

**PENERAPAN MODEL CNN DALAM APLIKASI ANDROID DAN
PURWARUPA DETEKTOR DIABETES RETINOPATI**

TUGAS AKHIR



**YOHANES VIANNEY DANI TJOROAMINJAYA
312010013**

**PROGRAM STUDI TEKNIK INFORMATIKA
FAKULTAS TEKNOLOGI DAN DESAIN
UNIVERSITAS MA CHUNG
MALANG
2024**

LEMBAR PENGESAHAN

TUGAS AKHIR

**PENERAPAN MODEL CNN DALAM APLIKASI ANDROID DAN
PURWARUPA DETEKTOR DIABETES RETINOPATI**

Oleh:

YOHANES VIANNEY DANI TJOKROAMINJAYA

NIM. 312010013

dari:

PROGRAM STUDI TEKNIK INFORMATIKA

FAKULTAS TEKNOLOGI DAN DESAIN

UNIVERSITAS MA CHUNG

Telah dinyatakan lulus dalam melaksanakan Tugas Akhir sebagai syarat kelulusan
dan berhak mendapatkan gelar Sarjana Komputer

Dosen Pembimbing 1,



Windra Swastika, S.Kom., MT.,
Ph.D.
NIP. 20100006

Dosen Pembimbing 2,



Prof. Dr. Eng. Romy Budhi, S.T.,
M.T., M.Pd.
NIP. 20070035

Dekan Fakultas Sains dan Teknologi,



Prof. Dr. Eng. Romy Budhi, S.T., M.T., M.Pd.
NIP. 20070035

PERNYATAAN KEASLIAN TUGAS AKHIR

Dengan ini saya menyatakan bahwa isi sebagian maupun keseluruhan Tugas Akhir saya dengan “Penerapan Model CNN dalam Aplikasi Android dan Purwarupa Detektor Diabetes Retinopati” adalah benar-benar hasil karya intelektual mandiri, diselesaikan tanpa menggunakan bahan-bahan yang tidak diizinkan, dan bukan merupakan karya pihak lain yang saya akui sebagai karya sendiri.

Semua referensi yang dikutip maupun dirujuk telah ditulis secara lengkap pada daftar pustaka. Apabila ternyata pernyataan ini tidak benar, saya bersedia menerima sanksi sesuai peraturan yang berlaku.

Malang, 17 Juli 2024



Yohanes Vianney Dani

NIM. 312010013

PENERAPAN MODEL CNN DALAM APLIKASI ANDROID DAN PURWARUPA DETEKTOR DIABETES RETINOPATI

Yohanes Vianney Dani Tjokroaminjaya, Windra Swastika, Romy Budhi Widodo

Abstrak

Diabetes retinopati (DR) merupakan salah satu komplikasi dari diabetes melitus yang berdampak pada kesehatan mata. Komplikasi tersebut dapat mengakibatkan gangguan penglihatan hingga kebutaan. Meski demikian, biaya untuk diagnosis maupun pengobatan diabetes retinopati masih sangat tinggi, dengan estimasi biaya mencapai 2,4 miliar rupiah pada tahun 2017. Di samping beban biaya, pasien juga harus menghadapi waktu tunggu yang cukup lama untuk mendapatkan hasil diagnosis. Untuk mengatasi tantangan ini, telah dikembangkan sebuah aplikasi android yang bertujuan untuk menyediakan solusi deteksi diabetes retinopati yang lebih terjangkau dan fleksibel.

Dalam pengembangan aplikasi ini, dilakukan penelitian untuk mengevaluasi arsitektur *Convolutional Neural Network* (CNN) yang digunakan untuk mendeteksi diabetes retinopati. Penelitian ini menguji dua arsitektur CNN, yaitu *ResNet-50* dan *VGG-19*. Sebagai bagian dari penelitian ini, penggunaan alat oftalmoskop funduskopi digantikan dengan lensa 20D sebagai metode untuk memperoleh citra fundus retina.

Hasil dari penelitian menunjukkan bahwa aplikasi telah dikembangkan menggunakan arsitektur *VGG-19* yang sudah disederhanakan. Arsitektur memiliki akurasi data latih sebesar 95% dan akurasi aktual 87,661%. Aplikasi android ini menghadirkan dua fitur utama, yaitu deteksi citra melalui gambar dari memori *handphone* dan deteksi foto secara langsung. Namun, fitur deteksi citra secara langsung masih terkendala dikarenakan gagal dalam melakukan akuisisi citra secara langsung. Penelitian ini juga berhasil menetapkan kondisi optimal untuk pengambilan citra fundus secara langsung menggunakan lensa 20D.

Kata Kunci : *Diabetes Retinopati (DR), Fundus Retina, VGG-19, ResNet-50, Lensa 20D, Aplikasi Android*

IMPLEMENTING CNN MODELS IN AN ANDROID APPLICATION AND A PROTOTYPE FOR DIABETIC RETINOPATHY DETECTION

Yohanes Vianney Dani Tjokroaminjaya, Windra Swastika, Romy Budhi Widodo

Abstract

Diabetic retinopathy (DR) is a significant complication of diabetes mellitus that adversely affects eye health and can lead to vision impairment or even blindness. Despite the severity of this condition, the costs associated with both diagnosis and treatment are prohibitively high, with estimates reaching up to 2.4 billion rupiah in 2017. Additionally, patients often face lengthy wait times to receive diagnostic results. To address these challenges, an android application has been developed to provide a more accessible and cost-effective solution for diabetic retinopathy detection.

In the development of this application, research was conducted to evaluate the effectiveness of Convolutional Neural Network (CNN) architectures for detecting diabetic retinopathy. This study specifically tested two CNN architectures: ResNet-50 and VGG-19. For the acquisition of retinal fundus images, the traditional fundus camera was substituted with a 20D lens to capture retinal images.

The research findings indicate that the application successfully employed a simplified version of the VGG-19 architecture, achieving a training accuracy of 95% and an actual accuracy of 87.661%. The Android application features two primary functions: image detection from the phone's memory and direct photo detection. However, the direct photo detection feature encountered challenges related to successful image acquisition. The study also established optimal conditions for capturing fundus images using the 20D lens.

Keywords : *Diabetic Retinopathy (DR), Retinal Fundus, VGG-19, ResNet-50, 20D Lens, Android Application*

KATA PENGANTAR

Puji syukur ke hadirat Tuhan Yang Maha Esa atas rahmat dan berkat-Nya, sehingga Laporan Tugas Akhir ini dapat diselesaikan dengan sebaik-baiknya. Laporan ini menjelaskan mengenai hasil proyek Tugas Akhir yang sudah dikerjakan yang berjudul “Penerapan Model CNN dalam Aplikasi Android dan Purwarupa Detektor Diabetes Retinopati”.

Besar rasa terima kasih yang ingin diberikan kepada pihak-pihak yang telah membantu selama proses penyelesaian proyek Tugas Akhir, di antaranya :

1. Bapak Windra Swastika, S.Kom, MT, Ph.D selaku Dosen Pembimbing I Tugas Akhir;
2. Prof. Dr. Eng. Romy Budhi Widodo selaku Dekan Fakultas Teknologi dan Desain dan Dosen Pembimbing II Tugas Akhir;
3. Bapak Hendry Setiawan, ST, M.Kom selaku Kepala Program Studi Teknik Informatika Universitas Ma Chung sekaligus Dosen Penguji Tugas Akhir;
4. dr. Sony Agung Santoso, Sp.M selaku fasilitator dan pembimbing lapangan dalam pengambilan subjek;
5. Secara khusus partner dan teman yang telah membembantu secara langsung dalam memberikan tenaganya secara langsung di lapangan untuk proyek Tugas Akhir, Alicia Angelica Mulyatno dan Laurentius Adityas Tino;
6. Kedua orang tua dan teman-teman yang telah memberikan dukungan dan semangat selama mengerjakan proyek Tugas Akhir;

Besar harapan laporan ini dapat memberikan gambaran mengenai hasil proyek Tugas Akhir yang sudah dikembangkan dan dapat bermanfaat untuk penelitian selanjutnya. Demikian yang dapat disampaikan, penulis mengucapkan terima kasih.

Malang, 20 Juli 2024

Yohanes Vianney Dani T.

DAFTAR ISI

KATA PENGANTAR	i
DAFTAR ISI	ii
DAFTAR GAMBAR	v
DAFTAR TABEL	vii
BAB I Pendahuluan	1
1.1 Latar Belakang	1
1.2 Identifikasi Masalah	3
1.3 Batasan Masalah	3
1.4 Rumusan Masalah	3
1.5 Tujuan Penelitian	4
1.6 Manfaat Penelitian	4
1.7 Luaran Tugas Akhir	4
1.8 Sistematika Penulisan	5
BAB II Tinjauan Pustaka	6
2.1 Diabetes Retinopati	6
2.2 Kamera Fundus	7
2.3 Android Studio	9
2.4 <i>Deep Learning</i>	9
2.5 <i>Convolution Neural Network (CNN)</i>	9
2.5.1 <i>Convolution Layer</i>	10
2.5.2 <i>Pooling Layer</i>	10
2.5.3 VGG	11
2.5.4 <i>Residual Network (ResNet)</i>	12
2.6 Pencahayaan	13
2.7 Penelitian Terdahulu	14
BAB III Metode Penelitian	18
3.1 Alur Penelitian	18
3.2 Analisis Masalah	19
3.3 Studi Literatur	19
3.4 Perancangan Model CNN	20

3.4.1 Perancangan Model VGG-19	20
3.4.2 Perancangan Model <i>ResNet</i> -50	21
3.4.3 Perancangan Penyederhanaan Model	27
3.5 <i>Dataset</i>	28
3.5.1 Data 1	30
3.5.2 Data 2	30
3.5.3 Data 3	31
3.5.4 Data 4	32
3.6 Perancangan Aplikasi	33
3.7 Integrasi Perangkat Lunak dan Perangkat Keras	34
3.8 Pengujian	36
3.8.1 Pengujian Fitur Deteksi dari Citra Galeri ke Aplikasi	37
3.8.2 Pengujian Fitur Deteksi dari Citra Foto Cetak	37
3.8.3 Pengujian Prototipe dan Fitur Deteksi dari Citra Foto Langsung	37
3.8.4 Pengujian Hasil Deteksi Berdasarkan Parameter Uji	38
BAB IV Hasil dan Pembahasan	39
4.1 Penyederhanaan Arsitektur Model	39
4.1.1 Arsitektur VGG-19 Sederhana 1	39
4.1.2 Arsitektur VGG-19 Sederhana 2	40
4.2 Hasil Arsitektur Model	40
4.3 Pembuatan Aplikasi Android	47
4.4 Pengujian Deteksi dengan Aplikasi Android	52
4.4.1 Pengujian Deteksi Menggunakan Citra dari <i>Gallery</i>	52
4.4.2 Pengujian Deteksi dengan Kamera pada Citra Gambar Cetak	58
4.4.3 Pengujian Deteksi dengan Kamera pada Citra Retina Asli	62
4.4.3.1 Pengujian seperti rancangan awal	63
4.4.3.2 Peneliti dengan lensa 20D dan <i>smartphone</i> tanpa adaptor	64
4.4.3.3 Peneliti dengan <i>direct oftalmoskop</i> dan <i>smartphone</i>	65
4.4.3.4 Peneliti dengan lensa 78D dan disertai <i>smartphone</i>	65

4.4.4 Pengujian Kondisi Optimal Penangkapan Citra Fundus	67
BAB V Kesimpulan dan Saran	68
5.1 Kesimpulan	68
5.2 Saran	69
DAFTAR PUSTAKA	vi

DAFTAR GAMBAR

Gambar 2.1 Perbandingan citra retina normal dibandingkan diabetes retinopati	6
Gambar 2.2 Pembagian kelas diabetes retinopati	7
Gambar 2.3 Oftalmoskop funduskopi	8
Gambar 2.4 Ilustrasi proses konvolusi	10
Gambar 2.5 Ilustrasi konsep <i>pooling layer</i>	11
Gambar 2.6 Arsitektur VGG	11
Gambar 2.7 Blok residual fondasi arsitektur <i>ResNet</i>	12
Gambar 2.8 Arsitektur <i>ResNet-50</i>	13
Gambar 2.9 Lux meter	13
Gambar 3.1 Alur penelitian	18
Gambar 3.2 <i>Layer</i> dalam arsitektur VGG-19	21
Gambar 3.3 <i>Layer</i> dalam arsitektur <i>ResNet-50</i> (bagian 1)	22
Gambar 3.4 <i>Layer</i> dalam arsitektur <i>ResNet-50</i> (bagian 2)	23
Gambar 3.5 <i>Layer</i> dalam arsitektur <i>ResNet-50</i> (bagian 3)	24
Gambar 3.6 <i>Layer</i> dalam arsitektur <i>ResNet-50</i> (bagian 4)	25
Gambar 3.7 <i>Layer</i> dalam arsitektur <i>ResNet-50</i> (bagian 5)	26
Gambar 3.8 Peringatan batas ukuran maksimal model pada Android Studio	28
Gambar 3.9 Perbandingan retina <i>No DR</i> dengan keempat kelas lainnya	30
Gambar 3.10 Contoh citra yang dianggap rusak	31
Gambar 3.11 Contoh citra <i>Proliferative DR</i> yang digandakan	32
Gambar 3.12 Contoh citra data 4	32
Gambar 3.13 Alur pembuatan aplikasi android	33
Gambar 3.14 Rancangan <i>interface</i>	33
Gambar 3.15 Oftalmoskop funduskopi adaptor <i>smartphone</i>	36
Gambar 3.16 Oftalmoskop funduskopi yang terintegrasi dengan <i>smartphone</i>	36
Gambar 4.1 Arsitektur VGG-19 sederhana 1	39
Gambar 4.2 Arsitektur VGG-19 sederhana 2	40
Gambar 4.3 <i>Confusion matrix</i> P1	43
Gambar 4.4 <i>Confusion matrix</i> P2	43
Gambar 4.5 <i>Confusion matrix</i> P3 dan P4	43

Gambar 4.6 <i>Confusion matrix</i> P5	43
Gambar 4.7 <i>Confusion matrix</i> P6	43
Gambar 4.8 <i>Confusion matrix</i> P7	43
Gambar 4.9 <i>Confusion matrix</i> P8	44
Gambar 4.10 <i>Confusion matrix</i> P9	44
Gambar 4.11 Tata letak tampilan menu utama aplikasi	47
Gambar 4.12 Potongan kode fungsi interaksi tombol dan tampilan	47
Gambar 4.13 Potongan kode fungsi <i>onActivityResult</i>	48
Gambar 4.14 Potongan kode fungsi prediksi dan pemanggilan model (bagian 1)	49
Gambar 4.15 Potongan kode fungsi prediksi dan pemanggilan model (bagian 2)	50
Gambar 4.16 Tampilan menu utama aplikasi	51
Gambar 4.17 Tampilan aplikasi tombol “ <i>Select Image</i> ”	51
Gambar 4.18 Tampilan aplikasi tombol “ <i>Capture Image</i> ”	51
Gambar 4.19 Tampilan aplikasi tombol “Prediksi”	52
Gambar 4.20 Penerapan perangkat keras <i>smartphone</i> dan lensa 20D	62
Gambar 4.21 Penangkapan citra fundus dengan lensa 20D, adaptor, dan <i>smartphone</i>	63
Gambar 4.22 Penangkapan citra fundus dengan lensa 20D dan <i>smartphone</i>	64
Gambar 4.23 Kondisi ruangan pengambilan data sampel	64
Gambar 4.24 Penangkapan citra fundus dengan <i>direct oftalmoskop</i> dan <i>smartphone</i>	65
Gambar 4.25 Penangkapan citra fundus dengan lensa 78D, <i>slit lamp</i> dan <i>smartphone</i>	66
Gambar 4.26 Kondisi ruangan pengambilan citra fundus dengan lensa 78D, <i>slit lamp</i> dan <i>smartphone</i>	66

DAFTAR TABEL

Tabel 2.1 Tabel perbandingan besarnya lux dan gambaran kecerahan ruangan	14
Tabel 3.1 Tabel pelabelan jenis data dan perbedaannya	29
Tabel 3.2 Tabel <i>Dataset</i> 1	30
Tabel 3.3 Tabel <i>Dataset</i> 1 hasil modifikasi pada kelas <i>No DR</i> dan <i>Proliferative</i>	33
Tabel 3.4 Tabel spesifikasi <i>smartphone</i>	35
Tabel 3.5 Tabel spesifikasi oftalmoskop funduskopi adaptor <i>smartphone</i>	36
Tabel 3.6 Tabel spesifikasi lux meter	38
Tabel 3.7 Tabel pengujian parameter terhadap hasil pendeteksian	38
Tabel 4.1 Tabel percobaan pembuatan arsitektur dan hasilnya (bagian 1)	41
Tabel 4.2 Tabel percobaan pembuatan arsitektur dan hasilnya (bagian 2)	42
Tabel 4.3 Tabel keterangan untuk folder data yang diuji	44
Tabel 4.4 Tabel perbandingan hasil prediksi model CNN	45
Tabel 4.5 Tabel rata-rata prediksi benar dari model yang dipilih	46
Tabel 4.6 Tabel hasil deteksi citra No_DR dari memori <i>handphone</i> (bagian 1)	53
Tabel 4.7 Tabel hasil deteksi citra No_DR dari memori <i>handphone</i> (bagian 2)	54
Tabel 4.8 Tabel hasil deteksi citra No_DR dari memori <i>handphone</i> (bagian 3)	55
Tabel 4.9 Tabel hasil deteksi citra DR dari memori <i>handphone</i> (bagian 1)	55
Tabel 4.10 Tabel hasil deteksi citra DR dari memori <i>handphone</i> (bagian 2)	56
Tabel 4.11 Tabel hasil deteksi citra DR dari memori <i>handphone</i> (bagian 3)	57
Tabel 4.12 Tabel hasil deteksi citra No_DR dari gambar cetak (bagian 1)	58
Tabel 4.13 Tabel hasil deteksi citra No_DR dari gambar cetak (bagian 2)	59
Tabel 4.14 Tabel hasil deteksi citra DR dari kamera pada gambar cetak (bagian 1)	60
Tabel 4.15 Tabel hasil deteksi citra DR dari kamera pada gambar cetak (bagian 2)	61
Tabel 4.16 Tabel variabel uji kondisi optimal penangkapan citra fundus	67
Tabel 4.17 Tabel hasil uji kondisi optimal penangkapan citra fundus	67

BAB I

Pendahuluan

1.1 Latar Belakang

Diabetes merupakan suatu kondisi kronis yang disebabkan oleh tingginya kadar glukosa dalam darah. Hal ini dapat mengakibatkan kerusakan parah pada pembuluh darah dan berbagai masalah kesehatan serius. Menurut World Health Organization (WHO), diabetes saat ini menempati peringkat ketujuh sebagai penyebab kematian utama di seluruh dunia, dengan lebih dari 422 juta individu yang terkena penyakit ini (Zhou, dkk. 2016). Menurut data tersebut diperkirakan bahwa jumlah penderita diabetes akan terus meningkat di masa depan. Di Indonesia, diabetes menduduki peringkat ketiga dalam daftar penyebab kematian setelah stroke dan penyakit jantung, dengan jumlah penderita yang saat ini mencapai lebih dari 10 juta orang, dan proyeksi menyebutkan angka ini bisa mencapai 30 juta orang pada tahun 2030 (P2PTM Kemenkes RI. 11 Desember 2018). Berdasarkan data yang diperoleh dari artikel karya Wang beserta rekan-rekannya (2017), sekitar 80% dari penderita diabetes mengalami komplikasi yang disebut diabetes retinopati (DR). Diabetes retinopati suatu kondisi yang memengaruhi pembuluh darah di retina dan dapat menyebabkan masalah penglihatan serta berpotensi mengakibatkan kebutaan. Kondisi tersebut menjadikan diabetes retinopati sebagai sebuah masalah kesehatan masyarakat yang sangat mahal biayanya untuk penanganannya, dengan perkiraan biaya mencapai 2,4 miliar dolar pada tahun 2017 dan diperkirakan akan meningkat menjadi 8,9 miliar dolar pada tahun 2025 (Sasongko, dkk. 2020).

Diabetes retinopati (DR) dapat dikelompokkan menjadi dua jenis utama, yaitu *nonproliferative diabetic retinopathy* (NPDR) dan *proliferative diabetic retinopathy* (PDR). Dalam NPDR, tanda-tanda awal mencakup adanya *mikroaneurisma* (pembengkakan dan peningkatan kebocoran pada pembuluh darah kecil di retina), *retinal hemorrhages* (pecahnya pembuluh darah kecil di retina dan pelepasan darah ke dalam retina), serta *capillary nonperfusion* (penyumbatan pada pembuluh darah kecil di retina). Pada PDR, terjadi iskemia retina yang lebih parah,

yang mengakibatkan pertumbuhan pembuluh darah baru di permukaan retina atau diskus optikus.

Dengan biaya pengobatan yang mahal, maka solusi yang masuk akal adalah dengan melakukan pencegahan terhadap gejala diabetes retinopati. Pencegahan tersebut dapat dilakukan dengan menjalani pemeriksaan mata secara rutin guna mendeteksi dini tanda-tanda kerusakan retina. Deteksi dan pemeriksaan ini biasanya dilakukan secara manual oleh dokter mata, dengan menggunakan citra fundus (bagian belakang mata) yang diperoleh melalui penggunaan peralatan kamera fundus *nonmydriatic* yang mahal. Proses pemeriksaan manual citra fundus ini memerlukan waktu yang cukup lama, memerlukan tenaga ahli yang terlatih, termasuk dokter spesialis mata, serta memiliki biaya tinggi dan berisiko terhadap potensi kesalahan. Salah satu pendekatan untuk mengakuisisi citra fundus dengan biaya yang terjangkau adalah dengan memanfaatkan lensa khusus yang disematkan di kamera pada *smartphone*. Pengujian dan evaluasi perlu dilakukan untuk memastikan resolusi yang optimal dan fitur autofokus yang baik, sehingga gambar fundus mata yang dihasilkan dapat memiliki detail dan kejelasan yang memadai.

Pengembangan kamera fundus berbasis *smartphone* diinisiasi melalui penelitian Chen, Chang, Zhao, dan Liu (2021) yang menghasilkan sebuah purwarupa perangkat fotografi fundus berbasis *smartphone*. Namun, citra fundus yang didapatkan pada penelitian ini masih memerlukan *editing* menggunakan *software image editing*. Pada penelitian tersebut diketahui bahwa citra fundus masih perlu dilakukan evaluasi terhadap pengaturan pencahayaan, resolusi *smartphone*, dan jarak pengambilan citra yang optimal. Chalam, Chamchikh, dan Gasparian (2022) juga telah mengembangkan prototipe kamera fundus yang menggunakan *smartphone* dengan lensa 25D. Meskipun dari segi komponen sudah lebih unggul dari pada yang digunakan dalam penelitian Chen, Chang, Zhao, dan Liu (2021), namun belum ada pengujian terhadap parameter-parameter yang diperlukan untuk mengoptimalkan pengambilan citra. Prototipe kamera fundus berbasis *smartphone* perlu diuji dengan berbagai parameter, seperti pencahayaan, jarak, dan autofokus, guna memastikan bahwa citra fundus yang dihasilkan memiliki kualitas yang setara dengan kamera fundus konvensional.

