# PENERAPAN MODEL VGG-19 DAN *RESNET*-50 DALAM APLIKASI ANDROID DAN PURWARUPA DETEKTOR DIABETES RETINOPATI

# **TUGAS AKHIR**



# YOHANES VIANNEY DANI 312010013

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

#### **KATA PENGANTAR**

Puji dan syukur dipanjatkan 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 VGG-19 dan *ResNet-50* dalam Aplikasi Android dan Purwarupa Detektor Diabetik Retinopati".

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

- 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 sebagai asisten saat proses pengambilan sampel di lapangan selama mengerjakan 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 serta penelitain ini dapat terus dikembangkan. Demikian yang dapat disampaikan, penulis mengucapkan terima kasih.

Malang, 20 Juli 2023

Yohanes Vianney Dani

# **DAFTAR ISI**

KATA PENGANTAR	i
DAFTAR ISI	iii
DAFTAR GAMBAR	vi
DAFTAR TABEL	viii
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	5
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 Deep Learning	9
2.5 Convolution Neural Network	10
2.5.1 Convolution Layer	10
2.5.2 Pooling Layer	11
2.5.3 VGG	12
2.5.4 Residual Network (ResNet)	12
2.6 Pencahayaan	14
2.7 Penelitian Terdahulu	15
BAB III Metode Penelitian	19
3.1 Alur Penelitian	19
3.2 Analisis Masalah	20
3.3 Studi Literatur	20
3.4 Perancangan Aplikasi	21

3.4.1 Perancangan Arsitektur CNN	21
3.4.2 Perancangan Aplikasi	23
3.5 Integrasi Perangkat Lunak dan Perangkat Keras	24
3.6 Pengujian	26
3.6.1 Pengujian Fitur Deteksi dari Citra Gallery ke Aplikasi	26
3.6.2 Pengujian Fitur Deteksi dari Citra Foto Langsung	27
3.6.3 Pengujian Hasil Deteksi Berdasarkan Parameter Uji	28
BAB IV Hasil dan Pembahasan	29
4.1 Penelitian Arsitektur Model	29
4.1.1 Data	30
4.1.1.1 Data 1	32
4.1.1.2 Data 2	32
4.1.1.3 Data 3	33
4.1.1.4 Data 4	34
4.1.2 Arsitektur	35
4.1.2.1 Arsitektur VGG-19	35
4.1.2.2 Arsitektur ResNet-50	36
4.1.2.3 Arsitektur VGG-19 sederhana 1	39
4.1.2.4 Arsitektur VGG-19 sederhana 2	40
4.2 Hasil Percobaan Model	41
4.3 Pembuatan Aplikasi Android	45
4.4 Pengujian Deteksi dengan Aplikasi Android	51
4.4.1 Pengujian Deteksi Menggunakan Citra dari Gallery	51
4.4.2 Pengujian Deteksi dengan Kamera pada Citra Gambar	
Cetak	58
4.4.3 Pengujuan 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 smartphone tanpa	
adaptor	64
4.4.3.3 Peneliti dengan direct oftalmoskop dan smartphone	265
4.4.3.4 Peneliti dengan lensa 78D dan disertai smartphone	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	ix

