASIL WAWANCARA

Narasumber

Nama: dr. Havriza Vitresia, Sp.M (K)

Tempat/Tanggal Lahir: Padang / 27 April 1973

Pendidikan

S1 : Dokter Umum FK. Unand

S2 : Dokter Spesialis Mata FK. Unand

Pewawancara

Nama: Rahmatul Husna

Tempat/Tanggal Lahir: Serang / 30 September 1998

Pendidikan

S1 : Teknik Komputer FTI Unand

Waktu dan Tempat

Waktu: Selasa, 9 Maret 2021

Tempat: Poli mata, Rumah Sakit Universitas Andalas

Transkrip Hasil Wawancara

1. Apakah fotografi fundus termasuk pemeriksaan rutin untuk penderita diabetes militus?

Jawaban:

Diabetes militus merupakan penyakit yang mengenai banyak organ, salah satunya adalah mata. Pada mata, bagian yang paling mudah dikenai yaitu retina. Kelainan retina yang dikarenakan diabetes militus dinamakan retinopati diabetik.

Retinopati diabetik memiliki derajat yang bervariasi, mulai dari yang paling ringan hingga yang paling berat. Retinopati diabetik juga memiliki beberapa jenis dalam dunia medis, yaitu non-poliferatif dan poliperatif. Masing-masing tingkat keparahan memiliki penanganan yang berbeda. Misalnya, jika penanganan tingkat ringan, dilakukan *follow up* dan observasi terlebih dahulu

sebelum diberikan tindakan. Setiap tingkat keparahan memiliki waktu yang berbeda untuk *follow up* dan observasi. Tahap *follow up* dapat dilakukan dengan funduskopi, yaitu menilai kondisi retina untuk mengetahui sudah ada kelainan atau tidak. Jika sudah ada kelainan, ringan, sedang atau berat, barulah diambil tindakan.

Pemeriksaan foto fundus dianjurkan. Namun jika rumah sakit yang bersangkutan tidak memiliki alat untuk melakukan foto fundus, pemeriksaan dapat dilakukan menggunakan funduskopi oleh dokter mata.

2. Bagaimana cara melakukan pemeriksaan retinopati diabetik?

Jawaban:

Ada dua cara yang dapat dilakukan untuk melakukan pemeriksaan retinopati diabetik berdasarkan fundus. Yang pertama yaitu dengan funduskopi, yaitu alat pemeriksaan fundus yang dapat dilihat langsung atau pun tidak langsung ke retina oleh dokter saja. Namun, hasil yang didapat hanya berupa tulisan analisis hasil pemeriksaan. Yang kedua yaitu dengan melakukan foto fundus, yaitu dengan melihat, menilai, juga dapat mendokumentasikan retina. Sehingga, pada pemeriksaan retinopati diabetik cenderung menggunakan foto fundus. Karena dapat menilai fundus juga mendokumentasikan sebagai rekam medis ketika pasien kembali melakukan kontrol.

3. Apakah kegunaan lensa genggam yang pada fotografi fundus?

Jawaban:

Funduskopi adalah pemeriksaan yang kita lakukan untuk melihat fundus, funduskopi dibagi menjadi *direct* dan *indirect*. Lensa 20D merupakan lensa yang digunakan untuk funduskopi *indirect*. Funduskopi *direct* yaitu dilakukan menggunakan alat funduskopi *direct* dengan melihat langsung ke mata pasien dan kemudian hasil funduskopi dapat diberi penilaian. Funduskopi *direct* dilakukan dengan cara, menyalakan cahaya pada alat dan langsung diarahkan ke mata. Sedangkan Funduskopi *indirect*, dilakukan dengan meletakkan alat funduskopi *indirect* diatas (kepala), kemudian dinyalakan cahayanya, lalu diamati melalui lensa 20D.