Dengan menggabungkan berbagai penelitian terkait pembuatan model CNN untuk mendeteksi diabetes retinopati melalui citra retina serta penelitian terkait pengujian purwarupa alat pengambil citra melalui *smartphone*, maka penelitian kali ini akan difokuskan pada proses untuk memanfaatkan model yang sudah ada ke dalam purwarupa aplikasi fotografi retina berbasis *smartphone* supaya dapat mendeteksi lebih optimal secara perangkat dan pengaturannya.

1.2 Identifikasi Masalah

Persoalan yang ditemukan dalam penelitian ini adalah ketiadaan aplikasi *mobile* untuk diagnosis diabetes retinopati, kekurangan penelitian yang memadai terkait parameter yang diperlukan saat pengambilan citra secara langsung dalam pendeteksian, serta upaya dalam mengetahui kondisi optimal parameter terkait.

1.3 Batasan Masalah

Masalah dalam proyek Tugas Akhir ini dibatasi sebagai berikut:

1. Penelitian terfokus pada proses penyesuaian model CNN untuk aplikasi, pembuatan aplikasi *mobile*, rancangan dan pengintegrasian purwarupa, serta pengujian parameter kondisi optimal pengambilan citra;
2. Arsitektur yang digunakan hanya memanfaatkan VGG-19, *ResNet*-50, serta arsitektur modifikasi dari keduanya;
3. Aplikasi akan diintegrasikan pada *smartphone* android;
4. Pendeteksian dilakukan pada retina mata dan fungsinya spesifik untuk melakukan diagnosis terkait diabetes retinopati.

1.4 Perumusan Masalah

Masalah-masalah yang ditemukan dalam penelitian ini yaitu sebagai berikut:

1. Bagaimana arsitektur dan kondisi model CNN yang ditanamkan pada aplikasi?
2. Bagaimana desain dari aplikasi *smartphone* yang dibuat?
3. Bagaimana rancangan dan perwujudan purwarupa kamera fundus yang terintegrasi dengan *smartphone* untuk mendeteksi diabetes retinopati?
4. Bagaimana keadaan optimal untuk mendeteksi diabetes retinopati pada retina mata berdasarkan variabel yang sudah ditentukan?

1.5 Tujuan

Tujuan dari penelitian ini adalah sebagai berikut:

1. Mengetahui arsitektur dan kondisi model CNN yang ditanamkan pada aplikasi;
2. Mengetahui desain dari aplikasi *smartphone* yang dibuat;
3. Mengetahui rancangan dan perwujudan purwarupa kamera fundus yang terintegrasi dengan *smartphone* yang digunakan untuk mendeteksi diabetes retinopati;
4. Mengetahui keadaan optimal untuk mendeteksi diabetes retinopati pada retina mata berdasarkan variabel yang sudah ditentukan.

1.6 Manfaat

Manfaat yang kelask akan diperoleh melalui proyek ini sebagai berikut:

1. Bagi mahasiswa

Melalui proyek ini, diharapkan mahasiswa dapat menerapkan materi yang dipelajari selama melakukan perkuliahan di Universitas Ma Chung. Selain itu, proyek ini juga menambah pengalaman dan memperluas wawasan mahasiswa sebagai bekal untuk masuk ke dunia kerja maupun jenjang akademis yang lebih tinggi.

2. Bagi Universitas Ma Chung, khususnya Program Studi Teknik Informatika

Dengan adanya proyek ini, Universitas Ma Chung, khususnya Program Studi Teknik Informatika dapat mempersiapkan mahasiswa untuk menjadi lulusan yang kompeten. Selain itu, besar harapan bahwa penelitian ini masih bisa menjadi bekal bagi adik tingkat di jurusan yang sama.

1.7 Luaran Penelitian

Luaran dari penelitian ini adalah sebuah jurnal hasil penelitian, aplikasi pendeteksian retina mata penderita diabetes retinopati berbasis android, serta rancangan dari purwarupa perangkat.

1.8 Sistematika Penulisan

Sistematika penulisan laporan Tugas Akhir ini dibagi menjadi beberapa bab seperti berikut:

BAB I Pendahuluan

Bab satu ini berisi latar belakang, identifikasi masalah, batasan masalah, perumusan masalah, tujuan, manfaat, luaran penelitian, dan sistematika penulisan.

BAB II Tinjauan Pustaka

Bab dua ini terdiri dari uraian mengenai teori dari diabetes retinopati, kamera fundus, serta alat bantu untuk pengerjaan program seperti Android Studio, *machine learning*, *deep learning*, *convolution neural network*, dan penelitian terdahulu.

BAB III Analisis dan Perancangan Sistem

Bab tiga ini berisi tentang tahapan perancangan proyek Tugas Akhir, yang terdiri dari alur penelitian, analisis masalah, studi literatur, perancangan aplikasi, integrasi perangkat lunak dan perangkat keras, serta pengujian.

BAB IV Hasil dan Pembahasan

Bab empat ini berisi hasil dan pembahasan dari proyek Tugas Akhir yang sudah dikerjakan, yang terdiri dari hasil dari tampilan aplikasi serta hasil pengujian pendeteksian beserta analisisnya.

BAB V Kesimpulan dan Saran

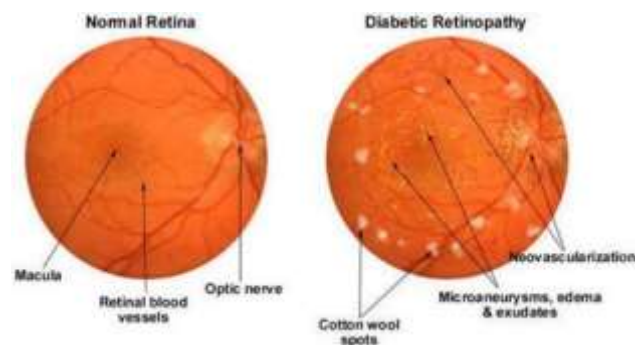
Bab lima ini berisi simpulan dari hasil proyek Tugas Akhir yang telah dikerjakan beserta saran terkait hal-hal yang dapat diperbaiki dari purwarupa maupun aplikasi untuk dilanjutkan pada penelitian selanjutnya.

BAB II

Tinjauan Pustaka

2.1 Diabetes Retinopati

Dikutip dari penjelasan Sabrina dan Buditjahjanto (2017) serta Pratama, Atmaja, dan Fauzi (2016), diabetes retinopati (DR) merupakan salah satu komplikasi yang terjadi akibat diabetes melitus. Kondisi tersebut diakibatkan peningkatan kadar glukosa secara berlebih dalam darah sehingga pembuluh darah kapiler kecil di retina melemah dan pecah (Subarkah, 2020). Selanjutnya, darah akan mengalami kebocoran dan menyebabkan penebalan jaringan, pembengkakan, dan pendarahan yang luas (Subarkah, 2020). Secara umum, diabetes retinopati berdampak langsung pada gangguan penglihatan dan bila terlambat ditangani dapat berakibat pada kebutaan secara permanen. Hingga saat ini, proses penyembuhan penyakit diabetes retinopati masih menjadi tantangan karena sebagian besar orang yang mengidap cenderung melakukan pemeriksaan ketika kondisinya sudah mencapai tahap berbahaya dan hal ini tidak terlepas dari fakta karakteristik dari penyakit DR yang tidak menunjukkan gejala awal yang (Subarkah, 2020).

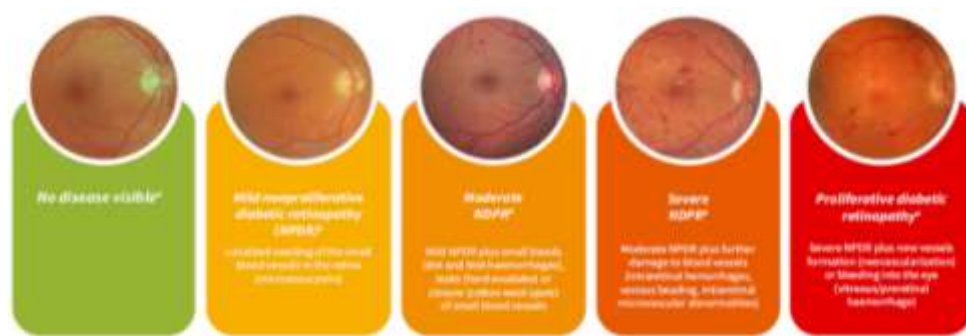


Gambar 2.1 Perbandingan citra retina normal dibandingkan diabetes retinopati

Sumber: Pratama, Atmaja, dan Fauzi, 2016

Dampak yang terlihat melalui visual dari kebocoran pembuluh darah ditunjukkan melalui perubahan pada retina seperti perubahan vena, *cotton wool spots*, *mikroaneurisma*, eksudat keras, dan pendarahan (Subarkah, 2020). Diperjelas kembali oleh Sabrina dan Buditjahjanto (2017) bahwa gejala yang tampak pada penderita diabetes retinopati (DR) antara lain *mikroaneurisma*, pendarahan dalam rongga vena (*hemorrhages*), eksudat keras dan lunak (lemak

protein), serta pembentukan pembuluh darah baru (*neovascularis*). Tingkat keparahan gejala tersebut dapat berfungsi sebagai petunjuk untuk menentukan tahap diabetes retinopati pada penderita. Berdasarkan penggolongan oleh *International Diabetic Retinopathy* (ICDR), tahap tersebut dibagi dalam tiga fase, yaitu *non-proliferative diabetic retinopathy* (NPDR), *proliferative diabetic retinopathy* (PDR), serta *macula edema* (ME) (Vitriana, 2023). NPDR dan DME kemudian dibagi lagi menjadi 3 tingkatan yaitu, ringan (*mild*), sedang (*moderate*), dan berat (*serve*). Penggolongan tersebut dapat diperhatikan pada Gambar 2.2.



Gambar 2.2 Pembagian kelas diabetes retinopati

Sumber: Vitriana, 2023

2.2 Kamera Fundus

Kamera fundus atau oftalmoskop adalah alat yang dipergunakan untuk mengambil foto bagian dasar mata atau dikenal dengan nama fundus (Mishra & Koushik, 2023). Sementara pengertian tambahan menurut Dhanorkar (2021), Kamera fundus adalah mikroskop khusus berdaya rendah yang dipasang pada kamera yang digunakan untuk memeriksa struktur seperti cakram optik, retina, dan lensa mata. Hasil dari fotografi fundus menangkap gambar retina, kepala saraf optik, makula, pembuluh darah retina, koroid, dan cairan vitreus. Fotografi fundus bermanfaat untuk melakukan diagnosis dini melalui pupil (Mishra & Koushik, 2023). Oftalmoskop dibedakan menjadi dua yaitu oftalmoskop funduskopi langsung (*direct ophthalmoscope fundoscopy*) dan oftalmoskop funduskopi tidak langsung (*indirect ophthalmoscope fundoscopy*), dapat dilihat pada Gambar 2.3.

Keunggulan dari penggunaan kamera fundus dalam proses deteksi kelainan pada mata adalah sebagai berikut (Dhanorkar, 2021):

- a. Hanya membutuhkan waktu satu sampai dua menit untuk proses pendapatkan gambar;
- b. Lebih mudah memvisualisasikan detail retina melalui foto fundus stereoskopis dibandingkan memeriksa mata secara langsung;
- c. Sering kali beberapa bagian mata terlihat lebih detail dalam fotografi fundus dibandingkan pemeriksaan mata lainnya seperti angiografi, dan *fluorescein*;
- d. Memungkinkan diagnosis dini akurat dan berkala untuk mempelajari perubahan halus pada mata untuk menentukan terapi dan pengobatan yang sesuai.

Beberapa kendala saat pengambilan fotografi fundus yang perlu diperhatikan adalah sebagai berikut (Mishra & Koushik, 2023):

- a. Warna oranye atau bulan sabit cerah di tepi foto disebabkan oleh iris yang menyala, khususnya pada pupil yang melebar dengan buruk;
- b. Bulu mata menciptakan selubung kabur di beberapa area gambar. Dalam hal ini, kelopak mata harus dibiarkan terbuka sehingga bulu mata tidak menghalangi cahaya atau menghalangi jalur cahaya;
- c. Jarak kamera yang terlalu dekat atau jauh dari mata akan menghilangkan detail dari retina. Jarak antara mata dan kamera harus optimal untuk mendapatkan rona warna fundus yang paling dalam;



(a)



(b)

Gambar 2.3 Oftalmoskop funduskopi a) langsung; b) tidak langsung

- d. Noda yang menghalangi pandangan lensa akan menghasilkan titik di tempat yang sama. Hal ini biasanya disebabkan oleh sentuhan lensa objektif pada wajah, hidung, atau bulu mata pasien, yang hilang setelah kotoran atau noda dibersihkan.

2.3 Android Studio

Android Studio merupakan salah satu *Integrated Development Environment* (IDE) yang dapat digunakan untuk membangun aplikasi android, yang berbasiskan dari IntelliJ IDEA. Selain menawarkan kode editor dan fitur pengembang yang canggih dari IntelliJ, Android Studio juga menghadirkan berbagai fitur tambahan yang dapat meningkatkan efisiensi dalam pengembangan aplikasi android.

Android merupakan salah satu sistem operasi yang banyak diminati karena sifatnya yang *open source*, yang memungkinkan pengguna untuk melakukan pengembangan (Akbar, Sardjono, Cahyanti, & Swedia, 2020). Android merupakan generasi terbaru dari platform *mobile* berbasis Linux yang mencakup sistem operasi, *middleware*, dan beragam aplikasi (Hansun, Kristanda, & Saputra, 2018).

2.4 Deep Learning

Deep learning merupakan salah satu bidang ilmu dari *machine learning* yang sering digunakan dalam melakukan pengklasifikasian secara digital (Akbar, Sardjono, Cahyanti, & Swedia, 2020). Pendapat tersebut dipertegas oleh Fredonovale (2022) bahwa *deep learning* adalah salah satu cabang dari *machine learning* yang menggunakan jaringan saraf dengan tiga atau lebih lapisan. Jaringan saraf ini mencoba untuk meniru fungsi otak manusia serta memungkinkan pemrosesan data dalam jumlah besar dan meningkatkan akurasi prediksi.

Deep learning telah meningkatkan perkembangan aplikasi dan layanan dalam kecerdasan buatan (AI). Teknologi ini memungkinkan otomatisasi tugas analitis dan fisik tanpa perlu campur tangan manusia. *Deep learning* berperan penting dalam produk dan layanan sehari-hari, seperti asisten digital, kontrol TV dengan suara, dan deteksi penipuan kartu kredit, serta berkontribusi pada teknologi yang sedang berkembang, seperti mobil *self-driving*.

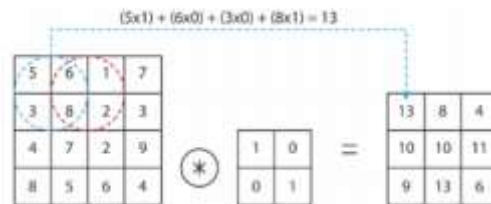
2.5 Convolution Neural Network (CNN)

Convolution Neural Network (CNN) adalah perkembangan dari *Multilayer Perceptron* (MLP) dalam jaringan saraf tiruan, khususnya untuk memproses data berupa citra dua dimensi (Fuadah, Saidah, Khofiya, Magdalena, & Ubaidullah, 2022). CNN dapat memecahkan permasalahan *object detection* dan *object recognition* (Akbar, Sardjono, Cahyanti, & Swedia, 2020).

CNN dapat mengidentifikasi objek berkat dua bagian utama dalam proses pembelajarannya, yaitu mengekstraksi ciri khusus dari citra (*feature extraction*) dan klasifikasi berdasarkan ciri-ciri tersebut (*classification*) (Fuadah, Saidah, Khofiya, Magdalena, & Ubaidullah, 2022). Proses ekstraksi pada CNN menggunakan proses konvolusi dengan menggerakkan sebuah *kernel* konvolusi (*filter*) berukuran tertentu ke sebuah gambar (Fredonovalle, 2022). CNN memiliki neuron-neuron tersusun disusun dalam tiga dimensi, yaitu panjang, lebar, dan tinggi sehingga membuatnya sangat efektif dan efisien untuk menganalisis gambar (Primartha, 2018).

2.5.1 Convolution Layer

Convolution layer merupakan lapisan utama dalam CNN. Pada *convolution layer*, proses konvolusi dilakukan antara citra dan *filter* berbentuk matriks dengan ukuran tertentu untuk menghasilkan *feature map* seperti yang ditunjukkan pada Gambar 2.4 (Fuadah, Saidah, Khofiya, Magdalena, & Ubaidullah, 2022). Konvolusi dilakukan dengan operasi aljabar linear yang mengalikan perkalian matriks *filter* dengan citra yang akan diproses (Akbar, Sardjono, Cahyanti, & Swedia, 2020).



Gambar 2.4 Ilustrasi proses konvolusi

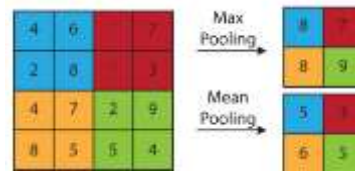
Sumber: Fuadah, Saidah, Khofiya, Magdalena, & Ubaidullah, 2022

Convolution layer dikhususkan untuk mengolah data dua dimensi sebab *convolution layer* menjadi bagian terpenting dalam membentuk kedalaman suatu fitur (Fredonovalle, 2022). Misalnya, jika gambar berukuran 32 x 32 x 3, maka dapat diartikan 32 x 32 merupakan ukuran dari *convolutional layer* sementara angka 3 merupakan jumlah lapisan warna.

2.5.2 Pooling Layer

Pooling layer berfungsi untuk mengurangi *input* secara spasial, yaitu dengan mengurangi jumlah parameter atau ukuran *feature map* melalui proses *down-sampling* (Fredonovalle, 2022). Sementara itu menurut Akbar, Sardjono, Cahyanti, dan Swedia (2020), *Pooling layer* berfungsi untuk

mengurangi ukuran *feature map* melalui proses *down-sampling*. Terdapat beberapa jenis *pooling layer*, namun yang paling umum digunakan adalah *max pooling* dan *average pooling* atau *mean pooling*.

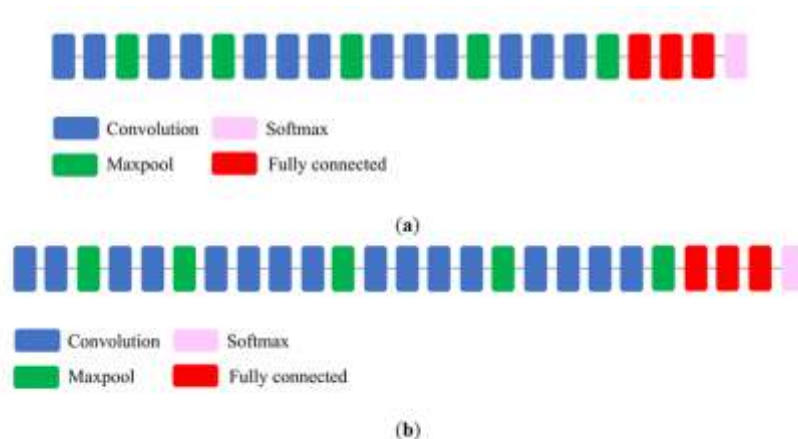


Gambar 2.5 Ilustrasi konsep *pooling layer*

Sumber: Fuadah, Saidah, Khofiya, Magdalena, & Ubaidullah, 2022

Max pooling merupakan lapisan yang digunakan untuk mengambil nilai maksimal dari setiap *grid* pada sebuah citra (Akbar, Sardjono, Cahyanti, & Swedia, 2020). Sementara *average pooling* atau *mean pooling* merupakan lapisan yang digunakan untuk mengambil rata-rata dari setiap *grid* pada sebuah citra (Akbar, Sardjono, Cahyanti, & Swedia, 2020). Nilai yang diambil tersebut, nantinya akan digunakan untuk menyusun matriks baru (Fuadah, Saidah, Khofiya, Magdalena, & Ubaidullah, 2022). Gambar 2.4 merupakan ilustrasi dari konsep *pooling layer*.

2.5.3 VGG



Gambar 2.6 a) Arsitektur VGG-16 dan b) Arsitektur VGG-19

Sumber: Mahdianpari, Salehi, Rezaee, Mohammadimanesh, & Zhang, 2018

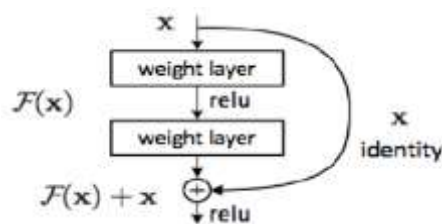
VGG merupakan salah satu arsitektur pada CNN yang dibuat oleh grup bernama *Visual Geometry Group* di Oxford (Marcella, Yohannes, & Devella, 2022). VGG memiliki beberapa variasi dan variasi yang terkenal adalah VGG-16 dan VGG-19. Menurut Pradeep (2023), VGG-16 merupakan model

dengan 16 lapisan bobot dalam arsitekturnya yang terdiri dari 13 *convolution layer* dan 3 *fully connected layer* sementara VGG-19 merupakan model dengan 19 lapisan bobot yang terdiri dari 16 *convolution layer* dan 3 *fully connected layer* (Pradeep, 2023). Perbandingan dari kedua arsitektur tersebut dapat dilihat pada Gambar 2.6. Keduanya memiliki *convolution layer* dengan *filter* 3x3 kecil, dan jaringan ini menggunakan lapisan *max pooling* 2x2. Terdapat 5 lapisan *max pooling* yang mengikuti beberapa lapisan konvolusi. VGG menggunakan konsep semakin banyak layer akan menghasilkan akurasi yang semakin baik (Setiawan, 2019).

2.5.4 Residual Network (ResNet)

Kedalaman yang tinggi dalam arsitektur CNN adalah salah satu hal penting supaya memiliki performa yang baik, namun di sisi lain menimbulkan kekurangan yaitu *vanishing gradient problem*. *Vanishing gradient problem* adalah kondisi di mana gradien yang dipelajari oleh model tidak dapat mencapai layer pertama karena mengalami perkalian berulang, sehingga CNN tidak dapat belajar dari *error* yang telah dikalkulasi (Hu, Huber, Anumula, & Liu, 2018).

Teori tersebut mendukung kekuatan dari kegunaan arsitektur *ResNet*. Pada tahun 2016, He, Zhang, Ren, dan Sun (2016) menggunakan struktur *residual block*. Blok ini terdapat pada setiap lapis dan menjadi komponen dasar arsitektur CNN *Resnet*. Konsep dari *residual block* dapat dilihat pada Gambar 2.7.

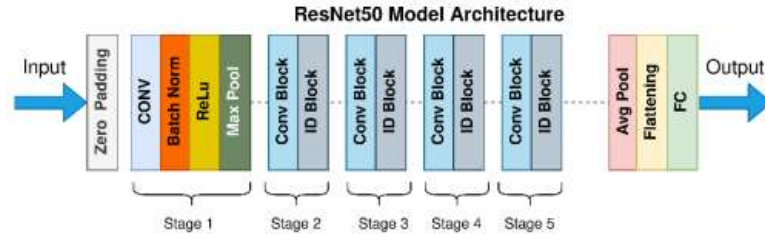


Gambar 2.7 Blok residual fondasi arsitektur *ResNet*

Sumber: Riyadi, Wardhani, Wulandari, & Widayati, 2022

Berdasarkan pernyataan Rikendry dan Maharil (2022), *Residual Network (ResNet)* merupakan model yang menggunakan kerangka pembelajaran residual yang mendalam. Sementara oleh pencetus konsep

resnet sendiri, (He, Zhang, Ren, dan Sun, 2016), *ResNet* dinyatakan sebagai arsitektur yang populer pada zamannya dan memiliki berbagai macam jenis arsitektur, mulai dari 18, 34, 50, 101, hingga 152 layer.