# **DAFTAR GAMBAR**

Gambar 2.1 Perbandingan citra retina normal dan retina pengidap diabetes	
retinopati	6
Gambar 2.2 Pembagian kelas diabetes retinopati	7
Gambar 2.3 Oftalmoskop funduskopi	8
Gambar 2.4 Ilustrasi proses konvolusi	11
Gambar 2.5 Ilustrasi konsep pooling layer	11
Gambar 2.6 Arsitektur VGG	12
Gambar 2.7 Blok residual fondasi arsitektor ResNet	13
Gambar 2.8 Arsitektur <i>ResNet-50</i>	13
Gambar 2.9 Lux meter	14
Gambar 3.1 Alur penelitian	19
Gambar 3.2 Layer dalam arsitektur VGG-19	22
Gambar 3.3 Alur pembuatan aplikasi android	23
Gambar 3.4 Rancangan interface	23
Gambar 3.5 Oftalmoskop funduskopi adaptor smartphone	25
Gambar 3.6 Oftalmoskop funduskopi yang terintegrasi dengan <i>smartphone</i>	26
Gambar 4.1 Peringatan batas ukuran maksimal model pada Android Studio	29
Gambar 4.2 Perbandingan retina No DR dengan keempat kelas lainnya	32
Gambar 4.3 Contoh citra yang dianggap rusak	33
Gambar 4.4 Contoh citra <i>Proliferative DR</i> yang digandakan	34
Gambar 4.5 Arsitektur VGG-19	35
Gambar 4.6 Arsitektur <i>ResNet-50</i> bagian 1	37
Gambar 4.7 Arsitektur <i>ResNet-50</i> bagian 2	37
Gambar 4.8 Arsitektur <i>ResNet-50</i> bagian 3	38
Gambar 4.9 Arsitektur <i>ResNet-50</i> bagian 4	38
Gambar 4.10 Arsitektur <i>ResNet-50</i> bagian 5	39
Gambar 4.11 Arsitektur VGG-19 sederhana 1	40
Gambar 4.12 Arsitektur VGG-19 sederhana 2	40
Gambar 4.13 Convolution matrix dari model yang sudah dibuat	42
Gambar 4.14 Tampilan menu utama dari aplikasi android	46

Gambar 4.15 Potongan kode fungsi interaksi tombol dan tampilan	46
Gambar 4.16 Potongan kode fungsi onActivityResult	47
Gambar 4.17 Potongan kode fungsi prediksi dan pemanggilan model	48
Gambar 4.18 Tampilan menu utama aplikasi	50
Gambar 4.19 Tampilan aplikasi tombol "Select Image"	50
Gambar 4.20 Tampilan aplikasi tombol "Capture Image"	50
Gambar 4.21 Tampilan aplikasi tombol "Prediksi"	51
Gambar 4.22. Penerapan perangkat keras <i>smartphone</i> dan lensa 20D	62
Gambar 4.23 Penangkapan citra fundus dengan lensa 20D, adaptor, dan	
smartphone	63
Gambar 4.24 Penangkapan citra fundus dengan lensa 20D dan smartphone	64
Gambar 4.25 Kondisi ruangan pengambilan data sampel	64
Gambar 4.26 Penangkapan citra fundus dengan direct oftalmoskop dan	
smartphone	65
Gambar 4.27 Penangkapan citra fundus dengan lensa 78D, slit lamp dan	
smartphone	66
Gambar 4.28 Kondisi ruangan pengambilan citra fundus dengan lensa 78D, slit	ţ
lamp dan smartphone	66

# **DAFTAR TABEL**

Tabel 2.1 Tabel perbandingan besarnya lux dan gambaran kecerahan ruangan	15
Tabel 3.1 Tabel spesifikasi smartphone	24
Tabel 3.2 Tabel spesifikasi oftalmoskop funduskopi adaptor smartphone	25
Tabel 3.3 Tabel spesifikasi lux meter	28
Tabel 3.4 Tabel pengujian parameter terhadap hasil pendeteksian	28
Tabel 4.1 Perbedaan percobaan pembuatan model CNN	30
Tabel 4.2 Pelabelan jenis data dan perbedaannya	31
Tabel 4.3 Dataset 1	32
Tabel 4.4 Dataset 1 yang sudah imoifikasi untuk kelas No DR dan Proliferativ	'e34
Tabel 4.5 Perbedaan hasil percobaan pembuatan model CNN	41
Tabel 4.6 Perbandingan hasil prediksi model CNN	43
Tabel 4.7 Rata-rata prediksi benar dari model yang dipilih	45
Tabel 4.8 Hasil deteksi citra No_DR dari memori handphone	51
Tabel 4.9 Hasil deteksi citra DR dari memori handphone	55
Tabel 4.10 Hasil deteksi citra <i>No_DR</i> dari kamera pada gambar cetak	59
Tabel 4.11 Hasil deteksi citra <i>DR</i> dari kamera pada gambar cetak	60
Tabel 4.12 Variabel uji kondisi optimal penangkapan citra fundus	67
Tabel 4.13 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 diabetic retinopathy (DR). Diabetic retinopathy suatu kondisi yang memengaruhi pembuluh darah di retina dan dapat menyebabkan masalah penglihatan serta berpotensi mengakibatkan kebutaan. Kondisi tersebut menjadikan diabetic retinopathy 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).