4. Bagaimana proses fotografi fundus dengan menggunakan obat tetes dilatasi pupil?

Jawaban:

Posisi retina berada di dalam bola mata, lebih tepatnya pada dinding bola mata. Retina hanya dapat dilihat melalui pupil. Pupil merupakan bagian di tengah mata yang berbentuk bulat dan berukuran kecil, dapat membesar dan mengecil, tergantung intensitas cahaya yang masuk.

Obat tetes dilatasi diberikan untuk melebarkan pupil. Pada pemeriksaan fundus, akan lebih leluasa ketika melihat dari pupil yang membesar sehingga kita dapat melihat hingga daerah pinggir retina. Sehingga akan lebih baik ketika menggunakan obat tetes dilatasi ketika melakukan pemeriksaan fundus.

5. Apakah obat tetes dilatasi dapat ditemukan di luar rumah sakit?

Jawaban:

Obat tetes dilatasi tidak dapat ditemukan di luar rumah sakit atau di optik, karena digunakan untuk pemeriksaan bukan untuk pengobatan. Obat tetes dilatasi hanya tersedia di rumah sakit dan digunakan oleh dokter atau pun perawat.

6. Apakah setelah penggunaan obat tetes dilatasi pupil memiliki efek samping?

Jawaban:

Sejauh ini tidak ada efek samping dari penggunaan obat tetes dilatasi kepada pasien.

7. Berapakah lama proses dilatasi pupil setelah diberikan obat tetes dilatasi pupil hingga pupil membesar?

Jawaban:

Efek yang dirasa hanya pelebaran pupil dilatasi yang bertahan sekitar 3 sampai 4 jam, kemudian kembali lagi seperti semula.

8. Apakah pada proses fotografi fundus dapat dilakukan tanpa adanya pencahayaan?

Jawaban:

Pada setiap pemeriksaan harus disertai cahaya.

9. Apakah proses fotografi fundus dapat dilakukan tanpa menggunakan obat tetes dilatasi?

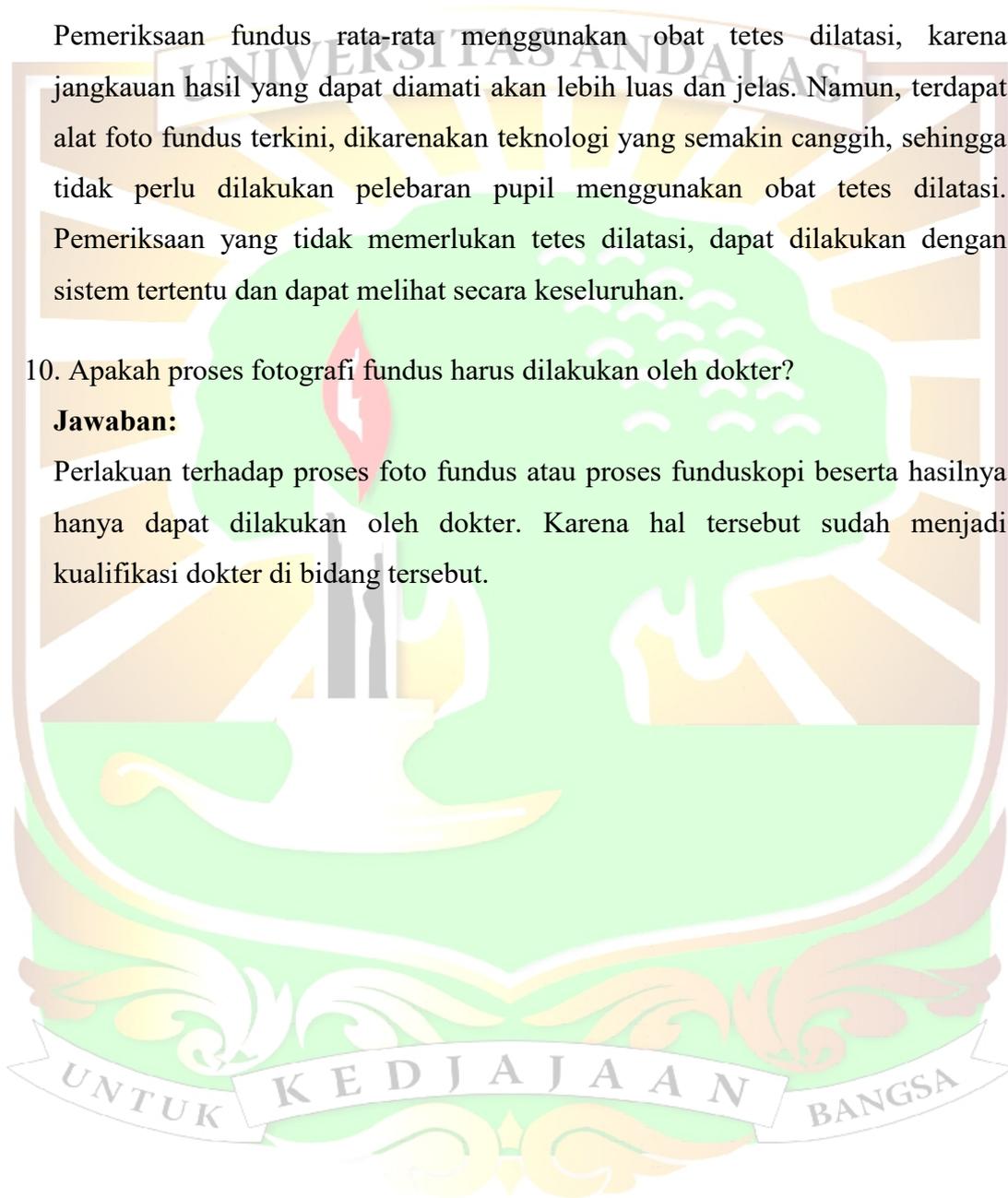
Jawaban:

Pemeriksaan fundus rata-rata menggunakan obat tetes dilatasi, karena jangkauan hasil yang dapat diamati akan lebih luas dan jelas. Namun, terdapat alat foto fundus terkini, dikarenakan teknologi yang semakin canggih, sehingga tidak perlu dilakukan pelebaran pupil menggunakan obat tetes dilatasi. Pemeriksaan yang tidak memerlukan tetes dilatasi, dapat dilakukan dengan sistem tertentu dan dapat melihat secara keseluruhan.

10. Apakah proses fotografi fundus harus dilakukan oleh dokter?

Jawaban:

Perlakuan terhadap proses foto fundus atau proses funduskopi beserta hasilnya hanya dapat dilakukan oleh dokter. Karena hal tersebut sudah menjadi kualifikasi dokter di bidang tersebut.



SURAT PERNYATAAN

Saya bertanda tangan di bawah ini:

Nama : Rahmatul Husna

NIM : 1611513011

Jurusan : Teknik Komputer

Menyatakan bahwa telah melakukan pemeriksaan *similarity* dengan Turnitin dengan *similarity index* <25%. Hal ini dapat dilihat pada lampiran yang saya sertakan bersama surat ini.

Demikian surat pernyataan ini saya buat agar dapat digunakan sebaik-baiknya.

Padang, 24 Juni 2021

Yang bertanda tangan,



Rahmatul Husna
1611513011

Lampiran

The screenshot displays a Turnitin report for a thesis. The main content area shows the title "SISTEM PORTABEL DIAGNOSA DINI RETINOPATI DIABETIK DENGAN MENERAPKAN ALGORITMA CONVOLUTIONAL NEURAL NETWORK" and the author's name "RAHMATUL HUSNA" with ID "1611513011". The report is from Universitas Andalas. The similarity score is 18%. The sources are listed as follows:

Rank	Source	Percentage
1	repo.unand.ac.id Sumber Internet	2%
2	dspace.uui.ac.id Sumber Internet	1%
3	epdf.pub Sumber Internet	1%
4	www.ncchta.org Sumber Internet	1%
5	libraryproceeding.telk... Sumber Internet	1%
6	www.scribd.com Sumber Internet	1%
7	Diserahkan ke Universit... Sumber Internet	1%