Gambar 2.8 Arsitektur *ResNet*-50

Keunggulan dari arsitektur *ResNet* yaitu karena adanya pengaruh lapisan pembobotan dari lapisan sebelumnya menjadikan *ResNet* memiliki optimalisasi lebih tinggi dibandingkan arsitektur lain yang tidak memiliki lapisan residual. Sebagai tambahan, *ResNet* memiliki kemampuan yang lebih akurat jika digunakan untuk mengklasifikasi citra medis (Mutawalli, Zaen, & Yuliadi, 2023). *ResNet50* sendiri merupakan model arsitektur jaringan dasar yang menggunakan CNN dengan 50 lapisan (Rikendry & Maharil, 2022).

2.6 Pencahayaan

Cahaya merupakan bentuk radiasi elektromagnetik pada sebagian dari spektrum elektromagnetik (Roth & Stahl, 2019). Cahaya memiliki panjang gelombang khusus sehingga membedakannya dari energi cahaya lain dalam spektrum elektromagnetiknya (Barnadi, 2021). Tiap panjang gelombang juga menciptakan kesan intensitas yang berbeda pada mata manusia (Putra et al., 2020).



Gambar 2.9 Lux meter

Salah satu satuan dalam menentukan intensitas cahaya adalah lux. Satuan lux merupakan satuan dari fluks luminus (lumen) yang mengenai permukaan yang disinari (SNI 6197, 2011). Dalam mengukur tingkat pencahayaan, diperlukan alat bernama lux meter. Gambar 2.9 merupakan foto dari lux meter.

Rumus 2 – 1 merupakan rumus untuk menghitung besarnya lux:

$$\text{total lux (lux)} = \frac{\text{total lumen (lumen)}}{\text{luas permukaan (m}^2\text{)}} \quad (2 - 1)$$

Berbeda dengan lux, lumen dapat diartikan sebagai ukuran besarnya cahaya yang dihasilkan oleh sebuah sumber cahaya (SNI 6197, 2011). Namun, dalam penelitian ini akan digunakan lux sebagai satuannya. Tabel 2.1 merupakan tabel yang menampilkan gambaran besarnya lux dengan gambaran terang cahaya dalam ruangan.

Tabel 2.1 Tabel perbandingan besarnya lux dan gambaran kecerahan ruangan

No.	Intensitas cahaya (lux)	Ruangan
1.	60	Ruang teras atau garasi
2.	150	Ruang tamu
3.	250	Ruang makan, ruang tidur, atau kamar mandi
4.	300	Ruang kerja

2.7 Penelitian Terdahulu

Penelitian terkait pendeteksian diabetes retinopati menggunakan CNN pada citra fundus bukanlah penelitian pertama. Sudah terdapat beberapa penelitian terkait dengan berbagai metode yang berbeda untuk menghasilkan pendeteksi diabetes retinopati menggunakan citra fundus. Pemilihan algoritma VGG sebagai model pendeteksi mendapat referensi dari jurnal milik Setiawan (2019) yang berjudul “Perbandingan Arsitektur *Convolutional Neural Network* untuk Klasifikasi Fundus”. Penelitian tersebut membandingkan akurasi dari beberapa algoritma dan metode optimasi CNN untuk melakukan klasifikasi citra fundus. Algoritma yang digunakan meliputi *AlexNet*, VGG-16, VGG-19, *GoogLeNet*, *Inception-V3*, *ResNet50*, *ResNet101*, *InceptionResNetV2*, dan *SqueezeNet*. Sementara metode optimasi yang digunakan meliputi tanpa optimasi, optimasi *gradient descent* (GDM), RmsProp, dan Adam. Berdasarkan penelitian tersebut, akurasi paling tinggi dihasilkan oleh model VGG-19 jika tanpa optimasi dengan

akurasi 89,3%, sensitivitas 90,4% dan spesifisitas 88,2%. Sementara jika dengan metode optimasi, model VGG-16 dengan metode optimasi GDM dengan hasil akurasi 92,3%, sensitivitas 90,7%, dan spesifisitas 94%. Berdasarkan penelitian ini, maka dipilihlah model dengan algoritma VGG sebagai acuan referensi berikutnya.

Salah satu penelitian yang berkaitan dengan penggunaan model klasifikasi VGG untuk mengklasifikasi penyakit pada retina mata dilakukan oleh William dan Lubis (2022) dengan judul “Klasifikasi Penyakit Mata Menggunakan CNN”. Terdapat lima klasifikasi mata pada penyakit ini yaitu mata normal, diabetes retinopati, glaukoma, katarak, dan retinopati hipertensi. Pada penelitian tersebut, digunakan *dataset* berupa citra fundus sejumlah 5740 citra yang telah melalui seleksi validasi. Citra tersebut kemudian diperbanyak dengan melakukan refleksi dan rotasi. Model yang digunakan secara spesifik adalah algoritma VGG-16. Berdasarkan penelitian tersebut, terdapat dua hasil yang penting sebagai landasan penelitian kali ini yaitu keberhasilan model untuk mengklasifikasi penyakit mata memiliki akurasi 91% dan loss 12,28%. Namun, dikatakan dalam penelitian tersebut, penyakit diabetes retinopati merupakan penyakit yang paling sulit diklasifikasi. Berdasarkan laporan penelitian, ada kemungkinan kesulitan tersebut akibat sedikitnya *dataset* pada kelas diabetes retinopati (112 citra sementara kelas dengan jumlah citra di atasnya adalah retinopati hipertensi dengan jumlah 340).

Penelitian selanjutnya membandingkan beberapa arsitektur untuk secara spesifik mendeteksi diabetes retinopati. Penelitian ini dilakukan oleh Afkariansyah (2022) dengan judul “Klasifikasi Fundus *Diabetic Retinopathy* Menggunakan *Deep Learning*”. Penelitian tersebut membandingkan lima arsitektur yaitu VGG-16, VGG-19, *ResNet101*, *ResNet50*, dan *Inception-V3* menggunakan *dataset* sejumlah 35.100 gambar yang terdiri dari mata normal, *mild*, *moderate*, *severe*, dan *proliferative diabetic retinopathy*. Berdasarkan penelitian tersebut, didapatkan hasil bahwa akurasi dari yang tertinggi adalah *Inception-V3* (77,99%), VGG-19 (75,32%), *ResNet50* (74,63%), VGG-16 (74,08%), dan *ResNet101* (73,67%). Sementara itu, berdasarkan total waktu komputasi dari yang tercepat sebagai berikut VGG-16 (13.291 detik), VGG-19 (13.300 detik), *ResNet50* (13.884 detik), *ResNet101* (14.987 detik), *Inception-V3* (22.870 detik). Dengan hasil tersebut, algoritma *Inception-V3* memiliki akurasi tertinggi diikuti VGG-19.

Penelitian selanjutnya telah menggunakan model VGG-16 dan spesifik untuk mendeteksi diabetes retinopati. Penelitian tersebut dilakukan oleh Wibowo & Alamsyah (2023) dengan judul “Klasifikasi *Diabetic Retinopathy* Menggunakan CNN dengan Arsitektur yang Dimodifikasi”. *Dataset* yang digunakan berisi 3664 gambar fundus retina dengan lima kelas dari diabetes retinopati. Hasil evaluasi menunjukkan klasifikasi diabetes retinopati menggunakan CNN berarsitektur VGG-16 menghasilkan akurasi 96,24%.

Keempat penelitian tersebut merupakan landasan yang masih berkaitan dengan proses pendeteksian diabetes retinopati menggunakan citra fundus. Namun, berdasarkan keempat penelitian tersebut, masih belum ada yang melakukan pendeteksian menggunakan pengambilan gambar langsung. Namun, untuk melakukan pendeteksian dengan gambar langsung, diperlukan sebuah aplikasi yang terintegrasi dengan kamera. Maka dari itu aplikasi yang paling memungkinkan adalah aplikasi *smartphone*.

Penelitian terkait pembuatan aplikasi android mengambil referensi dari penelitian karya Akbar, Sardjono, Cahyanti, dan Swedia (2020) dengan judul “Perancangan Aplikasi *Mobile* untuk Klasifikasi Sayuran Menggunakan *Deep Learning Convolutional Neural Network*”. Penelitian ini menguji penerapan arsitektur CNN pada lima perangkat seluler yang berbeda untuk melakukan klasifikasi sayuran. Berdasarkan penelitian tersebut didapatkan *test accuracy* sebesar 98.1%.

Penelitian yang digunakan sebagai referensi dari pembuatan prototipe detektor dari *smartphone* diambil dari beberapa jurnal. Referensi pertama diambil dari jurnal Bolster, Giardini, dan Bastawrous (2016), dengan judul “*The Diabetic Retinopathy Screening Workflow: Potential for Smartphone Imaging*”. Jurnal tersebut mengkompilasi berbagai perkembangan mekanisme untuk mendapat citra fundus untuk peneteksian diabetes retinopati. Salah satu referensi yang menjadi dasar pembuatan rancangan prototipe pada penelitian ini adalah penelitian Lord, Shah, San Filippo, dan Krishna (2010) yang berjudul “*Novel Uses of Smartphones in Ophthalmology*”. Penelitian tersebut membuat rancangan mirip oftalmoskop tidak langsung dengan memposisikan lensa 20D dekan dengan mata dan citra dari fundus diambil menggunakan *smartphone* yang ditambahi senter pena.

Referensi kedua diambil dari penelitian Haddock, Kim, dan Mukai (2013) dengan judul “*Simple, Inexpensive Technique for High-Quality Smartphone Fundus Photography in Human and Animal Eyes*”. Pada penelitian tersebut, digunakan kamera iPhone 4 atau iPhone 5, lensa 20D, dengan atau tanpa tambahan *koeppe lens*, serta menggunakan cahaya dari *smartphone* sehingga dapat berfungsi seperti oftalmoskop tidak langsung. Pada penelitian tersebut juga digunakan aplikasi Filmic Pro yang berfungsi dalam mengontrol secara independen fokus, eksposur, dan intensitas cahaya saat merekam video sehingga bisa didapatkan gambar berkualitas. Dikatakan bahwa tanpa aplikasi tersebut, akan sulit didapatkan citra yang optimal. Pengambilan citra fundus berhasil dengan baik dengan lensa 20D dan tanpa menggunakan *koeppe lens*, serta muncul pernyataan bahwa lensa 28D dan 30D lebih berguna dalam kasus tertentu (pupil yang lebih kecil).

Referensi ketiga diambil dari penelitian Ademola dan Olatunji (2017) dengan judul “*Retinal Imaging with Smartphone*”. Pada penelitian tersebut, digunakan Blackberry Z-10 sebagai *smartphone* dan lensa nonkontak 20D oleh Volk. Selain itu, untuk memperlebar pupil digunakan phenylephrine (2,5% untuk anak-anak dan 10% untuk dewasa) dan tetes mata tropicamide 1%. Jarak optimal didapatkan saat jarak mata dengan lensa 9 cm, dan jarak lensa ke *smartphone* 14,5 cm. Pada penelitian tersebut didapatkan citra retina yang jelas tanpa adanya opasitas media and tanpa perangkat lunak tambahan.

Referensi keempat diambil dari penelitian Iqbal (2021) dengan judul “*Smartphone Fundus Photography: A Narrative Review*”. Pada penelitian tersebut digunakan *smartphone* dan lensa nonkontak 20D oleh Volk dan menggunakan sumber cahaya dari senter. Pada jurnal ini dipaparkan juga secara lebih jelas desain dan komparasi dari beberapa metode yang digunakan. Dengan keempat referensi tersebut maka sangat memungkinkan untuk mengadaptasi konsep yang ada ke dalam penelitian ini.

BAB III

Metode Penelitian

3.1 Alur Penelitian

Proyek Tugas Akhir ini bertujuan untuk membuat aplikasi android untuk melakukan pendeteksian penyakit diabetes retinopati dengan masukan foto secara langsung melalui *smartphone* yang dilengkapi lensa fundus. Proses pengerjaan proyek Tugas Akhir ini dibagi menjadi beberapa tahap yang digambarkan pada diagram pada Gambar 3.1.



Gambar 3.1 Alur penelitian

Gambar 3.1 menunjukkan bahwa proses penelitian dimulai dengan tahapan analisis masalah. Pada tahap analisis masalah ini, langkah pertama yang dilakukan adalah mengidentifikasi dan menetapkan batasan masalah yang akan menjadi dasar dalam pelaksanaan proyek Tugas Akhir. Setelah tahap identifikasi masalah selesai, proses dilanjutkan dengan melakukan studi literatur untuk mendalami literatur yang

relevan dengan proyek Tugas Akhir. Langkah berikutnya adalah merancang aplikasi yang akan dikembangkan. Seluruh rancangan yang sudah disusun tersebut selanjutnya diimplementasikan dan diuji keberhasilannya. Selanjutnya, proses dilanjutkan dengan konfigurasi dan implementasi aplikasi bersama dengan perangkat keras. Rancangan gabungan aplikasi dan perangkat keras yang sudah selesai dibuat selanjutnya akan dibawa ke tahap pengujian sederhana lalu dilanjutkan dengan pengujian pada subjek penelitian sebagai langkah terakhir dari penelitian untuk mengukur keberhasilan hasil dari Tugas Akhir.

3.2 Analisis Masalah

Topik yang dijadikan sebagai acuan utama Tugas Akhir ini adalah risiko penyakit diabetes retinopati yang sangat berbahaya sehingga penting dilakukan pengecekan secara rutin. Namun, hal tersebut masih terbatas akibat biaya pengecekan yang mahal, kurang fleksibel, dan waktu tunggu hasil diagnosis yang tidak sebentar. Maka dari itu, perlu dilakukan sebuah upaya untuk menangani keterbatasan proses diagnosis konvensional. Salah satu upaya untuk menanggulangi kondisi tersebut dapat dilakukan dengan menghadirkan sebuah aplikasi sederhana yaitu aplikasi *smartphone* yang dapat dijangkau semua kalangan untuk mendeteksi secara cepat dan memiliki akurasi yang cukup baik. Selain membuat aplikasi *smartphone*, perlu diperhatikan bahwa proses pengambilan citra fundus juga memerlukan alat yang mahal sehingga baik jika dilakukan upaya untuk menyederhanakan perangkat yang mahal menjadi sebuah perangkat yang bisa diakselerasikan dengan *smartphone*.

3.3 Studi Literatur

Studi literatur bertujuan untuk memahami dan memperdalam pengetahuan terkait materi dari metode dan berbagai aspek berkaitan dengan proyek yang dikerjakan serta bagaimana cara mengimplementasikannya. Melalui studi literatur, peneliti juga akan mendapat referensi metode-metode sebelumnya yang sudah pernah digunakan untuk kemudian dilakukan pengembangan, penelitian metode yang berbeda, atau pengujian variabel yang belum dilakukan untuk melengkapi data yang ada. Studi literatur ini sendiri akan dilakukan dengan mencari dan memahami teori-teori dari jurnal artikel maupun artikel dari *website* yang dapat dipercaya.

3.4 Perancangan Model CNN

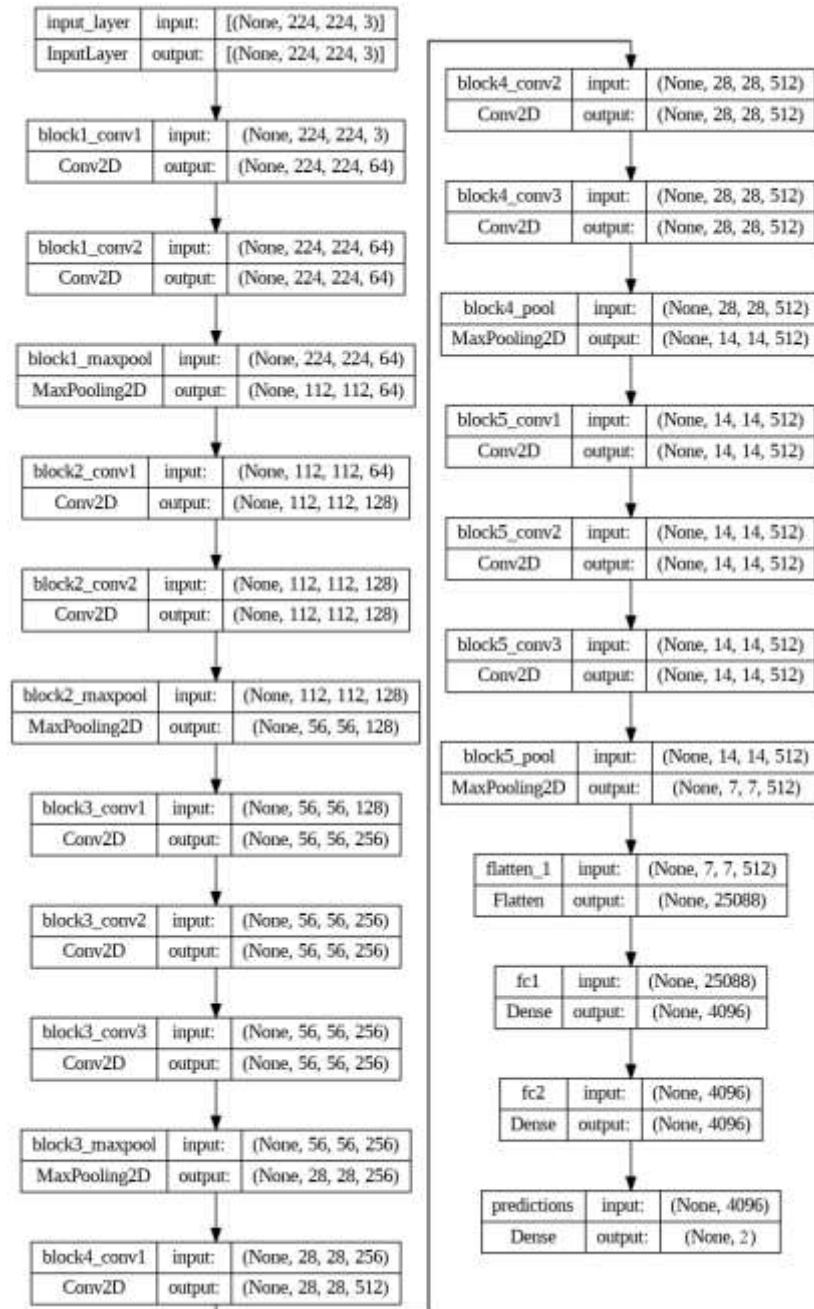
Sistem untuk mendeteksi retina penderita diabetes retinopati menggunakan arsitektur *Convolution Neural Network* (CNN). Arsitektur yang dipilih menggunakan algoritma VGG-19 dan *ResNet-50*. Arsitektur CNN yang dibuat dan akan diimplementasikan, harus diuji terlebih dahulu sehingga arsitektur yang dipilih memang dapat melakukan pendeteksian dengan baik. Pengujian ini nanti akan dilakukan dengan didasarkan pada dua hasil. Pertama yaitu berdasarkan *convolution matrix*, dan yang kedua berdasarkan hasil implementasi pendeteksian menggunakan gambar acak dari *dataset* yang ada.

Model harus mampu melakukan prediksi dengan akurasi di atas 90%. Hal ini didasarkan pada penelitian oleh Tantono (2024) yang berhasil mendapat akurasi rata-rata 92% untuk masing-masing pengujian dengan keempat kelas (*mild, moerate, severe, proliferative*) dengan arsitektur *VGG-16* maupun arsitektur *ResNet-50*. Namun, dalam penelitian yang sama, belum berhasil melakukan pendeteksian dengan benar pada citra uji. Selain itu, model diharapkan mampu memprediksi dengan benar setidaknya 80% citra dalam penerapan langsung.

3.4.1 Perancangan Model VGG-19

Filter pada VGG-19 menggunakan *convolution layer* dan *pooling layer* yang keduanya berukuran *kernel* 3x3 dan *stride* 1. VGG-19 dimulai dengan 2 *convolution layer* diikuti dengan 1 *pooling layer* dan menghasilkan *feature map* berdimensi 112 x 112 x 64. Dilanjutkan dengan 2 *convolution layer* dan 1 *pooling layer* dengan masukan *feature map* berukuran 128 dan menghasilkan *feature map* baru berdimensi 56 x 56 x 128. Dilanjutkan dengan 4 *convolution layer* dan 1 *pooling layer* dengan masukan *feature map* berukuran 256 dan menghasilkan *feature map* baru berdimensi 28 x 28 x 256. Dilanjutkan dengan 4 *convolution layer* dan 1 *pooling layer* dengan masukan *feature map* berukuran 512 dan menghasilkan *feature map* baru berdimensi 14 x 14 x 512. Dilanjutkan dengan 4 *convolution layer* dan 1 *pooling layer* dengan masukan *feature map* berukuran 512 dan menghasilkan *feature map* baru berdimensi 7 x 7 x 512. Kemudian diakhiri dengan *global average pooling* dan *softmax* dan menghasilkan klasifikasi citra. Rancangan arsitektur VGG-19 ini akan diterapkan sebagai model untuk melakukan pendeteksian

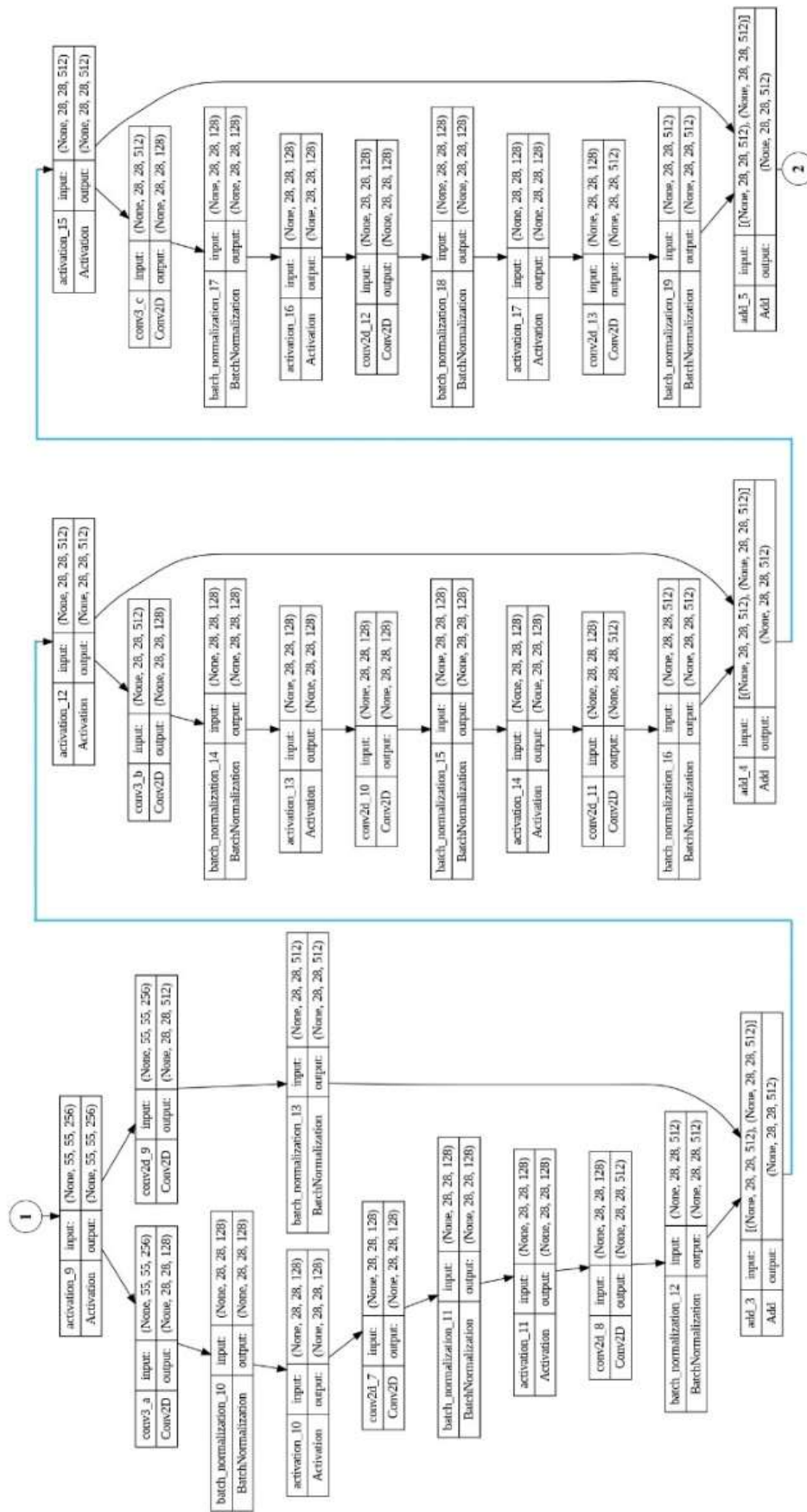
terhadap *dataset*. Ilustrasi *layer* dari arsitektur VGG-19 dapat dilihat pada Gambar 3.2.