Diabetic retinopathy (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 diabetic retinopathy. 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.

Berdasarkan paparan di atas, dapat diketahui bahwa terdapat dua kendala utama dalam menghadirkan aplikasi diagnosis *diabetic retinopathy* berbasis *smartphone*. Yang pertama adalah optimasi *preprocessing*, ekstraksi fitur, dan

hyperparameter tuning. Sementara yang kedua adalah pengujian untuk integrasi penggunaan optimized model. 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

Identifikasi masalah pada penelitian ini adalah belum hadirnya aplikasi *mobile* terkait diagnosis *diabetic retinopathy* serta kurang terujinya parameter pencahayaan dan penambahan *flash* saat pengambilan citra secara langsung dalam pendeteksian dan bagaimana upaya dalam mengoptimalkan pengkondisian parameter tersebut.

#### 1.3 Batasan Masalah

Batasan masalah yang ditentukan dalam proyek Tugas Akhir ini sebagai berikut

- 1. Penelitian terfokus pada rancangan purwarupa, pengintegrasian purwarupa, pembuatan aplikasi *mobile*, dan pengujian parameter pencahayaan dan penambahan *flash* pada purwarupa;
- 2. Aplikasi akan diintegrasikan pada *smartphone* android;
- 3. Pendeteksian dilakukan pada retina mata dan fungsinya spesifik untuk melakukan diagnosis terkait *diabetic retinopathy*.

#### 1.4 Perumusan Masalah

Pada penelitian ini ditemukan dua buah masalah yaitu sebagai berikut:

- 1. Bagaimana proses desain dari aplikasi *smartphone* yang dibuat?
- 2. Bagaimana rancangan dan perwujudan purwarupa kamera fundus yang terintegrasi dengan *smartphone* yang digunakan untuk mendeteksi *diabetic retinopathy*?
- 3. Bagaimana konfigurasi dalam pengoptimalan kamera *smartphone* untuk mendeteksi lebih akurat terkait *diabetic retinopathy*?

4. Bagaimana batasan keadaan optimal untuk mendeteksi *diabetic retinopathy* pada retina mata berdasarkan parameter pencahayaan dan penambahan *flash*?

# 1.5 Tujuan

Tujuan dari penelitian ini adalah sebagai berikut:

- 1. Mengetahui desain dari aplikasi *smartphone* yang dibuat;
- 2. Mengetahui rancangan dan perwujudan purwarupa kamera fundus yang terintegrasi dengan *smartphone* yang digunakan untuk mendeteksi *diabetic retinopathy*;
- 3. Mengetahui konfigurasi dalam pengoptimalan kamera *smartphone* untuk mendeteksi lebih akurat terkait *diabetic retinopathy*;
- 4. Mengetahui batasan keadaan optimal untuk mendeteksi *diabetic retinopathy* pada retina mata berdasarkan parameter pencahayaan dan penambahan *flash*.

#### 1.6 Manfaat

Manfaat yang diperoleh dari tugas akhir ini adalah sebagai berikut:

1. Bagi mahasiswa

Dengan adanya proyek ini, mahasiswa dapat menerapkan ilmu yang diperoleh 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 *diabetic retinopathy* 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 Tugas Akhir, dan sistematika penulisan.

# BAB II Tinjauan Pustaka

Bab dua ini terdiri dari uraian mengenai teori dari *diabetic retinopathy*, 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, perangcangan 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

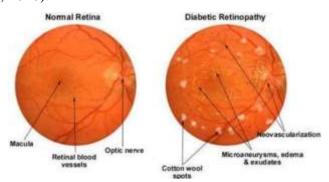
Bab lima ini berisi simpulan dari hasil proyek Tugas Akhir yang telah dikerjakan dan saran terhadap hal-hal yang dapat diperbaiki dari purwarupa maupun aplikasi untuk penelitian berikutnya.