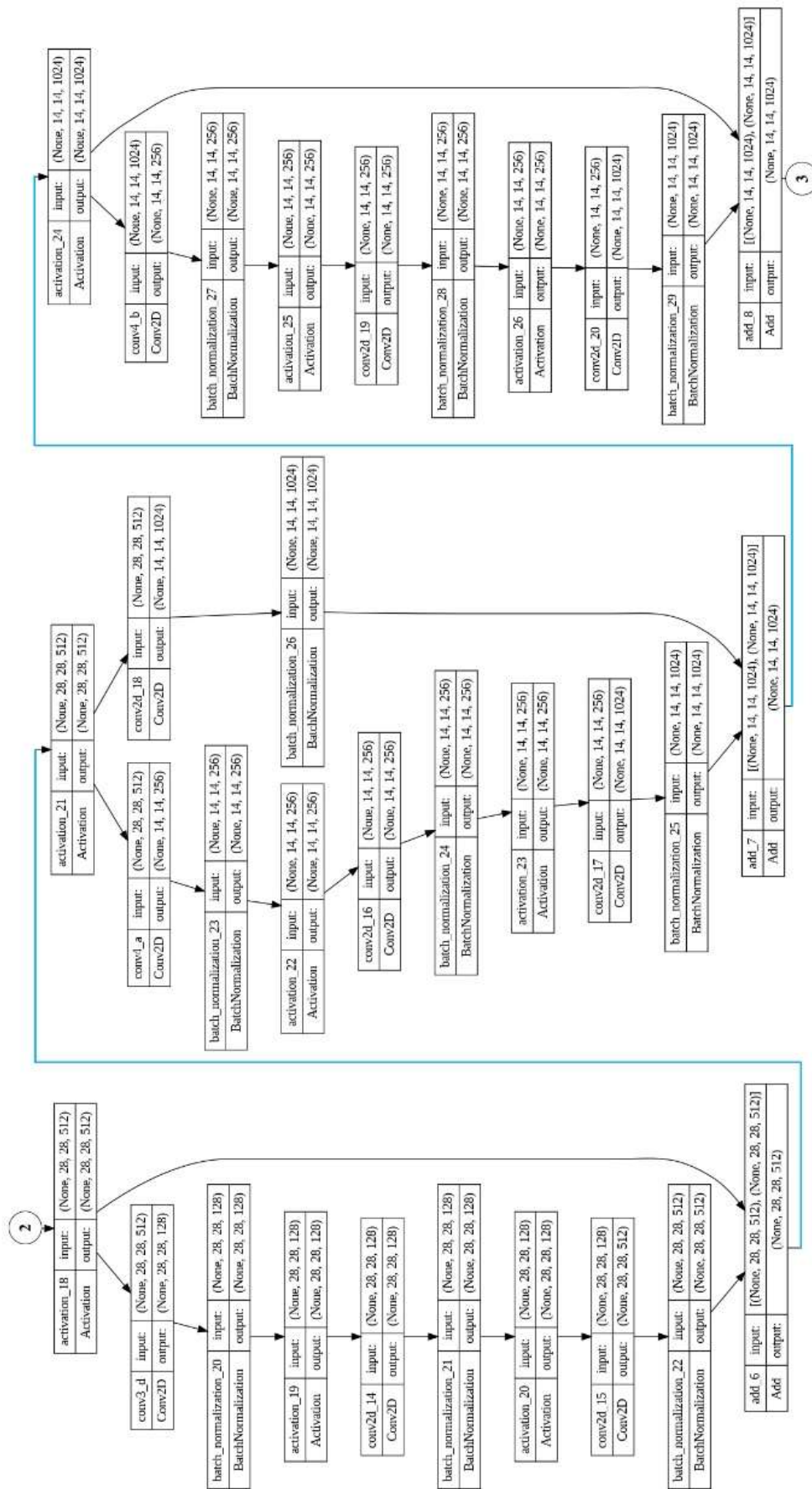
Gambar 3.2 *Layer* dalam arsitektur VGG-19

3.4.2 Perancangan Model *ResNet-50*

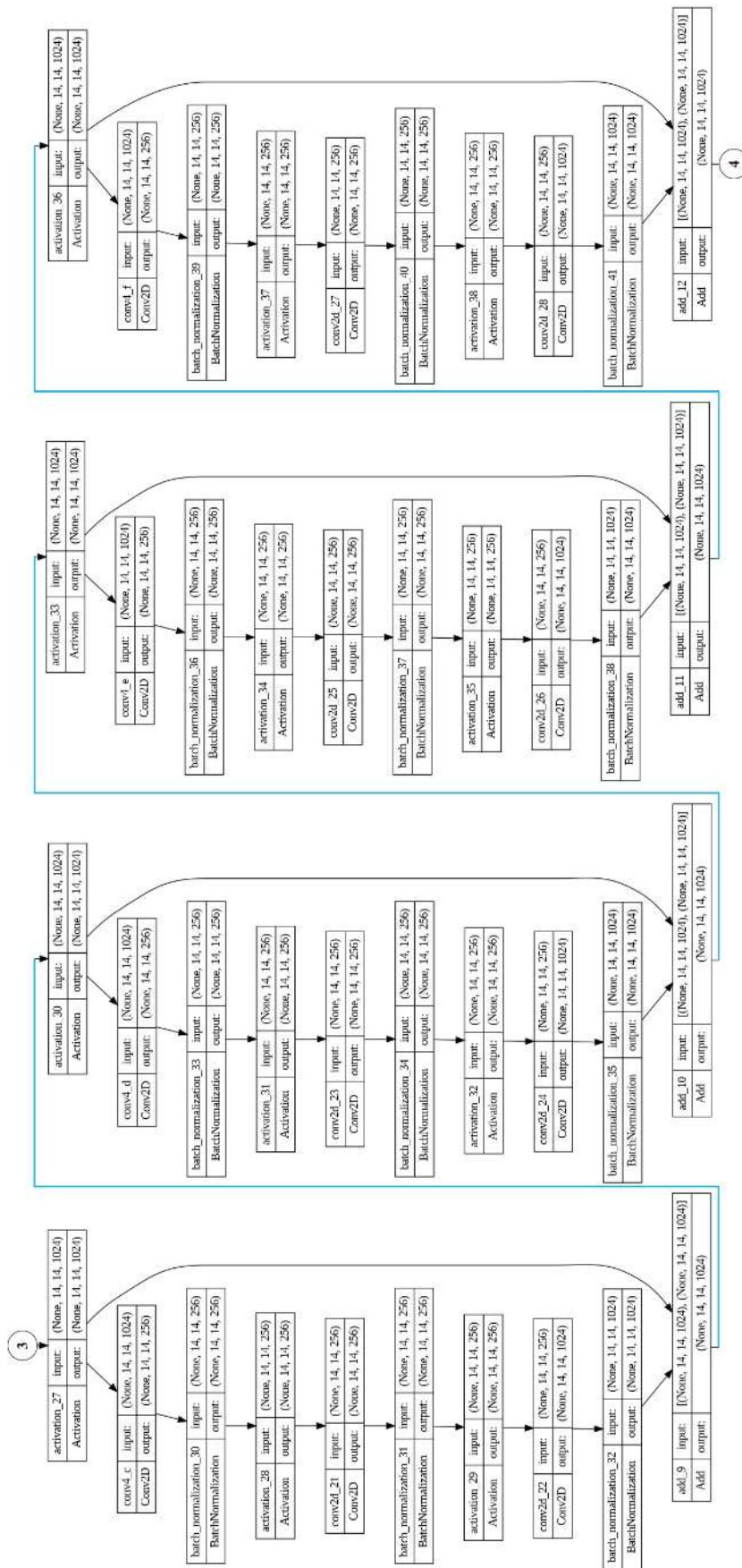
Rancangan untuk *ResNet-50* akan memanfaatkan proses melalui lima blok *convolution*. Blok pertama akan terdapat satu *convolution 2D layer*, satu *layer* normalisasi, satu lapisan aktivasi ReLu, dan satu *maxpooling layer*.



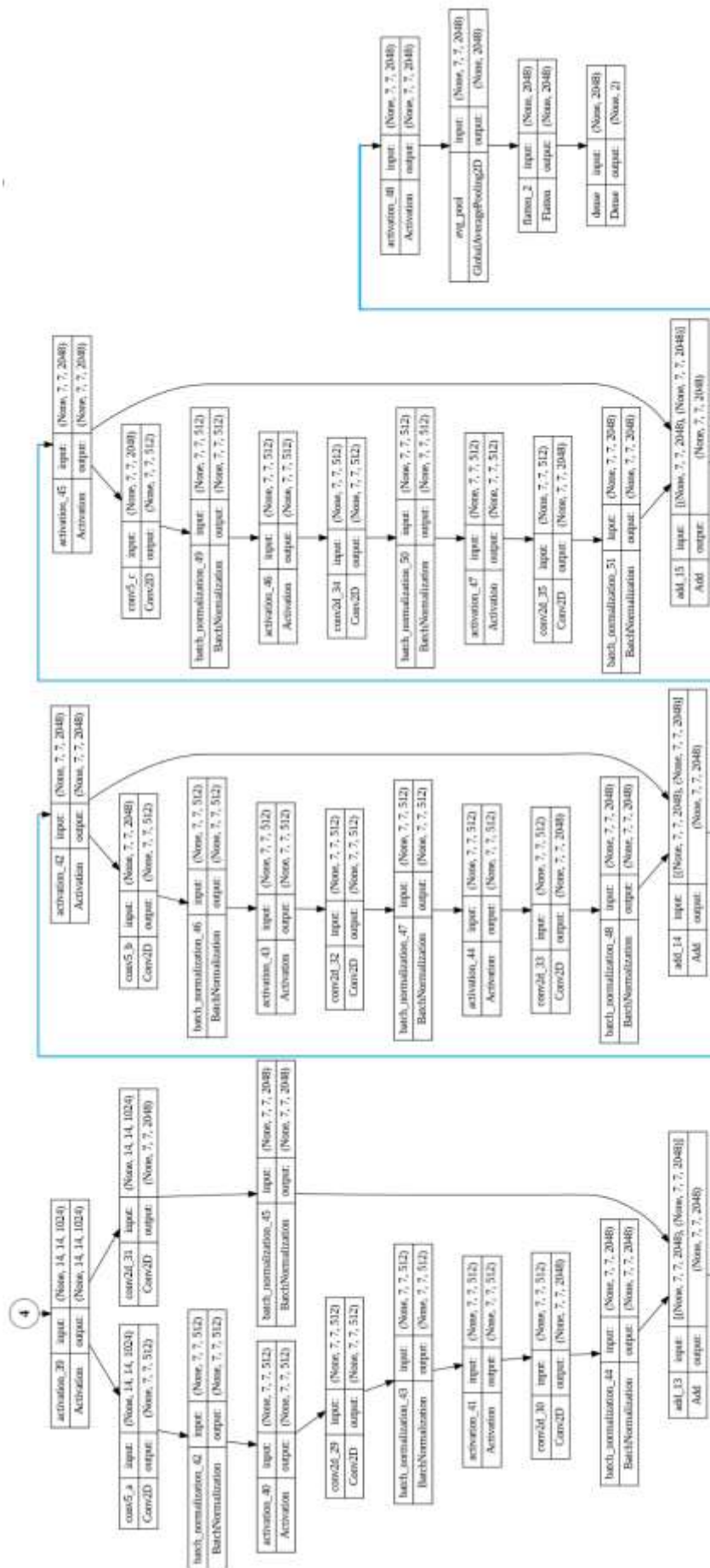
Gambar 3.4 Arsitektur *ResNet-50* bagian 2



Gambar 3.5 Arsitektur ResNet-50 bagian 3



Gambar 3.6 Arsitektur ResNet-50 bagian 4



Gambar 3.7 Arsitektur ResNet-50 bagian 5

Pada model tersebut, citra RGB berukuran 224x224 *pixel* akan diproses melalui lima blok *convolution*. Pada blok pertama, data akan melalui satu *convolution 2D layer*, satu *layer* normalisasi, satu lapisan aktivasi ReLu, dan satu *maxpooling layer*. Pada blok kedua hingga kelima, data akan diproses dengan beberapa blok yang lebih kecil yang disebut *residual block*. Dalam satu *residual block* terdapat tiga hingga empat *convolution 2D layer*, tiga hingga empat *layer* normalisasi, dan tiga lapisan aktivasi ReLu. Selanjutnya, data akan melalui satu *average pooling layer*, satu *flatten layer*, dan satu *dense layer*.

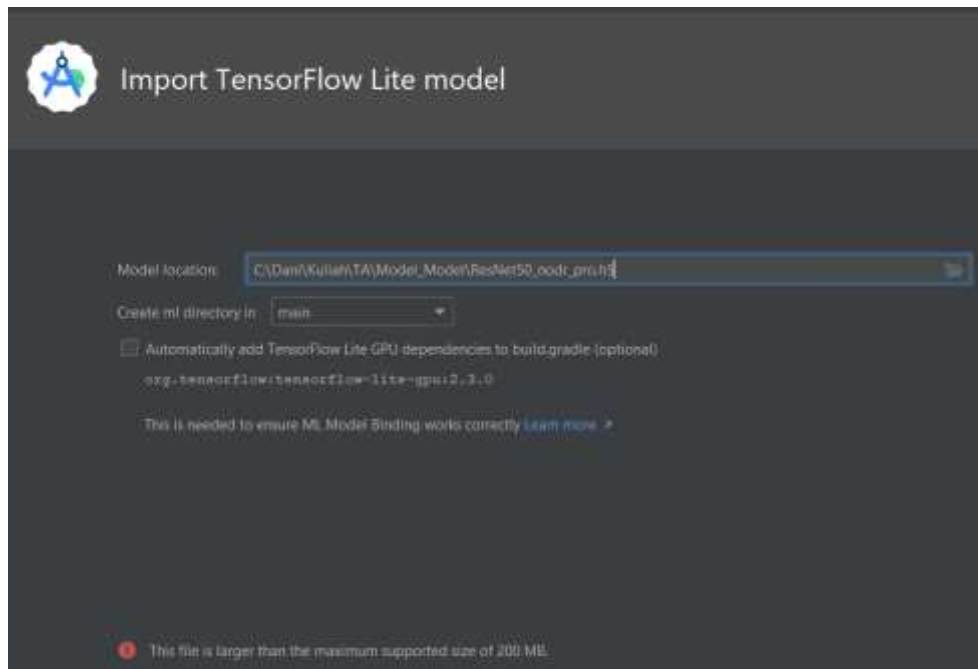
Blok *convolution* kedua memiliki total tiga *residual block*. Blok *convolution* ketiga memiliki total empat *residual block*. Blok *convolution* keempat memiliki total enam *residual block*. Blok *convolution* kelima memiliki total tiga *residual block*. Total terdapat 52 *convolution 2D layer*, 52 *batch normalization layer*, 48 *activation layer*, 1 *max pooling layer*, 1 *average pooling layer*, 1 *flatten layer*, dan 1 *dense layer*. Untuk dimensi dan susunan dari layer dapat dilihat pada masing-masing gambar yang dimaksud.

3.4.3 Perancangan Penyederhanaan Model

Terdapat ketentuan bahwa kondisi model yang dapat ditanamkan pada aplikasi android pada Android Studio dibatasi 200 Mb. Hal ini dibuktikan dengan keadaan di mana penulis telah memasukkan model berukuran lebih dari 200 Mb dan muncul peringatan seperti pada Gambar 3.8. Maka ada kemungkinan jika arsitektur yang dibuat akan berukuran lebih besar dari pada batas ukuran Android Studio. Maka, perlu dilakukan pemangkasan beberapa layer seperti *convolution layer* dan *dense layer*. Pada tahap awal, akan dipangkas terlebih dahulu beberapa *convolution layer*. Namun, jika ukuran file masih terlalu besar, maka akan dicoba dengan melakukan pemotongan *dense layer*.

Pemotongan *dense layer* akan lebih berdampak pada besar ukuran *file* model yang nanti disimpan. Hal ini karena beberapa alasan. Pertama, *dense layer* memiliki jumlah parameter yang lebih besar dibandingkan *convolution layer*. Setiap neuron dalam *dense layer* terhubung ke semua neuron di *layer* sebelum atau sesudahnya. Jumlah parameter yang disimpan *dense layer*

adalah hasil kali jumlah neuron layer sebelumnya dikali jumlah neuron *dense layer* tersebut. Sementara untuk *convolution layer* memiliki jumlah parameter yang lebih sedikit, bergantung pada ukuran *kernel*, jumlah *channel* dan filter. Kedua, pemrosesan data pada *dense layer* memerlukan representasi data yang rata (*flattened*) dari fitur yang diekstraksi oleh *convolution layer* sebelumnya. Hal ini menjadikan data vektor berukuran besar untuk disimpan. Penghapusan *dense layer* tidak hanya mengurangi ukuran *file* model, tetapi juga mempercepat proses komputasi karena mengurangi kompleksitas kalkulasi yang terlibat dalam lapisan ini. Teori tersebut didasarkan pada dua sumber buku yaitu buku karya Goodfellow, Bengio, dan Courville (2016) dan buku karya Sze, Chen, Yang, dan Emer (2020).



Gambar 3.8 Peringatan batas ukuran maksimal model pada Android Studio

3.5 Dataset

CNN menerapkan metode *supervised learning* yang memerlukan data dengan label untuk membentuk model (Holiyanti, Sukmawati, Fahmi, & Rozikin, 2022). Maka dari itu diperlukan data latih dan data uji untuk membuat model CNN untuk deteksi diabetes retinopati. Data yang digunakan merupakan citra fundus retina dari dua sumber. Dalam penelitian ini terdapat data yang dibiarkan secara original tanpa dilakukan perlakuan, ada yang diambil kelas tertentu saja, dan ada yang

mendapat perlakuan khusus. Data 1, data 2, dan data 3 diambil dari satu sumber yang sama yaitu <https://www.kaggle.com/competitions/diabetic-retinopathy-detection>. Tetapi, data 4 berasal dari sumber yang berbeda. Untuk lebih lengkapnya terkait perbedaan keempat jenis data tersebut, dapat dilihat pada Tabel 3.1.

Tabel 3.1 Tabel pelabelan jenis data dan perbedaannya

Label data	Sumber dan perlakuan	Jumlah citra	Jumlah kelas
Data 1	https://www.kaggle.com/competitions/diabetic-retinopathy-detection . Diakses pada tanggal 16 November 2023. Data hanya dibagi menjadi 60% data <i>train</i> , 20% data <i>test</i> , dan 20% data validasi.	<i>No DR</i> = 25.810 <i>Mild</i> = 2.443 <i>Moderate</i> = 5.292 <i>Proliferative</i> = 708 <i>Severe</i> = 873 Total = 35.126	5
Data 2	https://www.kaggle.com/competitions/diabetic-retinopathy-detection . Diakses pada tanggal 16 November 2023. Data dibagi menjadi 60% data <i>train</i> , 20% data <i>test</i> , dan 20% data validasi, dan diambil 2 kelas.	<i>No DR</i> = 25.810 <i>Proliferative</i> = 708 Total = 26.518	2
Data 3	https://www.kaggle.com/competitions/diabetic-retinopathy-detection . Diakses pada tanggal 16 November 2023. Serupa dengan data 2. Namun, mendapat perlakuan tambahan untuk kedua kelas.	<i>No DR</i> = 11.930 <i>Proliferative</i> = 8.607 Total = 20.537	2
Data 4	https://www.kaggle.com/datasets/pkdarabi/diagnosis-of-diabetic-retinopathy . Diakses pada tanggal 19 Januari 2024. Data dibagi menjadi 80% data <i>train</i> , 10% data <i>test</i> , dan 10% data validasi.	<i>No DR</i> = 1.430 <i>DR</i> = 1.408 Total = 2.838	2

3.5.1 Data 1

Data 1 bersumber dari <https://www.kaggle.com/competitions/diabetic-retinopathy-detection>. Data tersebut kemudian dibagi menjadi data *train*, *test*, dan *val* dengan rincian seperti yang tampak pada Tabel 3.2.

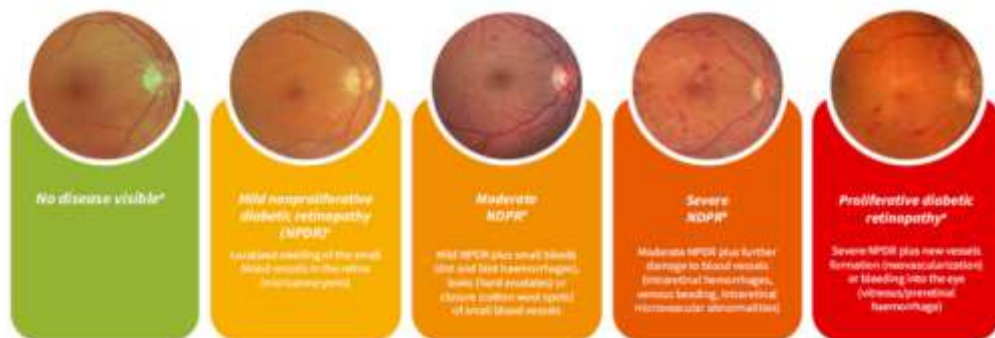
Tabel 3.2 Tabel *Dataset 1*

<i>Train</i>	<i>Test</i>	<i>Validate</i>
<i>No DR</i> = 15.431	<i>No DR</i> = 5.169	<i>No DR</i> = 5.210
<i>Mild</i> = 1.482	<i>Mild</i> = 484	<i>Mild</i> = 477
<i>Moderate</i> = 3.203	<i>Moderate</i> = 1.062	<i>Moderate</i> = 1.027
<i>Proliferative</i> = 428	<i>Proliferative</i> = 145	<i>Proliferative</i> = 135
<i>Severe</i> = 531	<i>Severe</i> = 166	<i>Severe</i> = 176

Dataset tersebut kemudian dijadikan dua kelas yaitu *No DR* dan *DR*. *DR* sendiri merupakan gabungan dari empat kelas *DR* yaitu *Mild*, *Moderate*, *Proliferative*, dan *Severe*.

3.5.2 Data 2

Data 2 merupakan Data 1 (poin 3.5.1), namun diambil dua kelas yaitu *No DR* dan *Proliferative DR*. Pemilihan kelas didasarkan pada citra yang menunjukkan bahwa *Proliferative DR* merupakan kelas yang perbedaan dengan *No DR* paling terlihat di antara kelas yang lain. Gambar 3.9 menampilkan perbedaan retina *No DR* dan keempat kelas lainnya.

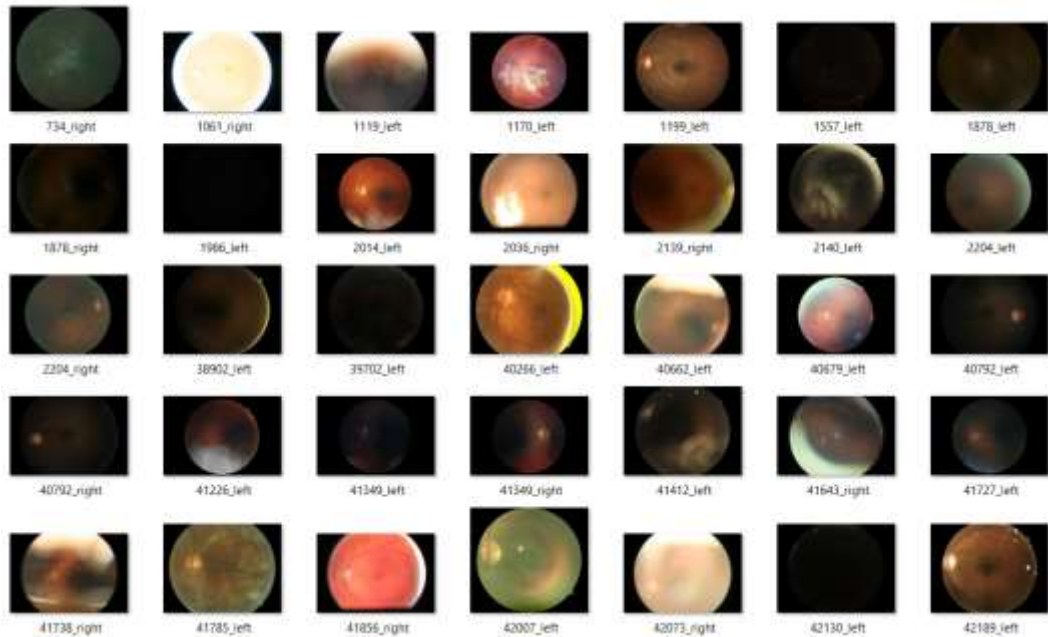


Gambar 3.9 Perbandingan retina *No DR* dengan keempat kelas lainnya

Sumber: Vitriana, 2023

3.5.3 Data 3

Data 3 merupakan Data 2 (poin 3.5.3), namun telah dilakukan penyeleksian mau pun penggandaan data dengan harapan dapat mempersempit selisih jumlah dari kedua kelas. Seleksi dilakukan untuk kelas *No DR* secara manual oleh penulis berdasarkan *noise* dan kejelasan gambar secara kasat mata. Ada pun contoh citra yang dianggap perlu dihilangkan terdapat pada Gambar 3.10.

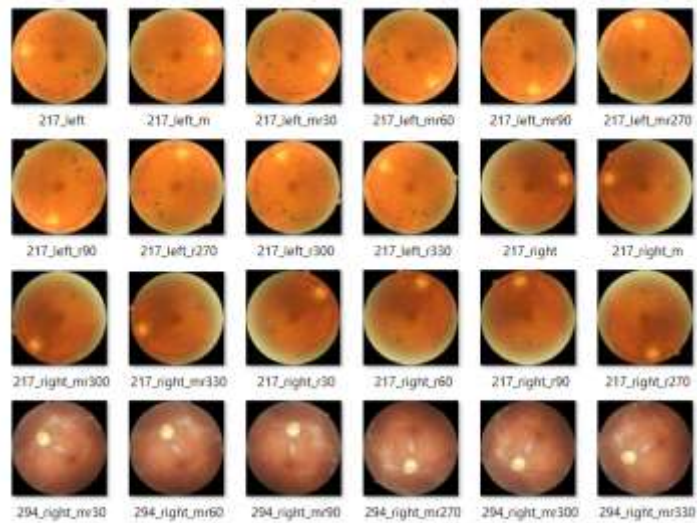


Gambar 3.10 Contoh citra yang dianggap rusak

Data *No_DR* juga kemudian dilakukan *crop image* dengan potongan kode yang dapat dilihat pada Lampiran 1. Sementara itu, *Proliferative DR* digandakan dengan melakukan perlakuan sebagai berikut:

- | | |
|---------------------------|---|
| a. Data original di-crop; | b. Data a direfleksikan terhadap sumbu y; |
| c. Data a dirotasi 30°; | d. Data c direfleksikan terhadap sumbu y; |
| e. Data a dirotasi 60°; | f. Data e direfleksikan terhadap sumbu y; |
| g. Data a dirotasi 90°; | h. Data g direfleksikan terhadap sumbu y; |
| i. Data a dirotasi 330°; | j. Data i direfleksikan terhadap sumbu y; |
| k. Data a dirotasi 300°; | l. Data k direfleksikan terhadap sumbu y; |
| m. Data a dirotasi 270°; | n. Data m direfleksikan terhadap sumbu y; |

Maka dari itu, dari satu data akan menjadi 14 citra. Potongan kode dapat dilihat pada Lampiran 2 dan contoh data dapat dilihat pada Gambar 3.11.



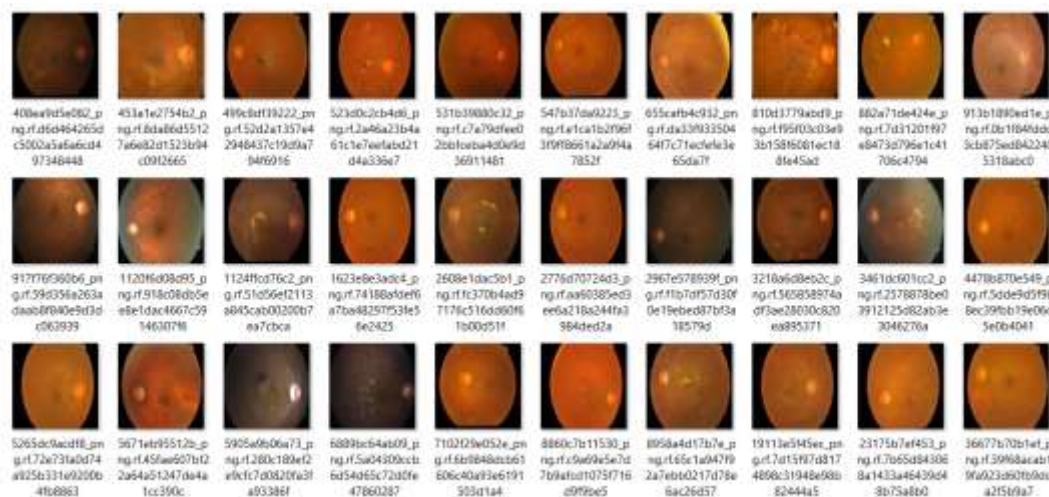
Gambar 3.11 Contoh citra *Proliferative DR* yang digandakan

Jumlah akhir setelah mengalami perlakuan terhadap data, dapat dilihat pada Tabel 3.3.

Tabel 3.3 Tabel *Dataset 1* hasil modifikasi pada kelas *No DR* dan *Proliferative*

<i>Train</i>	<i>Test</i>	<i>Validate</i>
<i>No DR</i> = 7.122 <i>Proliferative</i> = 5.824	<i>No DR</i> = 2.433 <i>Proliferative</i> = 837	<i>No DR</i> = 2.375 <i>Proliferative</i> = 1.946

3.5.4 Data 4

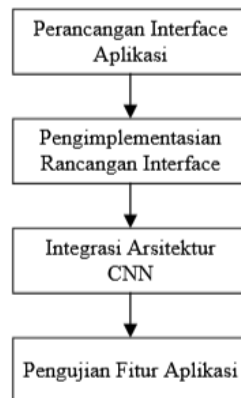


Gambar 3.12 Contoh citra data 4

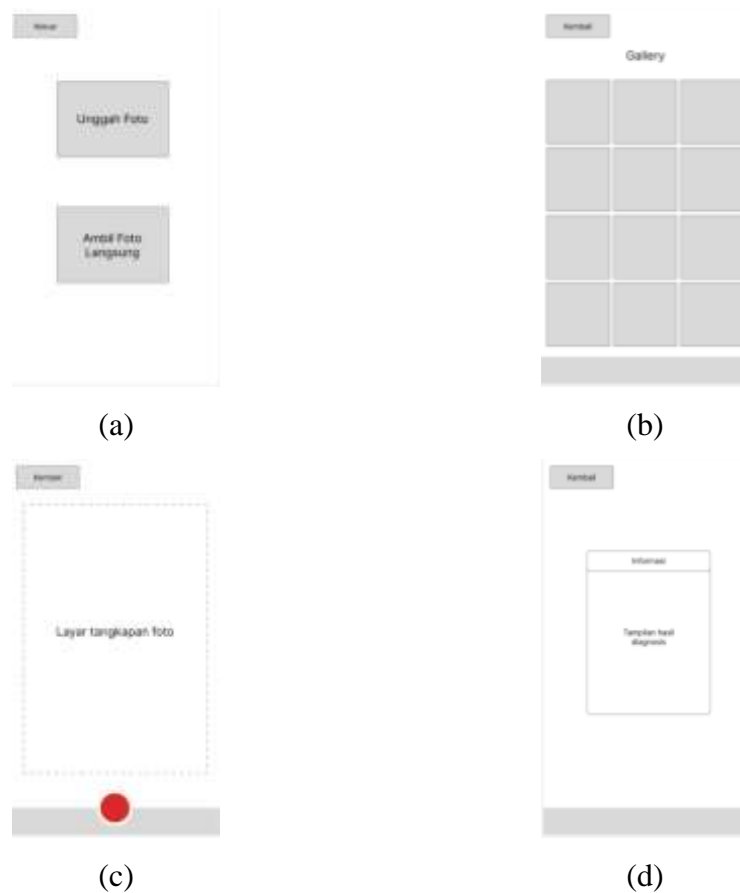
Data 4 bersumber dari <https://www.kaggle.com/datasets/pkdarabi/diagnosis-of-diabetic-retinopathy>. Data 4 merupakan data yang serupa

dengan data 1 namun telah dilakukan seleksi oleh pihak penyedia *dataset* dan hanya terdiri dari dua kelas yaitu *No DR* dan *DR*. Data ini nantinya akan langsung digunakan tanpa melalui proses pengolahan data. Contoh untuk citra data 4 dapat dilihat pada Gambar 3.12.

3.6 Perancangan Aplikasi



Gambar 3.13 Alur pembuatan aplikasi android



Gambar 3.14 Rancangan *interface* a) Menu utama b) Pilihan Unggah Foto
c) Pilihan Ambil Foto Langsung d) Informasi hasil diagnosis

Aplikasi akan terfokus pada kemampuan untuk deteksi apakah seseorang memiliki potensi atau ciri sebagai penderita diabetes retinopati atau tidak. *Dataset* serta arsitektur VGG-19 akan ditanamkan pada aplikasi sehingga aplikasi tidak terikat dengan *database* dalam jaringan. Gambar 3.13 merupakan alur perancangan aplikasi.

Proses perancangan aplikasi diawali dengan pembuatan rancangan *interface* aplikasi. Rancangan *interface* kemudian diimplementasikan menggunakan Android Studio. Gambar 3.14 merupakan rancangan *interface* dari aplikasi yang akan dibuat.

Proses selanjutnya adalah pengintegrasian model arsitektur CNN yang sudah dibuat ke dalam aplikasi android. Selanjutnya adalah proses uji coba apakah fitur-fitur dalam aplikasi bisa berjalan baik atau tidak.

3.7 Integrasi Perangkat Lunak dan Perangkat Keras

Perangkat keras yang digunakan terdiri dari *smartphone* dan kamera funduskopi. Tabel 3.4 merupakan spesifikasi dari *smartphone* yang akan digunakan dalam membuat aplikasi.

Kamera fundus yang digunakan merupakan adaptasi dari mekanisme oftalmoskop tidak langsung. Perangkat tersebut terdiri atas kamera *smartphone* yang ditambah lensa 20D dari Volk, serta dihubungkan dengan oftalmoskop funduskopi adaptor *smartphone* yang dapat dilihat pada Gambar 3.15. Tabel 3.5 merupakan spesifikasi dari lensa dan oftalmoskop funduskopi adaptor *smartphone* yang akan digunakan.

Gambar 3.16 merupakan rancangan dari integrasi *smartphone* dan kamera fundus. Ada pun cara penggunaan dari oftalmoskop funduskopi adaptor *smartphone* sebagai berikut:

1. Tutup lensa diputar, lensa 20D dimasukkan, lalu tutup lensa dikencangkan kembali;
2. Kamera *smartphone* dan *blitz* diposisikan pada pusat lingkaran penjepit;
3. Knop penjepit diputar untuk fiksasi *smartphone*;
4. Panjang lengan dapat diatur dan umumnya dimulai dari posisi paling panjang. Mur diputar untuk fiksasi lengan adaptor;
5. Mode video *smartphone* dinyalakan dengan kondisi *blitz* menyala;

6. Jarak lensa (dan adaptor) diatur menggunakan tangan non-dominan dengan jarak kira-kira 3 cm;
7. Cahaya kamera *smartphone* diarahkan ke pusat pupil hingga tampak bayangan *red reflex*;
8. Lensa (dan adaptor) dimundurkan secara perlahan hingga didapatkan gambaran retina yang diinginkan.

Tabel 3.4 Tabel spesifikasi *smartphone*

No.	Spesifikasi	Keterangan
Layar Utama		
1.	Jenis	AMOLED
2.	Ukuran	6.55 inci
3.	Resolusi	1080 x 2400 piksel
4.	Rasio	20:9
5.	Kerapatan	402 ppi
Processor		
1.	<i>Chipset</i>	Qualcomm Snapdragon 778G
2.	CPU	Octa-core (4x2.4 GHz Kryo 670 & 4x1.8 GHz Kryo 670)
3.	GPU	Adreno 642L
Memori		
1.	RAM	8 GB
2.	Memori Internal	256 GB
Kamera Utama		
1.	Jumlah Kamera	3
2.	Konfigurasi	108 MP (<i>wide</i>), f/1.9 8 MP (<i>ultrawide</i>), f/2.2 2 MP (<i>macro</i>), f/2.4
3.	Fitur	Dual-LED dual-tone flash, HDR, panorama, Video: 4K@30fps, 1080p@30/60/120fps; gyro-EIS
4.	Perbesaran	Foto Kamera Utama: 0.6x hingga 10x Video Kamera Utama: 0.6x hingga 6x

Tabel 3.5 Tabel spesifikasi oftalmoskop funduskopi adaptor *smartphone*

No.	Spesifikasi
1.	Cocok untuk <i>smartphone</i> apa pun dengan jarak maksimal lingkaran luar kamera utama dan <i>blitz</i> adalah 2,5 cm
2.	Ketebalan maksimal <i>smartphone</i> + <i>casing</i> : 1,4 cm
3.	Berat 143 gram
4.	Panjang 21-28 cm
5.	Lensa 20D merek Volk



Gambar 3.15 Oftalmoskop funduskopi adaptor *smartphone*

Sumber: <https://www.tokopedia.com/veronikamicrosurgery>



(a)

(b)

Gambar 3.16 Oftalmoskop funduskopi adaptor yang diintegrasikan dengan *smartphone* a) Tampak dari depan layar b) Tampak dari belakang layar

Sumber: <https://www.tokopedia.com/veronikamicrosurgery>

3.8 Pengujian

Pengujian yang akan dilakukan nantinya dibagi menjadi dua tahap, yang pertama adalah pengujian terkait keberhasilan fitur pada aplikasi dan yang kedua adalah pengujian terkait variabel-variabel penentu terhadap hasil pendeteksian diabetes retinopati.

3.8.1 Pengujian Fitur Deteksian dari Citra Galeri ke Aplikasi

Pengujian menggunakan citra yang diunggah ke aplikasi akan dilakukan dengan mengambil gambar diabetes retinopati dari *dataset* secara acak. Gambar tersebut telah diberi label sehingga dapat dipantau apakah pendeteksian sudah akurat atau belum. Pengujian akan dilakukan menggunakan 200 citra pada tiap klasifikasi. Akurasi dari pendeteksian tersebut dihitung menggunakan rumus 3-1. Rumus tersebut akan membagi data yang dideteksi dengan benar dengan total data yang dimasukkan dikali seratus persen. Perhitungan ini tidak akan digabung melainkan akan dilakukan pada setiap pengujian.

$$\text{Keakuratan(\%)} = \frac{\text{jumlah citra terdeteksi benar}}{\text{jumlah citra masukan}} \times 100\% \quad (3 - 1)$$

3.8.2 Pengujian Fitur Deteksi dari Citra Foto Cetak

Sebelum dilakukan pengujian secara langsung kepada subjek penelitian, pengujian akan dilakukan menggunakan kamera *smartphone* untuk mendeteksi citra penderita diabetes retinopati yang sudah ada (tidak secara langsung ke subjek). Citra yang sudah diseiakan kemudian dicetak pada kertas lalu difoto kembali menggunakan aplikasi yang sudah dibuat. Hal ini dimaksudkan untuk menguji apakah aplikasi dan pendeteksian secara langsung sudah bisa berjalan sesuai harapan atau belum.

3.8.3 Pengujian Prototipe dan Fitur Deteksi dari Citra Foto Langsung

Pengujian citra foto secara langsung akan dilakukan dengan pengambilan foto retina dari beberapa subjek penelitian. Subjek penelitian akan ditanya terkait beberapa informasi sebagai berikut:

- a. Subjek sudah pernah didiagnosis mengidap diabetes;
- b. Subjek sudah pernah didiagnosis mengidap diabetes retinopati;
- c. Subjek memiliki riwayat kadar kolesterol tinggi;
- d. Subjek memiliki riwayat tekanan darah tinggi;
- e. Subjek memiliki kebiasaan merokok;
- f. Subjek kadar gula darah tidak terkontrol;
- g. Subjek sering buang air kecil terutama di malam hari;
- h. Subjek sering merasa haus;

- i. Subjek sering merasa lapar;
- j. Subjek merasakan penglihatan kabur;
- k. Subjek merasa berat badan menurun.

Pertanyaan pada poin a dan b digunakan sebagai acuan apakah hasil pendeteksian memang sesuai dengan pernyataan subjek atau tidak. Sementara pertanyaan poin c hingga k digunakan bila subjek tidak tau kondisi atau jawaban terkait pertanyaan poin a dan b.

Tabel 3.6 Tabel spesifikasi lux meter

No.	Spesifikasi	Keterangan
1.	Merek	Smart Sensor
2.	Tipe	AS803
3.	Rentan pengukuran	1 – 200.000 Lux
4.	Akurasi	$\leq 10.000 \text{ Lux}, \pm (4\% + 10 \text{ word})$ $> 10.000 \text{ Lux}, \pm (5\% + 10 \text{ word})$
5.	<i>Repeatability</i>	$\pm 2\%$
6.	Daya listrik	3 x 1,5 volt AAA <i>battery</i>
7.	Suhu operasional	0°C – 40°C

Pencahayaan ruangan perlu dikondisikan terlebih dahulu sebab menjadi variabel bebas (*independent variable*). Nilai dari variabel pencahayaan ini sendiri akan dibagi menjadi 3 yaitu 50 – 150 lux, 151 – 250 lux, 251 – 350 lux. Dalam mengontrol variabel pencahayaan digunakan alat bernama lux meter. Lux meter yang digunakan memiliki spesifikasi yang tertera pada Tabel 3.6.

3.8.4 Pengujian Hasil Pendeteksian Berdasarkan Parameter Uji

Pengujian hasil pendeteksian berdasarkan parameter dilakukan bersamaan dengan pengujian citra foto secara langsung. Parameter-parameter yang akan diujikan terdapat pada Tabel 3.7.

Tabel 3.7 Tabel pengujian parameter terhadap hasil pendeteksian

No.	Aspek Pengujian	Nilai Uji
1.	Kecerahan	1 – 100 lux, 101 – 200 lux, 201 – 300 lux
2.	Penambahan <i>flash</i> kamera	Ada <i>flash</i> , tanpa <i>flash</i>

BAB IV

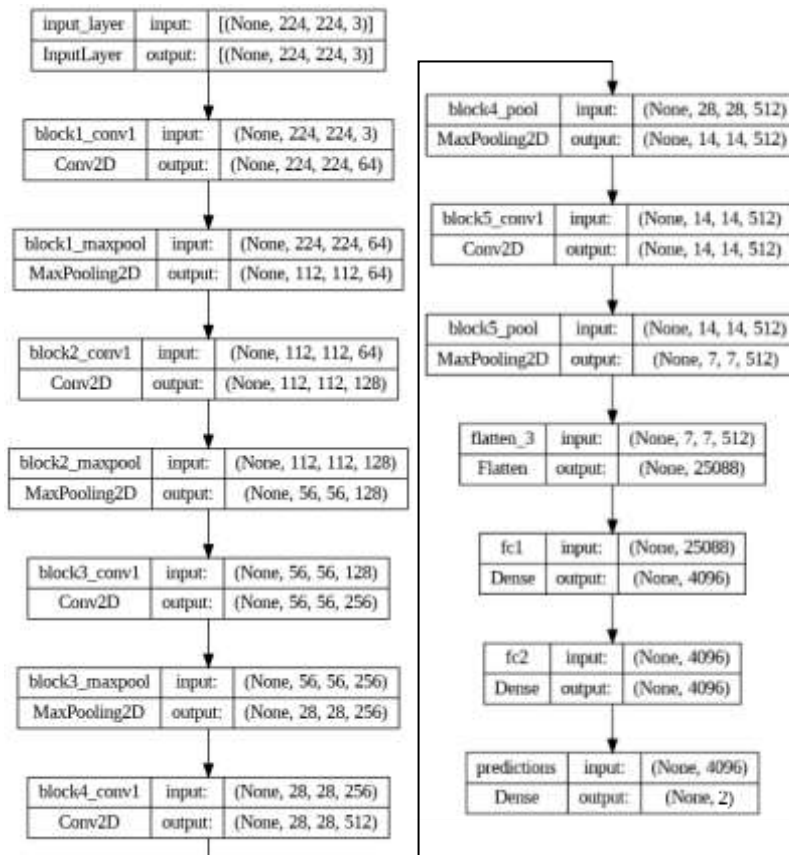
Hasil dan Pembahasan

4.1 Penyederhanaan Arsitektur Model

Berdasarkan hasil yang ditemukan dalam proses penelitian, didapati bahwa diperlukan adanya penyederhanaan model dari bentuk asli. Terdapat dua hasil model penyederhanaan.