#### **BAB II**

# Tinjauan Pustaka

#### 2.1 Diabetes Retinopati

Dikutip dari penjelasan Sabrina dan Buditjahjanto (2017) serta Pratama, Atmaja, dan Fauzi (2016), diabetes retinopati (DR) adalah salah satu komplikasi penyakit diabetes melitus. Komplikasi tersebut diakibatkan tingginya kadar glukosa 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). Pada umumnya diabetes retinopati berdampak secara langsung pada terganggunya penglihatan penderita dan bila terlambat ditangani maka akan menyebabkan penderita mengalami 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 yang jelas pada tahap awal (Subarkah, 2020).



Gambar 2.1 Perbandingan citra retina normal dan retina pengidap diabetes retinopati

Sumber: Pratama, Atmaja, dan Fauzi, 2016

Efek visual dari kebocoran pembuluh darah digambarkan dengan adanya perubahan pada retina seperti perubahan *venous*, *cotton wool spots*, *mikroaneurisma*, *hard exedute*, dan pendarahan (Subarkah, 2020). Diperjelas kembali oleh Sabrina dan Buditjahjanto (2017) bahwa tanda-tanda yang

ditunjukkan oleh penderita diabetes retinopati (DR) antara lain *mikroaneurisma*, pendarahan dalam rongga *vitreous* (*hemorrhages*), *hard* dan *soft exudate* (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 dilihat pada Gambar 2.2.



Gambar 2.2 Pembagian kelas diabetes retinopati

Sumber: Vitriana, 2023

#### 2.2 Kamera Fundus

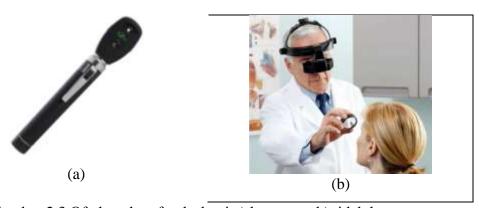
Kamera fundus atau oftalmoskop adalah alat yang dipergunakan untuk mengambil foto bagian dasar mata atau dikenal dengan nama fundus (Mishra & Koushik, 2023). Sementara pengertian tambahan menurut Dhanorkar (2021), Kamera fundus adalah mikroskop khusus berdaya rendah yang dipasang pada kamera yang digunakan untuk memeriksa struktur seperti cakram optik, retina, dan lensa. 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 langsung dan tidak langsung, dapat dilihat pada Gambar 2.3.

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

- 1. Hanya membutuhkan waktu satu sampai dua menit untuk proses pendapatan gambar;
- 2. Lebih mudah memvisualisasikan detail retina melalui foto fundus stereoskopis dibandingkan memeriksa mata secara langsung;
- 3. Sering kali beberapa bagian mata terlihat lebih detail dalam fotografi fundus dibandingkan pemeriksaan mata lainnya seperti angiografi, dan *fluorescein*;
- 4. 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):

- Warna oranye atau bulan sabit cerah di tepi foto disebabkan oleh iris yang menyala, khususnya pada pupil yang melebar dengan buruk;
- 2. 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;
- 3. 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;



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

4. 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 adalah *Integrated Development Environment* (IDE) resmi untuk membangun aplikasi android, yang berbasiskan dari IntelliJ IDEA. Selain menyediakan editor kode dan fitur pengembang yang handal dari IntelliJ, Android Studio juga menawarkan berbagai fitur tambahan yang dapat meningkatkan efisiensi dalam pembuatan aplikasi android.

Android merupakan salah satu sistem operasi yang populer di kalangan masyarakat karena bersifat *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 Fredonovalle (2022) bahwa deep learning merupakan bagian dari machine learning yang pada dasarnya adalah jaringan saraf dengan tiga atau lebih lapisan. Jaringan saraf ini mencoba untuk mensimulasikan perilaku otak manusia yang memungkinkannya untuk mempelajari sejumlah besar data. Jaringan saraf dengan satu lapisan masih mampu membuat perkiraan-perkiraan, namun dengan banyaknya lapisan tersembunyi di bawahnya mampu mengoptimalkan dan menyempurnakan akurasi.

Deep learning telah mendorong perkembangan berbagai aplikasi dan layanan dalam bidang kecerdasan buatan (AI). Deep learning memungkinkan otomatisasi tugas-tugas analitis dan fisik tanpa perlu campur tangan manusia. Teknologi deep

*learning* berperan penting dalam menggerakkan produk dan layanan sehari-hari, seperti asisten digital, kontrol TV melalui suara, dan deteksi penipuan kartu kredit, serta berkontribusi pada teknologi yang sedang dalam tahap perkembangan, 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).