4.1.1 Arsitektur VGG-19 Sederhana 1

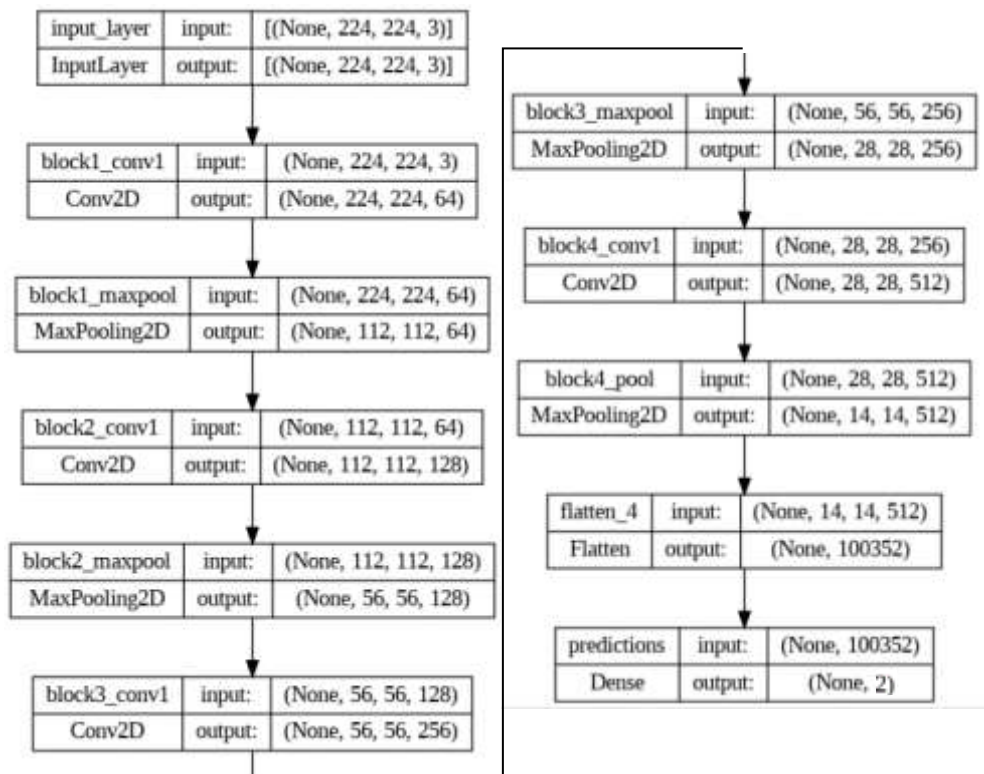
Gambar 4.1 merupakan bentuk arsitektur model VGG-19 sederhana 1 yang telah dibuat. Pada model tersebut, citra RGB berukuran $224 \times 224 \text{ pixel}$ akan diolah melalui lima blok pengolahan yang kemudian dilakukan *flatten* dan *dense* sehingga menghasilkan satu keluaran. Masing-masing blok terdiri dari satu *layer convolution 2D* dan diakhiri *max pooling layer* dengan dimensi yang dapat dilihat pada Gambar 4.1.



Gambar 4.1 Arsitektur VGG-19 sederhana 1

4.1.2 Arsitektur VGG-19 Sederhana 2

Gambar 4.2 merupakan bentuk arsitektur model VGG-19 sederhana 2 yang telah dibuat. Pada model tersebut, citra RGB berukuran 224x224 *pixel* akan diolah melalui empat blok pengolahan yang kemudian dilakukan *flatten* dan *dense* sehingga menghasilkan satu keluaran. Masing-masing blok terdiri dari satu *layer convolution 2D* dan diakhiri *max pooling layer* dengan dimensi yang dapat dilihat pada Gambar 4.2.



Gambar 4.2 Arsitektur VGG-19 sederhana 2

4.2 Hasil Arsitektur Model

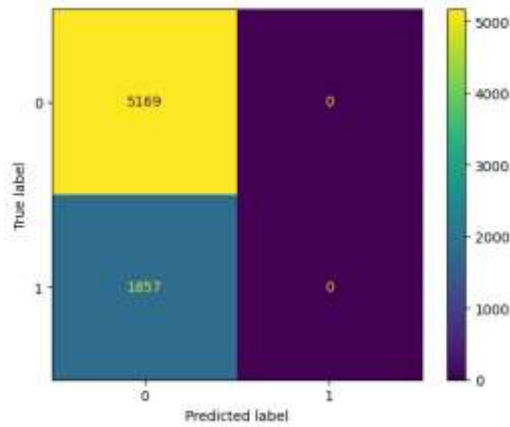
Pada proyek Tugas Akhir ini, penulis melakukan beberapa penelitian terkait model arsitektur CNN sebelum ditanamkan pada aplikasi android. Metode dan hasil percobaan ditampilkan pada Tabel 4.1 dan Tabel 4.2. Sementara untuk hasil dari *convolution matrix* dari percobaan pembuatan model deteksi yang sudah dibuat ditampilkan pada Gambar 4.3 hingga Gambar 4.10.

Tabel 4.1 Tabel percobaan pembuatan arsitektur dan hasilnya (bagian 1)

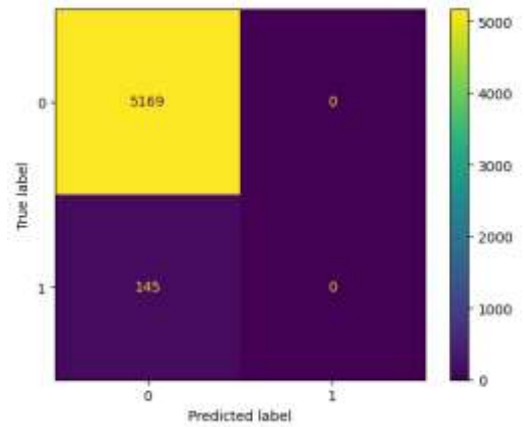
Percobaan	Arsitektur CNN	Dataset	Modeling	Akurasi dan Loss	Waktu Learning	Ukuran Model
P1	VGG-19	Data 1	$learning\ rate = 10^{-4}$ $step\ per\ epoch = 30$ $epoch = 10\ validate\ step = 30$	Train loss : 0.6 Train acc : 0.71 Test loss : 0.57 Test acc : 0.75	17751 detik = 4 jam 55 menit 51 detik	1.573.554 kb
P2	VGG-19	Data 2	$learning\ rate = 10^{-4}$ $step\ per\ epoch = 50$ $epoch = 20\ validate\ step = 30$	Train loss : 0.11 Train acc : 0.98 Test loss : 0.15 Test acc : 0.96	17632 detik = 4 jam 53 menit 52 detik	1.573.554 kb
P3	VGG-19	Data 3	$learning\ rate = 10^{-4}$ $step\ per\ epoch = 40$ $epoch = 10\ validate\ step = 30$	Train loss : 0.69 Train acc : 0.51 Test loss : 0.68 Test acc : 0.64	8211 detik = 2 jam 16 menit 51 detik	1.573.554 kb
P4	ResNet-50	Data 3	$learning\ rate = 10^{-4}$ $step\ per\ epoch = 40$ $epoch = 10\ validate\ step = 30$	Train loss : 1.37 Train acc : 0.49 Test loss : 0.44 Test acc : 0.80	6065 detik = 1 jam 41 menit 5 detik	276.945 kb
P5	VGG-19	Data 4	$learning\ rate = 10^{-4}$ $step\ per\ epoch = 40$ $epoch = 10\ validate\ step = 30$	Train loss : 0.16 Train acc : 0.93 Test loss : 0.20 Test acc : 0.92	12967 detik = 3 jam 36 menit 7 detik	1.573.554 kb

Tabel 4.2 Tabel percobaan pembuatan arsitektur dan hasilnya (bagian 2)

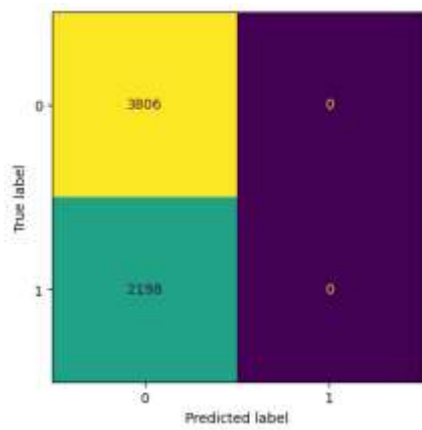
Percobaan	Arsitektur CNN	Dataset	Modeling	Akurasi dan Loss	Waktu Learning	Ukuran Model
P6	ResNet-50	Data 4	$learning\ rate = 10^{-4}$ $step\ per\ epoch = 40$ $epoch = 10\ validate\ step = 30$	Train loss : 0.14 Train acc : 0.95 Test loss : 0.18 Test acc : 0.94	5076 detik = 1 jam 24 menit 36 detik	276.945 kb
P7	VGG-19	Data 4	$learning\ rate = 10^{-4}$ $step\ per\ epoch = 50$ $epoch = 20\ validate\ step = 30$	Train loss : 0.15 Train acc : 0.95 Test loss : 0.18 Test acc : 0.93	26980 detik = 7 jam 29 menit 40 detik	1.573.554 kb
P8	VGG-19 sederhana 1	Data 4	$learning\ rate = 10^{-4}$ $step\ per\ epoch = 40$ $epoch = 20\ validate\ step = 30$	Train loss : 0.14 Train acc : 0.96 Test loss : 0.16 Test acc : 0.95	4211 detik = 1 jam 10 menit 11 detik	1.446.885 kb
P9 (Try_Model_4.h5)	VGG-19 sederhana 2	Data 4	$learning\ rate = 10^{-4}$ $step\ per\ epoch = 40$ $epoch = 20\ validate\ step = 30$	Train loss : 0.15 Train acc : 0.95 Test loss : 0.18 Test acc : 0.94	4244 detik = 1 jam 27 menit 24 detik	19.405 kb



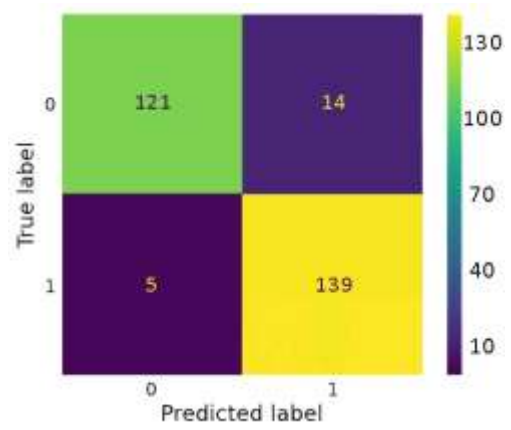
Gambar 4.3 Convolution matrix P1



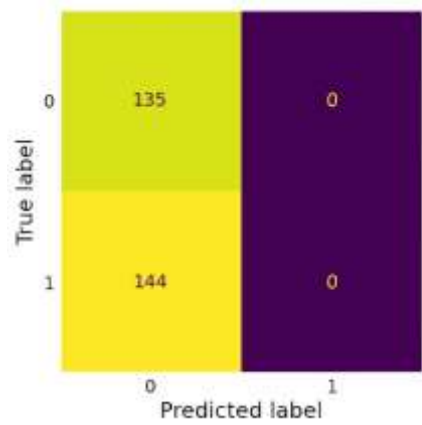
Gambar 4.4 Convolution matrix P1



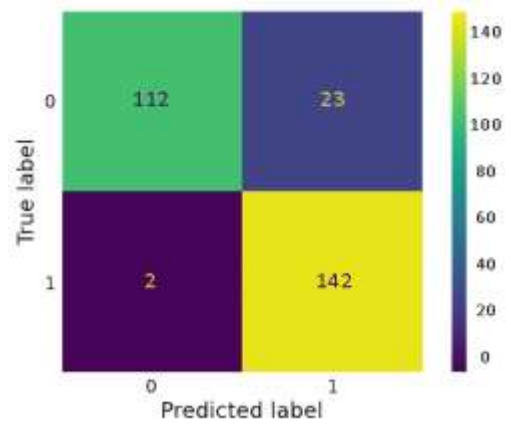
Gambar 4.5 Convolution matrix
P3 dan P4



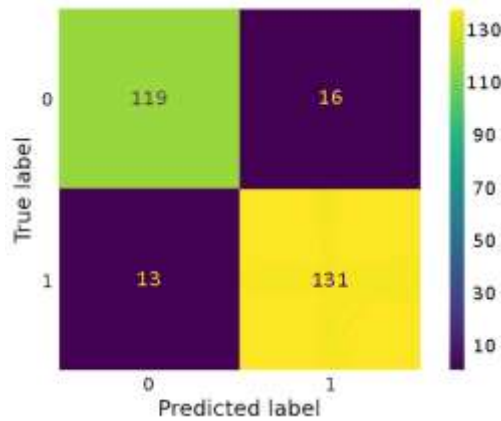
Gambar 4.6 Convolution matrix P5



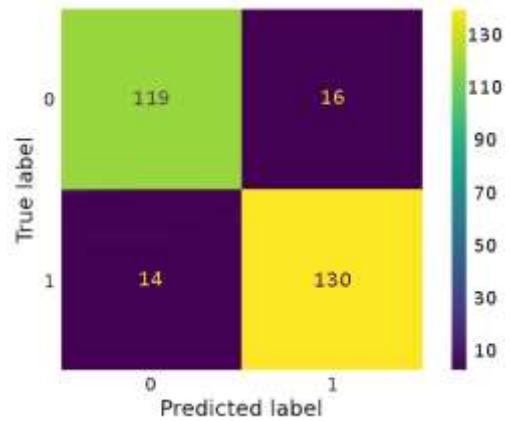
Gambar 4.7 Convolution matrix P6



Gambar 4.8 Convolution matrix P7



Gambar 4.9 Convolution matrix P8



Gambar 4.10 Convolution matrix P9

Berdasarkan hasil yang ditunjukkan pada Tabel 4.1 dan Tabel 4.2, dapat disimpulkan bahwa dalam pembuatan model deteksi, yang memiliki akurasi paling tinggi ditunjukkan oleh P2 disusul P8 dan P9. Berdasarkan kecepatan proses pembelajaran maka urutannya adalah P8 disusul P6 dan P9. Sementara itu, berdasarkan ukuran model yang paling ringan maka urutannya adalah P9 disusul P4 dan P6. Berdasarkan matriks konvolusi, P1 hingga P4 serta P6 menunjukkan hasil deteksi ke No_DR seluruhnya. Sementara itu, matriks konvolusi P5, P7 hingga P9 berhasil melakukan prediksi dengan benar. Untuk memastikan kemampuan model, maka model dipanggil untuk menguji beberapa citra dan didapatkan hasil seperti pada Tabel 4.4. Tabel 4.3 merupakan keterangan *path* dari data yang diambil sebagai data uji.

Tabel 4.3 Tabel keterangan untuk folder data yang diuji

Label	Path	Label	Path
T1	data4/val/DR	T5	data3/val/No DR
T2	data4/val/No_DR	T6	data3/test/No DR
T3	data3/val/Proliferative DR	T7	data4/test/No_DR
T4	data3/test/Proliferative DR		

Tabel 4.4 Tabel perbandingan hasil prediksi model CNN

Percobaan	Hasil Deteksi					
P1	T1	No_DR - 113	DR - 0	T5	No_DR - 200	DR - 0
	T2	No_DR - 118	DR - 0	T6	No_DR - 200	DR - 0
	T3	No_DR - 200	DR - 0	T7	No_DR - 200	DR - 0
	T4	No_DR - 200	DR - 0			
P2	T1	No_DR - 113	DR - 0	T5	No_DR - 200	DR - 0
	T2	No_DR - 118	DR - 0	T6	No_DR - 200	DR - 0
	T3	No_DR - 200	DR - 0	T7	No_DR - 200	DR - 0
	T4	No_DR - 200	DR - 0			
P3	T1	No_DR - 113	DR - 0	T5	No_DR - 200	DR - 0
	T2	No_DR - 118	DR - 0	T6	No_DR - 200	DR - 0
	T3	No_DR - 200	DR - 0	T7	No_DR - 200	DR - 0
	T4	No_DR - 200	DR - 0			
P4	T1	No_DR - 113	DR - 0	T5	No_DR - 200	DR - 0
	T2	No_DR - 118	DR - 0	T6	No_DR - 200	DR - 0
	T3	No_DR - 200	DR - 0	T7	No_DR - 200	DR - 0
	T4	No_DR - 200	DR - 0			
P5	T1	No_DR - 13	DR - 100	T5	No_DR - 164	DR - 36
	T2	No_DR - 113	DR - 5	T6	No_DR - 159	DR - 41
	T3	No_DR - 123	DR - 77	T7	No_DR - 190	DR - 10
	T4	No_DR - 163	DR - 37			
P6	T1	No_DR - 113	DR - 0	T5	No_DR - 200	DR - 0
	T2	No_DR - 118	DR - 0	T6	No_DR - 200	DR - 0
	T3	No_DR - 200	DR - 0	T7	No_DR - 200	DR - 0
	T4	No_DR - 200	DR - 0			
P7	T1	No_DR - 21	DR - 92	T5	No_DR - 179	DR - 21
	T2	No_DR - 116	DR - 2	T6	No_DR - 187	DR - 13
	T3	No_DR - 174	DR - 26	T7	No_DR - 195	DR - 5
	T4	No_DR - 186	DR - 14			
P8	T1	No_DR - 14	DR - 99	T5	No_DR - 136	DR - 64
	T2	No_DR - 107	DR - 11	T6	No_DR - 136	DR - 64
	T3	No_DR - 116	DR - 84	T7	No_DR - 175	DR - 25
	T4	No_DR - 129	DR - 71			
P9	T1	No_DR - 14	DR - 99	T5	No_DR - 180	DR - 20
	T2	No_DR - 109	DR - 9	T6	No_DR - 180	DR - 20
	T3	No_DR - 133	DR - 67	T7	No_DR - 186	DR - 14
	T4	No_DR - 158	DR - 42			

Berdasarkan Tabel 4.3, percobaan P1, P2, P3, P4, dan P6 mendeteksi citra No_DR sepenuhnya. Sementara percobaan P5, P7, P8, dan P9 terlihat adanya

variasi deteksi. Meski demikian, dapat dilihat untuk ketiga percobaan tersebut pada data T3 hingga T6 masih dominan untuk menebak No_DR. Jika melihat gagalnya proses pembuatan model P3 dan P4 yang menggunakan data 3, maka ada kemungkinan masih terdapat kekurangan pada data tersebut. Maka dari itu, untuk menghindari kegagalan, maka dianjurkan menggunakan model dan melakukan pengujian dengan data 4 sebagai acuan. Maka dari itu, berdasarkan Tabel 4.4, dapat disederhanakan untuk mengambil model dan data uji yang mengacu data 4 sekaligus menyisihkan data yang tampak memiliki variasi (model: P5, P7, P8, dan P9; data uji: T1, T2, dan T7). Hasil penyederhanaan Tabel 4.4 dapat dilihat pada Tabel 4.5.

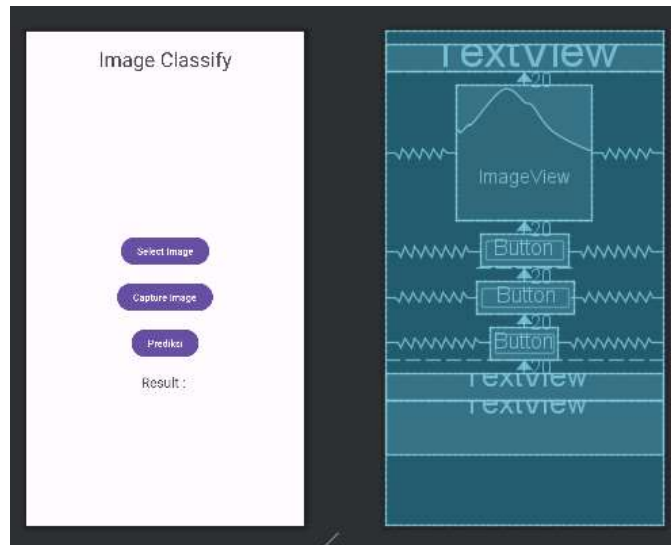
Tabel 4.5 Tabel rata-rata prediksi benar dari model yang dipilih

Percobaan	Hasil Deteksi			
P5	T1	No_DR - 13	DR - 100	Benar : 88,495%
	T2	No_DR - 113	DR - 5	Benar : 95,763%
	T7	No_DR - 190	DR - 10	Benar : 95,000%
	Rata-rata benar = 93,086%			
P7	T1	No_DR - 21	DR - 92	Benar : 81,416%
	T2	No_DR - 116	DR - 2	Benar : 98,305%
	T7	No_DR - 195	DR - 5	Benar : 97,500%
	Rata-rata benar = 92,407%			
P8	T1	No_DR - 14	DR - 99	Benar : 87,611%
	T2	No_DR - 107	DR - 11	Benar : 90,678%
	T7	No_DR - 175	DR - 25	Benar : 87,500%
	Rata-rata benar = 88,596%			
P9	T1	No_DR - 14	DR - 99	Benar : 87,611%
	T2	No_DR - 109	DR - 9	Benar : 92,373%
	T7	No_DR - 186	DR - 14	Benar : 83,000%
	Rata-rata benar = 87,661%			

Berdasarkan hasil pendeteksian dari P5, P7, P8, dan P9, didapati bahwa model dengan akurasi tertinggi dimiliki oleh P5. Namun, karena batasan model yang bisa diterima oleh Android Studio, maka dipilihlah model dari P9 yang memiliki selisih akurasi sebesar $\pm 6\%$ dibanding model P5. Meski demikian, model dari P9 masih dianggap layak sebab memiliki persentase keberhasilan pendeteksian di atas 85%.

4.3 Pembuatan Aplikasi Android

Aplikasi berbasis android ini dikembangkan dengan memanfaatkan Android Studio IDE. Desain pembuatan tampilan dari aplikasi dapat dilihat pada Gambar 4.11.



Gambar 4.11 Tata letak tampilan menu utama aplikasi

Terdapat beberapa fungsi penting untuk mengatur interaksi tombol dan tampilan lainnya pada aplikasi tersebut. Beberapa fungsi tersebut dapat dilihat pada Gambar 4.12.

```
064 // Set button to active gallery view click action, and it
    used for assign request code
065 selectBtn.setOnClickListener(new View.OnClickListener() {
066     @Override
067     public void onClick(View view) {
068         Intent intent = new Intent(Intent.ACTION_PICK,
            MediaStore.Image.Media.EXTERNAL_CONTENT_URI);
069         intent.setAction(Intent.ACTION_GET_CONTENT);
070         intent.setType("image/*");
071         startActivityForResult(intent, 10);
072     }
073 });
074
075 // Set button to active camera view click action, and it used
    for assign request code
076 captureBtn.setOnClickListener(new View.OnClickListener() {
077     @Override
078     public void onClick(View view) {
079         Intent intent = new Intent(Intent.ACTION_PICK,
            MediaStore.Image.Media.EXTERNAL_CONTENT_URI);
080         startActivityForResult(intent, 12);
081     }
082 });
```

Gambar 4.12 Potongan kode fungsi interaksi tombol dan tampilan

Potongan kode pada Gambar 4.12 baris 52 hingga 60 merupakan fungsi untuk membuka galeri *handphone* untuk mengambil gambar yang tersimpan untuk dideteksi. Fungsi tersebut akan diaktifkan jika pengguna menekan tombol “*Select Image*”. Baris 62 hingga 68 merupakan fungsi untuk mengaktifkan tampilan kamera seperti yang terdapat pada Gambar 4.12. Fungsi tersebut akan diaktifkan jika pengguna menekan tombol “*Capture Image*”. Selanjutnya, dalam kedua fungsi tersebut terdapat pemanggilan fungsi *startActivityResult* dan mengirimkan *requestCode* yang berbeda. Fungsi ini digunakan untuk mengaktifasi perintah apa yang akan dijalankan selanjutnya jika tombol bersangkutan ditekan.

```

184 public void onActivityResult(int requestCode, int
    resultCode, @Nullable Intent data) {
185     super.onActivityResult(requestCode, resultCode, data);
186     if (resultCode == RESULT_OK) {
187         // Trigger when button Choose Image clicked and
            requestcode change 10
188         if (requestCode == 10 && data != null) {
189             Uri uri = data.getData();
190             try {
191                 bitmap = MediaStore.Images.Media.get
                    Bitmap(this.getContentResolver(), uri);
192                 bitmap = Bitmap.createScaledBitmap(
                        bitmap, 224, 224, false);
193                 imageView.setImageBitmap(bitmap);
194             } catch (IOException e) {
195                 e.printStackTrace();
196             }
197         }
198
199         // Trigger when button Capture Image clicked and
            requestcode change 12
200         if (requestCode == 12 && data != null) {
201             bitmap = (Bitmap) data.getExtras().get
                    ("data");
202             int dimension = Math.min(bitmap.getWidth(),
                    bitmap.getHeight());
203             bitmap = ThumbnailUtils.extractThumbnail
                    (bitmap, dimension, dimension);
204             bitmap = Bitmap.createScaledBitmap(bitmap,
                    224, 224, false);
205             imageView.setImageBitmap(bitmap);
206         }
207     }
208 }

```

Gambar 4.13 Potongan kode fungsi *onActivityResult*

Fungsi *startActivityResult* dapat dilihat pada Gambar 4.13. Pada potongan kode tersebut, jika *requestCode* = 10, maka akan dijalankan proses pembukaan galeri, sementara jika *requestCode* = 12, maka akan dijalankan proses pembukaan

kamera untuk foto. Gambar yang dipilih dari galeri atau berhasil ditangkap dari kamera akan disimpan dalam variabel *bitmap* dengan *type* data *bitmap*. Data tersebut kemudian dibuat ukurannya menjadi persegi dengan ukuran 224x224 yang dapat dilihat pada baris 157 dan 169.

Gambar 4.14 dan Gambar 4.15 menampilkan fungsi yang akan aktif saat tombol “Prediksi” dipilih. Baris 76 merupakan proses pemanggilan model CNN. Baris 79 hingga 97 merupakan proses persiapan *buffer* untuk mengolah data gambar RGB yang diterima model. Baris 99 hingga 101 merupakan proses pendeteksian citra. Baris 103 hingga 111 merupakan bagian untuk menentukan indeks tertinggi dan menentukan label citra dan tingkat akurasi. Baris 113 dan 115 berfungsi menampilkan hasil dari pendeteksian.

```

085 predictBtn.setOnClickListener(new View.OnClickListener() {
086     @Override
087     public void onClick(View view) {
088         String text = new String();
089         String[] classes = {"DR", "No DR"};
090         int pred = 0;
091
092         // This is for try catch if there was an image or it
093         // wasn't
094         try {
095             pred = bitmap.getBytesCount();
096         }
097         catch (Exception e) {
098             result.setText("Choose the image");
099         }
100
101         if (pred > 0) {
102             // Detection process
103             try {
104                 ModelFix model = ModelFix.newInstance(
105                     MainActivity.this);
106
107                 // Creates inputs for reference
108                 TensorBuffer inputFeature0 = Tensor
109                 Buffer.createFixedSize(new int[]{1,224,
110                 224, 3}, DataType.FLOAT32);
111
112                 // Prepare for buffer image
113                 ByteBuffer byteBuffer = ByteBuffer.
114                 allocateDirect(4 * 224 * 224 * 3);
115                 byteBuffer.order(ByteOrder.nativeOrder());
116
117                 // Match the data pixel into the buffer
118                 int [] intValues = new int [224 * 224];

```

Gambar 4.14 Potongan kode fungsi prediksi dan pemanggilan model (bagian 1)

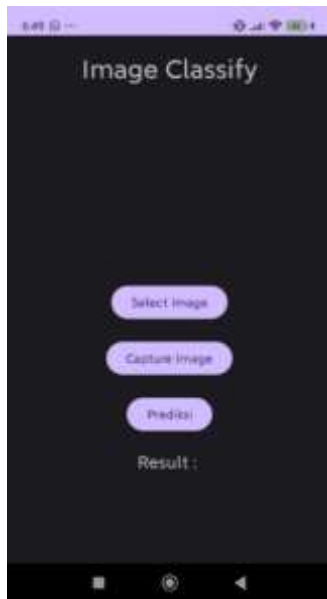
```

114         bitmap.getPixels(intValues, 0, bitmap.getWidth(), 0, 0, bitmap.getHeight(), bitmap.getHeight());
115         int pixel = 0;
116         for (int i = 0; i < 224; i++) {
117             for (int j = 0; j < 224; j++) {
118                 int val = intValues[pixel++];
119                 byteBuffer.putFloat(((val >> 16) & 0xFF) * (1.f / 1));
120                 byteBuffer.putFloat(((val >> 8) & 0xFF) * (1.f / 1));
121                 byteBuffer.putFloat((val & 0xFF) * (1.f / 1));
122             }
123         }
124
125         inputFeature0.loadBuffer(byteBuffer);
126
127         // Runs model inference and gets result.
128         ModelFix.Outputs outputs = model.process(inputFeature0);
129         TensorBuffer outputFeature0 = outputs.getOutputFeature0AsTensorBuffer();
130
131         // Get prediction and confidence
132         float [] confidences = outputFeature0.getFloatArray();
133         int maxPos = 0;
134         float maxConfidence = 0;
135         for (int i = 0; i < confidences.length; i++) {
136             if (confidences[i] > maxConfidence) {
137                 maxConfidence = confidences[i];
138                 maxPos = i;
139             }
140         }
141
142         // Set the output of prediction
143         text = classes[maxPos].concat("\n").concat(String.valueOf(maxConfidence * 100)).concat("%");
144         result.setText(text);
145
146         // Releases model resources if no longer used.
147         model.close();
148     } catch (IOException e) {
149         // TODO Handle the exception
150     }
151 }
152 }
153 });

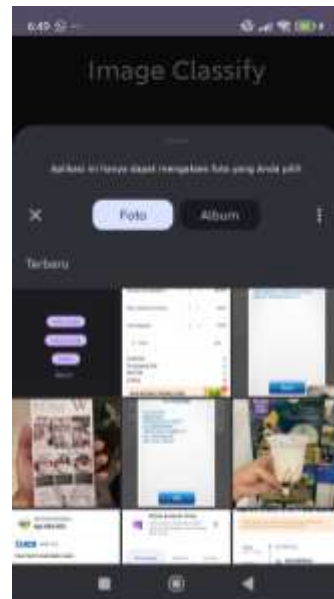
```

Gambar 4.15 Potongan kode fungsi prediksi dan pemanggilan model (bagian 2)

Hasil dari tampilan aplikasi yang sudah dibuat dapat dilihat pada Gambar 4.16 hingga Gambar 4.19.



Gambar 4.16 Tampilan menu utama aplikasi



Gambar 4.17 Tampilan aplikasi tombol “Select Image”

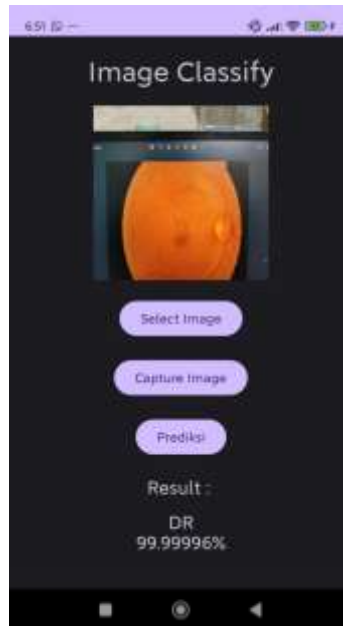


(a)



(b)

Gambar 4.18 Tampilan aplikasi tombol “Capture Image” (a) sebelum ambil gambar (b) setelah ambil gambar



Gambar 4.19 Tampilan aplikasi tombol “Prediksi”

4.4 Pengujian Deteksi dengan Aplikasi Android

4.4.1 Pengujian Deteksi Menggunakan Citra dari *Gallery*

Tabel 4.6 hingga Tabel 4.11 merupakan tabel hasil pendeteksian diabetes retinopati melalui aplikasi menggunakan citra dari Data 4 *Test* yang diunggah ke aplikasi. Tabel 4.6 hingga Tabel 4.8 menunjukkan data citra No_DR dengan hasil pendeteksian dan akurasi. Tabel 4.9 hingga Tabel 4.11 menunjukkan data citra DR dengan hasil pendeteksian dan akurasi.

Berdasarkan Tabel 4.6 hingga Tabel 4.8, dari 20 citra No_DR terdapat 6 citra yang dideteksi sebagai DR. Data tersebut menunjukkan keberhasilan untuk mendeteksi citra No_DR sebesar $\frac{14}{20} \times 100\% = 70\%$. Sementara itu, Berdasarkan Tabel 4.9 hingga Tabel 4.11, dari 20 citra DR terdapat 4 citra yang dideteksi sebagai No_DR. Data tersebut menunjukkan keberhasilan untuk mendeteksi citra DR sebesar $\frac{16}{20} \times 100\% = 80\%$.


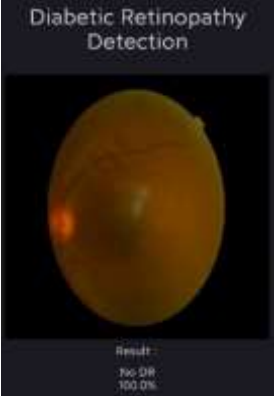


Tabel 4.6 Tabel hasil deteksi citra No_DR dari memori *handphone* (bagian 1)

Gambar hasil deteksi								
	No_DR	100%	No_DR	100%	No_DR	100%	No_DR	99.99%
Gambar hasil deteksi								
	DR	100%	DR	99.99%	DR	99.84%	DR	97.91%

Tabel 4.7 Tabel hasil deteksi citra No_DR dari memori *handphone* (bagian 2)

Gambar hasil deteksi								
	Hasil deteksi	No_DR	99.46%	No_DR	100%	DR	99.69%	DR
Gambar hasil deteksi								
	Hasil deteksi	No_DR	99.99%	No_DR	100%	No_DR	99.97%	No_DR

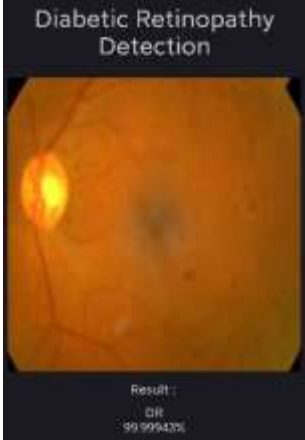



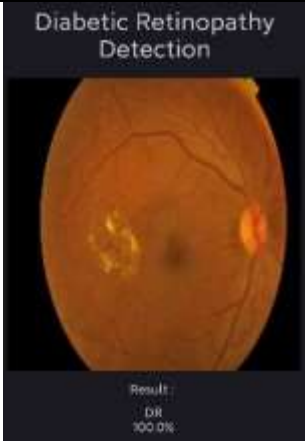



Tabel 4.8 Tabel hasil deteksi citra No_DR dari memori *handphone* (bagian 3)

Gambar hasil deteksi								
	Hasil deteksi	No_DR	99.99%	No_DR	100%	No_DR	99.93%	No_DR


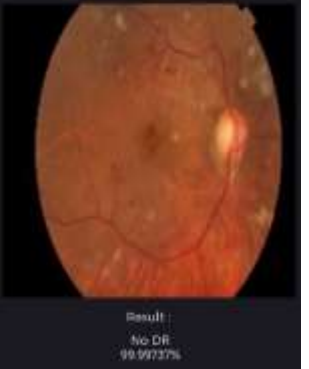

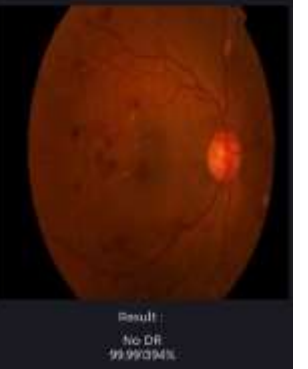
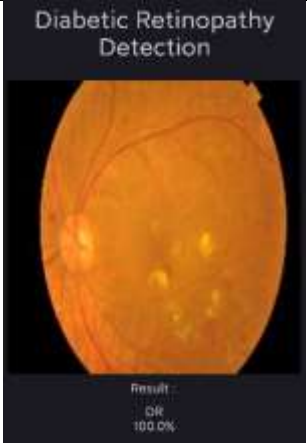

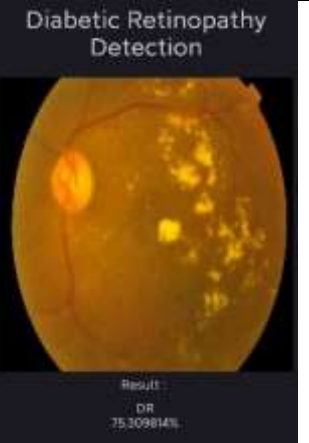
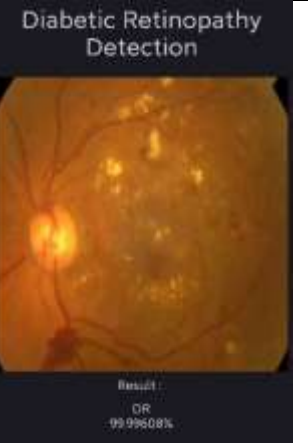
Tabel 4.9 Tabel hasil deteksi citra DR dari memori *handphone* (bagian 1)

Gambar hasil deteksi								
	Hasil deteksi	DR	100%	DR	88.65%	DR	99.99%	No_DR

Tabel 4.10 Tabel hasil deteksi citra DR dari memori *handphone* (bagian 2)

Gambar hasil deteksi								
	Hasil deteksi	DR	99.99%	DR	97.51%	No_DR	99.99%	DR
Gambar hasil deteksi								
	Hasil deteksi	DR	99.99%	DR	100%	DR	99.97%	DR










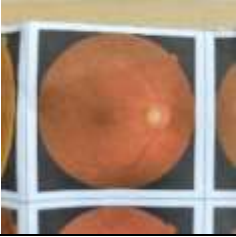
Tabel 4.11 Tabel hasil deteksi citra DR dari memori *handphone* (bagian 3)

Gambar hasil deteksi								
	Hasil deteksi	DR	100%	DR	100%	No_DR	99.99%	DR
Gambar hasil deteksi								
	Hasil deteksi	No_DR	99.99%	DR	100%	DR	99.82%	DR

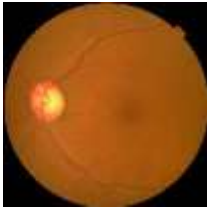

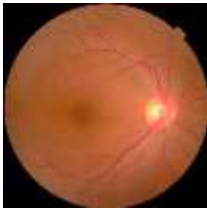











4.4.2 Pengujian Deteksi dengan Kamera pada Citra Gambar Cetak

Tabel 4.12 hingga Tabel 4.15 merupakan tabel hasil pendeteksian diabetes retinopati melalui aplikasi menggunakan citra dari data 4 Test yang diunggah ke aplikasi. Tabel 4.12 dan Tabel 4.13 menunjukkan data citra No_DR dengan hasil pendeteksian dan akurasi. Tabel 4.14 dan Tabel 4.15 menunjukkan data citra DR dengan hasil pendeteksian dan akurasi.

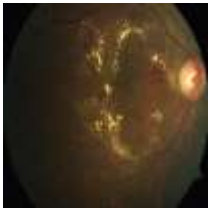



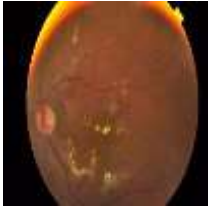

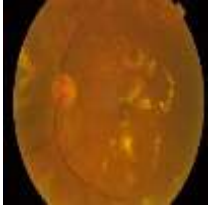





Tabel 4.12 Tabel hasil deteksi citra No_DR dari gambar cetak (bagian 1)

Citra asli	Citra Kamera Foto	Hasil Deteksi		
		Pertama	Kedua	Ketiga
		No_DR 99,96%	DR 58,76%	No_DR 100,0%
		No_DR 100,0%	No_DR 99,99%	No_DR 100,0%
		No_DR 100,0%	DR 96,53%	DR 82,81%
		No_DR 100,0%	No_DR 99,99%	No_DR 52,72%
		No_DR 100,0%	No_DR 99,99%	No_DR 100,0%



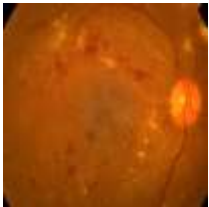

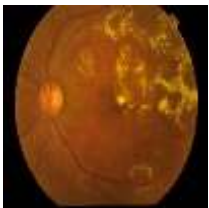



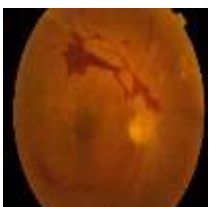



Tabel 4.13 Tabel hasil deteksi citra No_DR dari gambar cetak (bagian 2)

Citra asli	Citra Kamera Foto	Hasil Deteksi		
		Pertama	Kedua	Ketiga
		No_DR 99,99%	No_DR 99,96%	No_DR 100,0%
		No_DR 100,0%	No_DR 100,0%	No_DR 99,63%
		No_DR 99,98%	No_DR 99,97%	No_DR 99,48%
		No_DR 99,84%	No_DR 99,99%	No_DR 99,47%
		No_DR 100,0%	No_DR 99,97%	No_DR 99,99%
		No_DR 99,99%	No_DR 100,0%	No_DR 100,0%
		No_DR 96,64%	No_DR 94,88%	No_DR 100,0%

Tabel 4.14 Tabel hasil deteksi citra DR dari kamera pada gambar cetak (bagian 1)

Citra asli	Citra Kamera Foto	Hasil Deteksi		
		Pertama	Kedua	Ketiga
		No_DR 100,0%	No_DR 99,98%	No_DR 99,99%
		DR 99,97%	DR 100,0%	DR 99,93%
		No_DR 99,99%	No_DR 99,99%	DR 99,97%
		DR 100,0%	DR 83,12%	No_DR 99,99%
		No_DR 100,0%	No_DR 99.73%	No_DR 100,0%
		DR 95,17%	DR 99,99%	No_DR 99,99%

Tabel 4.15 Hasil deteksi citra DR dari kamera pada gambar cetak (bagian 2)

Citra asli	Citra Kamera Foto	Hasil Deteksi		
		Pertama	Kedua	Ketiga
		DR 99,99%	DR 100,0%	DR 100,0%
		DR 100,0%	DR 99,99%	DR 96,33%
		DR 99,99%	DR 99,05%	DR 100,0%
		DR 67,68%	No_DR 99,07%	DR 99,99%
		DR 100,0%	DR 99,13%	DR 99,96%
		DR 96,99%	No_DR 91,76%	DR 95,20%

Setiap citra dideteksi sebanyak tiga kali untuk mendapat hasil yang lebih akurat. Maka dari itu, data akan mengikuti hasil pendeteksian yang diakui setidaknya dua kali. Berdasarkan data pada Tabel 4.12 dan Tabel 4.13, terdapat 10 data yang berhasil menebak 3 kali No_DR, 1 data yang berhasil menebak 2 kali No_DR, dan 1 data yang berhasil menebak 1 kali No_DR. Berdasarkan data tersebut didapati keberhasilan untuk memprediksi data No_DR sebesar $\frac{11}{12} \times 100\% = 91,67\%$.

Berdasarkan data pada Tabel 4.14 dan Tabel 4.15, terdapat 5 data yang berhasil menebak 3 kali DR, 4 data yang berhasil menebak 2 kali DR, dan 3 data yang berhasil menebak 0 kali DR. Berdasarkan data tersebut didapati keberhasilan untuk memprediksi data DR sebesar $\frac{9}{12} \times 100\% = 75,00\%$. Maka dari itu, dapat disimpulkan aplikasi dapat melakukan pendeteksian No_DR sebesar 91,67%, DR sebesar 75,00%, dan keberhasilan rata-rata sebesar 83,33%.

4.4.3 Pengujian Deteksi dengan Kamera pada Citra Retina Asli

Pada tahap ini, peneliti melakukan penerapan rancangan dengan desain yang tampak pada Gambar 4.20.



Gambar 4.20. Penerapan perangkat keras *smartphone* dan lensa 20D

Pada Gambar 4.20. Bagian A merupakan lensa 20D, bagian B merupakan perangkat *smartphone*, dan bagian C merupakan adaptor dengan panjang lengan maksimal 25 cm.

Pengujian pada subjek dilakukan dengan metode dan langkah sebagai berikut:

4.4.3.1 Pengujian seperti rancangan awal

1. Meneteskan cairan *Cendo Mydiatil* 1% yang berfungsi untuk sikloplegia dan midriasis (memperbesar lubang pupil) sehingga pantulan cahaya dapat diterima hingga ke bagian fundus.
2. Melakukan pengujian untuk mendapat citra fundus. Pada kali ini digunakan video.
 - a. Metode pertama, citra diambil menggunakan video dengan keadaan ruangan terang (lampu ruangan menyala (± 160 lux)). Kondisi lengan adaptor dipasang maksimal (25cm) dan jarak mata ke lensa ± 8 cm. Pada metode ini, citra fundus tidak berhasil didapatkan. Ada pun hasil yang didapatkan hanya sebatas pupil mata saja. Hipotesis pada tahap ini diperkirakan akibat adanya sumber cahaya lain sehingga fundus memantulkan cahaya dari berbagai arah dan tidak terfokus.



Gambar 4.21 Penangkapan citra fundus dengan lensa 20D, adaptor, dan *smartphone*

- b. Maka dicobalah metode kedua, yaitu dengan mematikan lampu ruangan dan melakukan video untuk menangkap citra fundus dengan penambahan *flash* kamera. Kondisi lengan

adaptor dipasang maksimal (25cm) dan jarak mata ke lensa ± 8 cm. Namun, pada percobaan ini, citra fundus masih tidak berhasil didapatkan. Hipotesis gagalnya penangkapan citra fundus akibat jarak mata, lensa 20D, dan kamera *smartphone* yang kurang bisa leluasa untuk diubah. Hasil pengambilan citra pada metode ini dapat dilihat pada Gambar 4.21.

Karena kegagalan dua metode tersebut, peneliti melanjutkan dengan beberapa tahap sebagai berikut.

4.4.3.2 Peneliti dengan lensa 20D dan *smartphone* tanpa adaptor



Gambar 4.22 Penangkapan citra fundus dengan lensa 20D dan *smartphone*



Gambar 4.23 Kondisi ruangan pengambilan data sampel

Peneliti dilanjutkan dengan mencoba untuk menggunakan lensa 20D dan *smartphone* saja. Kondisi ruangan tetap gelap dengan menggunakan sumber cahaya *flash*, serta subjek menggunakan cairan *Cendo Mydiatil* 1%. Pengambilan citra dilakukan dengan keadaan jarak mata ke lensa ± 4 cm dan jarak lensa ke kamera dimulai dari ± 15 cm di jauhkan perlahan hingga ± 30 cm. Namun, pada metode ini fundus juga masih belum berhasil didapatkan. Gambar 4.22 menunjukkan hasil dari

pengambilan citra, sementara Gambar 4.23 menampilkan kondisi ruangan pengambilan data.