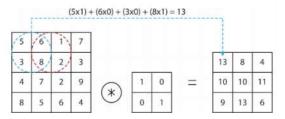
CNN mampu mengidentifikasi objek karena memiliki dua bagian utama dalam proses pembelajarannya, yaitu mengekstraksi ciri khusus dari citra (*feature extraction*) dan melakukan klasifikasi berdasarkan ciri-ciri tersebut (*classification*) (Fuadah, Saidah, Khofiya, Magdalena, & Ubaidullah, 2022). Proses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

#### 2.5.1 Convolution Layer

Convolution layer merupakan lapisan utama yang paling penting digunakan dalam CNN. Dalam convolution layer akan dilakukan proses konvolusi antara citra dengan suatu filter berupa matriks dengan ukuran tertentu untuk menghasilkan feature map seperti yang ditunjukkan pada Gambar 2.4 (Fuadah, Saidah, Khofiya, Magdalena, & Ubaidullah, 2022). Konvolusi tersebut dilakukan menggunakan operasi aljabar linear yang mengalikan matriks dari filter pada citra yang akan diproses (Akbar, Sardjono, Cahyanti, & Swedia, 2020).

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.

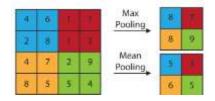


Gambar 2.4 Ilustrasi proses konvolusi

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

# 2.5.2 Pooling Layer

Pooling layer berfungsi untuk mereduksi input secara spasial (mengurangi jumlah parameter atau mengurangi 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

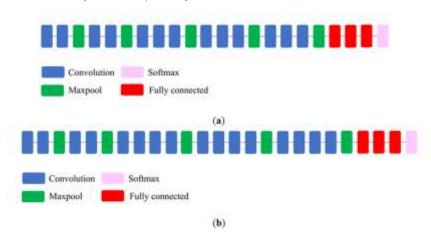
Sumber: Fuadah, Saidah, Khofiya, Magdalena, & Ubaidullah, 2022

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

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*. 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).



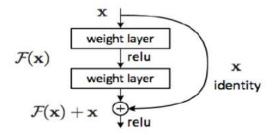
Gambar 2.6 a) Arsitektur VGG-16 dan b) Arsitektur VGG-19 Sumber: Mahdianpari, Salehi, Rezaee, Mohammadimanesh, & Zhang, 2018

# 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* yaitu suatu keadaan di mana hasil gradien yang dipelajari oleh model tidak

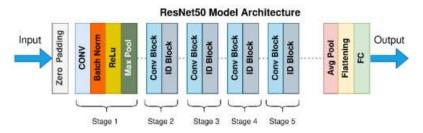
dapat mencapai layer pertama karena mengalami perkalian berkali-kali atau singkatnya CNN tidak dapat belajar dari error yang telah dikalkulasi (Hu, Huber, Anumula, & Liu, Overcoming the Vanishing Gradient Problem in Plain Recurrent Networks, 2018).

Teori tersebut mendukung kekuatan dari kegunaan arsitektur ResNet. Pada tahun 2016, He, Zhang, Ren, dan Sun (2016) menggunakan sesuatu yang bernama *residual block*. Blok ini adalah blok yang ada pada tiap lapis dan menjadi fundamental dari arsitektur CNN *Resnet*. Konsep dari *residual block* dapat dilihat pada Gambar 2.7.



Gambar 2.7 Blok residual fondasi arsitektor *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 yang terdapat dalam sebagian dari spektrum elektromagnetik (Roth & Stahl, 2019). Cahaya memiliki panjang gelombang khusus sehingga membedakannya dari 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).

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.