4.4.3.3 *Peneliti dengan direct oftalmoskop dan smartphone*

Peneliti dilanjutkan dengan mencoba untuk menggunakan *direct oftalmoskop* dan *smartphone*. Kondisi ruangan tetap gelap namun tanpa menggunakan bantuan *flash* dari *smartphone* sebab sumber cahaya sudah dihasilkan secara langsung melalui *direct oftalmoskop*. Pengambilan citra dilakukan dengan keadaan jarak mata ke alat $\pm 20\text{cm}$ serta subjek menggunakan cairan *Cendo Mydiatil* 1%. Saat dicoba, fundus berhasil terlihat melalui *direct oftalmoskop* dengan mata secara langsung. Namun, jika menggunakan kamera *smartphone*, lubang dari *direct oftalmoskop* terlalu kecil sehingga citra tidak dapat terlihat di dalamnya. Gambar 4.24 merupakan kondisi pengambilan citra fundus menggunakan *direct oftalmoskop*.

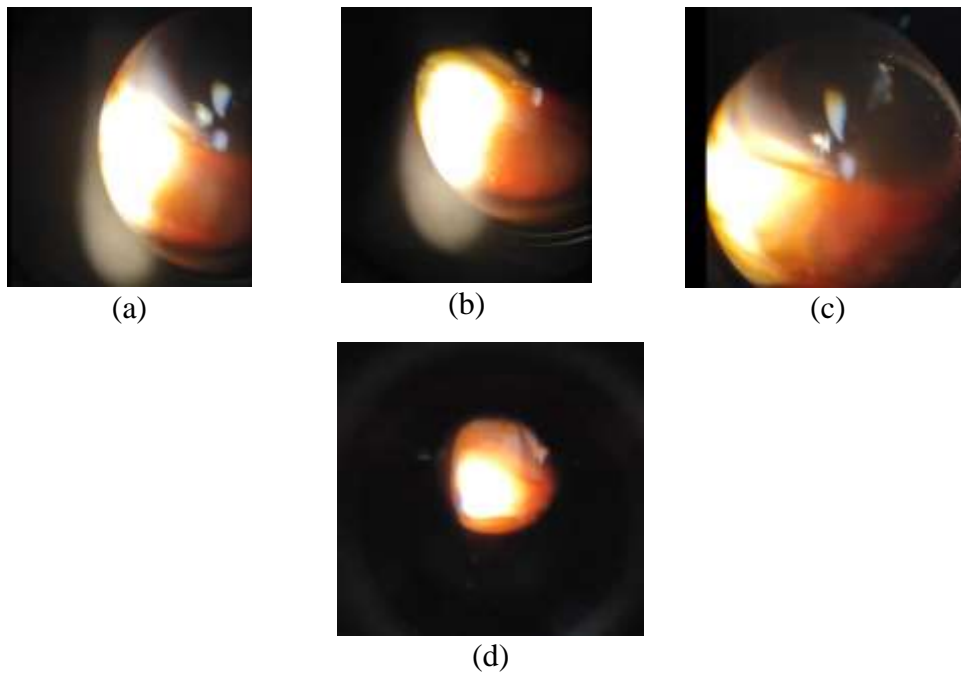


Gambar 4.24 Penangkapan citra fundus dengan *direct oftalmoskop* dan *smartphone*

4.4.3.4 *Peneliti dengan lensa 78D dan disertai smartphone*

Peneliti dilanjutkan dengan mencoba untuk menggunakan rangkaian alat medis meliputi *slit lamp* dan lensa 78D dan disertai *smartphone*. Kondisi ruangan terang dan tidak menggunakan bantuan *flash* dari *smartphone* sebab sumber cahaya sudah dihasilkan secara langsung melalui *slit lamp*. Pengambilan citra dilakukan dengan keadaan jarak mata ke alat $\pm 7\text{cm}$ serta subjek menggunakan cairan *Cendo Mydiatil* 1%. Saat dicoba, fundus berhasil terlihat melalui *slit*

lamp dengan mata secara langsung. Namun, jika menggunakan kamera *smartphone*, citra fundus sulit sekali ditangkap karena pergeseran sedikit saja antara mata, lensa 78D, *slit lamp*, dan kamera *smartphone* yang mengakibatkan ketidak sejajaran, dapat menyebabkan citra yang ditangkap tidak maksimal. Gambar 4.25 merupakan hasil pengambilan citra fundus sementara Gambar 4.26 merupakan kondisi pengambilan citra fundus menggunakan lensa 78D, *slit lamp*, dan kamera *smartphone*.



Gambar 4.25 Penangkapan citra fundus dengan lensa 78D, *slit lamp* dan *smartphone*



Gambar 4.26 Kondisi ruangan pengambilan citra fundus dengan lensa 78D, *slit lamp* dan *smartphone*

4.4.4 Pengujian Kondisi Optimal Penangkapan Citra Fundus

Dikarenakan kegagalan pada beberapa tahap, peneliti melanjutkan untuk menentukan kondisi optimal untuk mendapatkan citra fundus seperti yang telah dilakukan pada tahap 4.4.3.1. Variabel terkontrol meliputi kecerahan ruangan yang sudah diatur pada < 60 lux dan dengan penambahan *flash smartphone*. Pada tahap ini, peneliti menggunakan beberapa variabel dengan variasi nilai yang terdapat pada Tabel 4.16.

Tabel 4.16 Tabel variabel uji kondisi optimal penangkapan citra fundus

Variabel	Variasi Nilai
Posisi badan	Duduk, tidur
Jarak mata ke lensa	5 cm, 10 cm, 15 cm
Jarak lensa ke kamera <i>smartphone</i>	20 cm, 30 cm, 40 cm, 50 cm
Perbesaran kamera <i>smartphone</i>	1x, 2x, 5x, 10x

Berdasarkan penelitian yang dilakukan, hasil yang sempat berhasil mendapatkan citra terdapat pada Tabel 4.17.

Tabel 4.17 Tabel hasil uji kondisi optimal penangkapan citra fundus

Variabel	Variasi Nilai
Posisi badan	Duduk, tidur
Jarak mata ke lensa	± 7 hingga ± 12 cm
Jarak lensa ke kamera <i>smartphone</i>	± 25 cm hingga ± 30 cm
Perbesaran kamera <i>smartphone</i>	5x

Dengan demikian, untuk menangkap citra fundus secara optimal saat menggunakan lensa 20D, dapat digunakan kondisi seperti yang terdapat pada Tabel 4.17.

BAB V

Penutup

5.1 Kesimpulan

1. Arsitektur VGG-19 sederhana sudah dibuat dengan *dataset* dari <https://www.kaggle.com/datasets/pkdarabi/diagnosis-of-diabetic-retinopathy> untuk dua kelas yaitu DR dan No DR, serta menggunakan *learning rate* 0.0001, *epoch* 20, *step per epoch* 40, dan *validate step* 30. Arsitektur yang dibuat menghasilkan akurasi data latih 95%, *loss* data latih 15%, akurasi data tes 94%, *loss* data tes 18%, waktu *learning* 1 jam 27 menit 24 detik, ukuran *file* model mencapai 19.405 kb, serta keberhasilan simulasi prediksi mencapai 87,661% kebenaran.
2. Aplikasi android untuk mendeteksi diabetes retinopati berhasil dibuat dengan dua fungsi yaitu pendeteksian melalui gambar yang diimpor ke dalam model, serta gambar yang diambil secara langsung melalui kamera (gambar merupakan gambar cetak yang difoto kembali melalui aplikasi). Untuk fungsi pendeteksian gambar yang diimpor mendapatkan akurasi 70% benar menebak data No_DR dan 80% untuk data DR. Sementara untuk foto gambar cetak melalui kamera *smartphone* secara langsung mendapat hasil keberhasilan mendeteksi No_DR sebesar 91,67%, DR sebesar 75,00%, dan keberhasilan rata-rata sebesar 83,33%.
3. Prototipe kamera fundus yang mengadaptasi mekanisme oftalmoskop tidak langsung dengan memanfaatkan *smartphone*, oftalmoskop funduskopi adaptor dengan panjang 25 cm, dan lensa 20D Volk masih belum berhasil sebab masih gagal untuk mendapatkan citra fundus. Dengan demikian, pengujian aplikasi untuk mendeteksi citra fundus secara langsung ke subjek masih belum bisa dilakukan.
4. Penangkapan citra fundus dianjurkan untuk memenuhi kriteria ruangan dengan kecerahan kurang dari 60 lux, menggunakan pencahayaan *flash*, penambahan cairan *Cendo Mydiatil* 1% (dengan pengawasan tenaga

ahli), jarak lensa dengan mata $\pm 7 - 12$ cm, jarak lensa dengan *smartphone* 25 – 30 cm, serta menggunakan perbesaran 5x.

5.2 Saran

Untuk penelitian berikutnya bisa kembali dikembangkan untuk mendapat model yang lebih mutakhir sehingga mendapat akurasi yang tinggi dengan lebih konsisten. Penelitian berikutnya dapat memanfaatkan metode untuk ekstraksi citra untuk memperjelas vena, *cotton wool spots*, *mikroaneurisma*, eksedut keras, dan pendarahan. Peneliti juga bisa lebih mencermati terhadap kerusakan *dataset* sehingga tidak menyebabkan gagalnya proses *learning*. Untuk penerapan prototipe detektor melalui *smartphone*, pastikan untuk lebih memahami kondisi dan kemampuan untuk melakukan pendeteksian menggunakan alat oftalmoskop serta dianjurkan untuk merancang alat yang dapat mempermudah proses untuk menyejajarkan mata, lensa, dan kamera sehingga fitur foto (bukan video) untuk pendeteksian diabetes retinopati bisa digunakan lebih optimal.

DAFTAR PUSTAKA

- Acer. (2019). *Swift 3 Acer Day Edition, Si Tipis Responsif dengan Performa Tinggi*. Retrieved Januari 7, 2024, <https://www.acerid.com/laptop-swift-3-acer-day-edition-sf314-56g/>.
- Ademola-Popoola, D. S., & Olatunji, V. A. (2017). *Retinal Imaging with Smartphone*. Nigerian journal of clinical practice, 20(3), 341-345.
- Akbar, M. J., Sardjono, M. W., Cahyanti, M., & Swedia, E. R. (2020). *Perancangan Aplikasi Mobile untuk Klasifikasi Sayuran Menggunakan Deep Learning Convolution Neural Network*. Sebatik, 300-306.
- Afkariansyah, G. R. (2022). *Klasifikasi Fundus Diabetic Retinopathy menggunakan Deep Learning*. Doctoral dissertation, Universitas Dinamika, 1-43.
- Barnadi, Y. (2021). *Pengaruh Perubahan Dimensi Terhadap Nilai Isolasi pada Branch Line Coupler*. Jurnal Ilmiah Teknologi Infomasi Terapan, 8(1).
- Bolster, N. M., Giardini, M. E., & Bastawrous, A. (2016). *The Diabetic Retinopathy Screening Workflow: Potential for Smartphone Imaging*. Journal of diabetes science and technology, 10(2), 318-324.
- Chalam, K. V., Chamchikh, J., & Gasparian, S. (2022). *Optics and Utility of Low-Cost Smartphone-Based Portable Digital Fundus Camera System for Screening of Retinal Diseases*. Diagnostics, 12(6), 1499.
- Chen, W., Chang, J., Zhao, X., & Liu, S. (2021). *Optical Design and Fabrication of A Smartphone Fundus Camera*. Applied Optics, 60(5), 1420-1427.
- Dhanorkar, A. (2021, September 3). *What Is the Purpose of Fundus Photography*. Retrieved Oktober 5, 2023, [medicinenet.com: https://www.medicinenet.com/what-is-the-purpose-of-fundus-photography/article.htm](https://www.medicinenet.com/what-is-the-purpose-of-fundus-photography/article.htm)

- Fuadah, Y. N., Saidah, S., Khofiya, N., Magdalena, R., & Ubaidullah, I. D. (2022). *Glaucoma Classification Based on Fundus Images Processing with Convolution Neural Network*. Jurnal Teknik Informatika, 3(3), 717-722.
- Fredonovalle, M. A. (2022). *Implementasi Convolution Neural Network dalam Pengenalan Citra untuk Pengembangan Game Bahasa Arab: Pembelajaran Kitabah (Menulis) Berbasis Android*. Jakarta: Fakultas Sains dan Teknologi UIN Syarif Hidayatullah Jakarta.
- Goodfellow, I., Bengio, Y., & Courville, A. (2016). *Deep Learning*. MIT Press
- Haddock, L. J., Kim, D. Y., & Mukai, S. (2013). *Simple, Inexpensive Technique for High-Quality Smartphone Fundus Photography in Human and Animal Eyes*. Journal of Ophthalmology, 2013.
- Hansun, S., Kristanda, M. B., & Saputra, M. W. (2018). *Pemrograman Android Dengan Android Studio IDE*. Universitas Muhammadiyah Jakarta.
- He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 770-778).
- Holiyanti, R., Sukmawati, Fahmi, I., & Rozikin, C. (2022). *Pendeteksi Sampah Metal untuk Daur Ulang Menggunakan Metode Convolutional Neural Network*. Jurnal Teknik Informatika dan Sistem Informasi, 8(1), 261-274.
- Hu, Y., Huber, A., Anumula, J., & Liu, S. C. (2018). *Overcoming the vanishing gradient problem in plain recurrent networks*. arXiv preprint arXiv:1801.06105.
- Iqbal, U. (2021). *Smartphone Fundus Photography: A Narrative Review*. International Journal of Retina and Vitreous, 7(1), 44.
- Lord, R. K., Shah V. A., San Filippo, A. N., Krishna, R. (2010). *Novel Uses of Smartphones in Ophthalmology*. Ophthalmology. 117(6).

- Mahdianpari, M., Salehi, B., Rezaee, M., Mohammadimanesh, F., & Zhang, Y. (2018). *Very Deep Convolutional Neural Networks for Complex Land Cover Mapping Using Multispectral Remote Sensing Imagery*. *Remote Sensing*, 1-21.
- Marcella, D., Yohannes, & Devella, S. (2022). *Klasifikasi Penyakit Mata Menggunakan Convolutional Neural Network dengan Arsitektur VGG-19*. *Jurnal Algoritme*, 3(1), 60-70.
- Mishra, C., & Koushik, T. (2023, Juli 31). *Fundus Camera*. Retrieved Oktober 5, 2023, from ncbi.nlm.nih.gov: <https://www.ncbi.nlm.nih.gov/books/NBK585111>
- Mutawalli, L., Zaen, M. T. A., & Yuliadi, Y. (2023). Komparasi CNN dengan ResNet Untuk Klasifikasi Paling Akurat Tingkat Keganasan Diabetes Berdasarkan Citra Retinopathy. *Journal of Computer System and Informatics (JoSYC)*, 4(3), 522-529.
- P2PTM Kemenkes RI. (11 Desember 2018). *Diabetes Penderita di Indonesia Bisa Mencapai 30 Juta Orang pada Tahun 2030*. P2PTM, Kemenkes RI.. Disadur pada 5 September 2023 <https://p2ptm.kemkes.go.id/tag/diabetes-penderita-di-indonesia-bisa-mencapai-30-juta-orang-pada-tahun-2030>.
- Pradeep. (2023, Mei 23). *What is the difference between VGG16 and VGG19 neural network?* Retrieved Oktober 13, 2023, from www.quora.com: <https://www.quora.com/What-is-the-difference-between-VGG16-and-VGG19-neural-network>
- Pratama, A. P., Atmaja, R. D., & Fauzi, H. (2016). *Deteksi Diabetes Retinopati pada Foto Fundus Menggunakan Color Histogram & Transformasi Wavelet*. *eProceedings of Engineering*, 3(3), 4552 - 4559.
- Primartha, R. (2018). *Belajar Machine Learning – Teori dan Praktik*. Palembang. Informatika Bandung.

- Putra, I. B. N. W. A., Darmawan, I. D. M., & Wirawan, I. K. A. (2019). *Penerapan Teknik Pencahayaan Low Key Dan High Contrast Dalam Film “Niskala”*. Karya Ilmiah ISI Denpasar
- Rikendry, R., & Maharil, A. (2022). Perbandingan Arsitektur Vgg16 Dan Resnet50 Untuk Rekognisi Tulisan Tangan Aksara Lampung. *Jurnal Informatika dan Rekayasa Perangkat Lunak*, 3(2), 236-243.
- Roth, S., & Stahl, A. (2019). *Fotometrie. Optik: Experimentalphysik—anschaulich erklärt*, 213-236.
- Sabrina, E., & Buditjahjanto, I. G. (2017). *Klasifikasi Penyakit Diabetic Retinopathy menggunakan Metode Learning Vector Quantization (LVQ)*. *Jurnal Teknik Elektro*, 6(2), 97-104.
- Sasongko, M. B., Wardhana, F. S., Febryanto, G. A., Agni, A. N., Supanji, S., Indrayanti, S. R., *et al.* (2020). *The Estimated Healthcare Cost of Diabetic Retinopathy in Indonesia and Its Projection for 2025*. *British Journal of Ophthalmology*, 104(4), 487-492.
- Setiawan, W. (2019). *Perbandingan Arsitektur Convolutional Neural Network untuk Klasifikasi Fundus*. *Simantec*, 7(2), 49-54.
- SNI 6197. (2011). SNI 6197: 2011 Konservasi Energi pada Sistem Pencahayaan. Standar Nasional Indonesia, 1–38.
- Subarkah, P. (2020). *Penerapan Algoritma Klasifikasi Classification And Regression Trees (CART) untuk Diagnosis Penyakit Diabetes Retinopathy*. *Jurnal MATRIK*, 19(2), 294-301.
- Sze, V., Chen, Y. H., Yang, T. J., & Emer, J. S. (2020). *Efficient processing of deep neural networks*. San Rafael: Morgan & Claypool Publishers.
- Tantono, C. C. (2024). *Pengaruh Regularisasi pada Peningkatan Akurasi Deep Learning untuk Deteksi Penyakit Diabetes Retinopati Menggunakan Citra Fundus Retina Mata*. (Skripsi Sarjana, Universitas Ma Chung).

- Vitriana, A. (2023). *Aplikasi Artificial Intelligence (AI) dalam Skrining Diabetik Retinopati*. Oftalmologi Komunitas, 1-9.
- Wang, L. Z., Cheung, C. Y., Tapp, R. J., Hamzah, H., Tan, G., Ting, D., *et al.* (2017). *Availability and Variability in Guidelines on Diabetic Retinopathy Screening in Asian Countries*. British Journal of Ophthalmology, 101(10), 1352-1360.
- William, & Lubis, C. (2022). *Klasifikasi Penyakit Mata Menggunakan CNN*. Jurnal Ilmu Komputer dan Sistem Informasi, 10(1), 1-4.
- Wibowo, K. A., & Alamsyah. (2023). *Klasifikasi Diabetic Retinopathy Menggunakan CNN dengan Arsitektur yang Dimodifikasi*. Indonesian Journal of Mathematics and Natural Sciences, 46(1), 45-52.
- Zhou, B., Lu, Y., Hajifathalian, K., Bentham, J., Di Cesare, M., Danaei, *et al.* (2016). *Worldwide Trends in Diabetes Since 1980: A Pooled Analysis of 751 Population-Based Studies with 4.4 Million Participants*. The lancet, 387(10027), 1513-1530.

LAMPIRAN

Lampiran 1 *Crop No DR*

```
1 def crop_image_from_gray(img,tol=7):
2     if img.ndim ==2:
3         mask = img>tol
4         return img[np.ix_(mask.any(1),mask.any(0))]
5     elif img.ndim==3:
6         gray_img = cv2.cvtColor(img, cv2.COLOR_RGB2GRAY)
7         mask = gray_img>tol
8
9         check_shape = img[:, :,0][np.ix_(mask.any(1),mask.any(0))].shape[0]
10        if (check_shape == 0): # image is too dark so that we crop out everything,
11            return img # return original image
12        else:
13            img1=img[:, :,0][np.ix_(mask.any(1),mask.any(0))]
14            img2=img[:, :,1][np.ix_(mask.any(1),mask.any(0))]
15            img3=img[:, :,2][np.ix_(mask.any(1),mask.any(0))]
16            img = np.stack([img1,img2,img3],axis=-1)
17        return img
18
19 def copy_crop_nodr(img):
20     # Open Image
21     im = Image.open(img)
22
23     # Convert to array so can be cropped by function
24     numpy_array = np.array(im)
25
26     # Crop content
27     im = crop_image_from_gray(numpy_array)
28
29     # Convert array (numpy.ndarray) to image so can be saved
30     data = Image.fromarray(im)
```

Lampiran 2 Kode untuk Menggandakan Data *No DR*

```
1 def generate_data_moderate(image, directory, namefile_len):
2     # Open Image
3     img = Image.open(image)
4
5     # 1. SAVE THE CROPED ORIGINAL
6     # Convert to array so can be cropped by function
7     numpy_array = np.array(img)
8     # Crop content
9     im = crop_image_from_gray(numpy_array)
10    # Convert array (numpy.ndarray) to image so can be saved
11    data = Image.fromarray(im)
12    # save file
13    #save(data, directory, namefile_len)
14    os.chdir(directory)
15    filename = image[namefile_len:-5] + '.png'
16    data.save(filename)
17
18    # 2. SAVE THE MIRROR
19    mirror(img, directory, (image[namefile_len:-5] + '_m.png'))
20
21    # 3. SAVE THE ROTATE 30
22    rotate_image(img, 30, directory, (image[namefile_len:-5] + '_r30.png'))
23
24    # 4. SAVE THE MIRROR - ROTATE 30
25    rot_mir(img, 30, directory, (image[namefile_len:-5] + '_mr30.png'))
26
27    # 5. SAVE THE ROTATE 60
28    rotate_image(img, 60, directory, (image[namefile_len:-5] + '_r60.png'))
29
30    # 6. SAVE THE MIRROR - ROTATE 60
31    rot_mir(img, 60, directory, (image[namefile_len:-5] + '_mr60.png'))
32
33    # 7. SAVE THE ROTATE 90
34    rotate_image(img, 90, directory, (image[namefile_len:-5] + '_r90.png'))
35
36    # 8. SAVE THE MIRROR - ROTATE 90
37    rot_mir(img, 90, directory, (image[namefile_len:-5] + '_mr90.png'))
38
39    # 9. SAVE THE ROTATE 330
40    rotate_image(img, 330, directory, (image[namefile_len:-5] + '_r330.png'))
41
42    # 10. SAVE THE MIRROR - ROTATE 330
43    rot_mir(img, 330, directory, (image[namefile_len:-5] + '_mr330.png'))
44
45    # 11. SAVE THE ROTATE 300
46    rotate_image(img, 300, directory, (image[namefile_len:-5] + '_r300.png'))
47
48    # 12. SAVE THE MIRROR - ROTATE 300
49    rot_mir(img, 300, directory, (image[namefile_len:-5] + '_mr300.png'))
50
51
52    # 13. SAVE THE ROTATE 270
53    rotate_image(img, 270, directory, (image[namefile_len:-5] + '_r270.png'))
54
55    # 14. SAVE THE MIRROR - ROTATE 270
56    rot_mir(img, 270, directory, (image[namefile_len:-5] + '_mr270.png'))
57
58
59    print(image[namefile_len:-5] + ' finish')
```