Gambar 2.9 Lux meter

Rumus 2 – 1 merupakan rumus untuk menghitung besarnya lux:

total lux (lux) = 
$$\frac{\text{total lumen (lumen)}}{\text{luas permukaan (m}^2)}$$
 (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 diabetic retinopathy menggunakan CNN pada citra fundus bukanlah penelitian pertama. Sudah terdapat beberapa penelitian terkait dengan berbagai metode yang berbeda untuk menghasilkan pendeteksi diabetic retinopathy menggunakan citra fundus. Pemilihan algoritma VGG sebagai model pendeteksi mendapat referensi dari jurnal milik Setiawan (2019) dengan judul "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 SqeezeNet. 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%, sensivitas 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 mengklasifiaksi 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, diabetik 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 diabetik retinopati merupakan penyakit yang paling sulit diklasifikasi. Berdasarkan laporan penelitian, ada kemungkingan kesulitan tersebut akibat sedikitnya *dataset* pada kelas diabetik 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 diabetik 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 *poliferative 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. Dengan kecepatan komputasi yang juga berada di urutan kedua, VGG-19 masih layak dijadikan sebagai model.

Penelitian selanjutnya telah menggunakan model VGG-16 dan spesifik untuk mendeteksi diabetik 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 diabetik retinopati. Hasil evaluasi

menunjukkan klasifikasi diabetik retinopati menggunakan CNN berarsitektur VGG-16 menghasilkan akurasi 96,24%.

Keempat penelitian tersebut merupakan landasan yang masih berkaitan dengan proses pendeteksian *diabetic retinopathy* menggunakan citra fundus. Namun, berdasarkan ketiga 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 dadn menggunakan sumber cahaya dari senter. Pada jurnal ini dipaparkan juga secara lebih jelas disain dan komparasi dari ebebrapa 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 *diabetic retinopathy* 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 akan diimplementasikan dan diuji keberhasilannya. Setelah implementasi perangkat lunak, 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 diabetic retinopathy yang sangat berbahaya sehingga penting dilakukan pengecekan secara rutin. Namun, hal tersebut masih terbatas akibat biaya pengecekan yang mahal dan hasil diagnosis yang tidak sebentar. Maka dari itu perlu dilakukan sebuah upaya untuk menangani keterbatasan kurang praktis dan efisiensi dari proses diagnosis konvensional dengan membuat sebuah aplikasi yang sederhana dan mampu mendeteksi secara cepat dan memiliki akurasi yang cukup baik.

Maka dari itu, nantinya akan dibuat sebuah aplikasi *smartphone* untuk mendeteksi penyakit *diabetic retinopathy* lebih dini. Aplikasi yang dibuat akan diintegrasikan dengan *smartphone* android yang telah dilengkapi kamera fundus sehingga kamera bisa mengambil data gambar retina mata pasien sebagai materi yang akan dideteksi.

## 3.3 Studi Literatur

Studi literatur dilakukan 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 Aplikasi

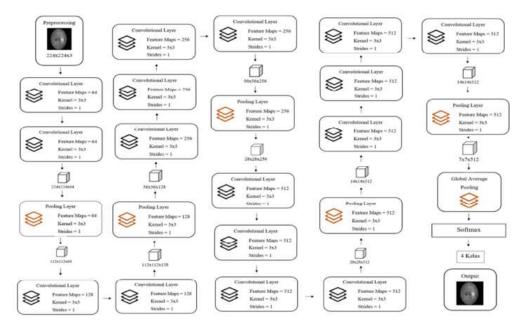
Dalam proyek ini, proses perancangan aplikasi dibagi menjadi dua. Pertama, pemilihan dan pengujian arsitektur *deep learning* yang sudah ada untuk digunakan kembali dalam proyek kali ini. Arsitektur *deep learning* yang ditemukan dan akan diimplementasikan harus diuji terlebih dahulu sehingga arsitektur yang dipilih memang dapat melakukan pendeteksian dengan baik. Kedua adalah proses utama dalam Tugas Akhir ini yaitu membuat aplikasi *smartphone* untuk mendeteksi *diabetic retinopathy*.

# 3.4.1 Perancangan Arsitektur CNN

Sistem untuk mendeteksi retina penderita *diabetic retinopathy* menggunakan arsitektur *Convolution Neural Network* (CNN). Arsitektur yang dipilih menggunakan algoritma VGG-19 dan *ResNet-50*. VGG-19 merupakan model dengan 19 lapisan bobot yang terdiri dari 16 *convolution layer* dan 3 *fully connected layer* (Pradeep, 2023). Gambar 3.2 merupakan lapisan dalam arsitektur VGG-19.

Filter pada VGG-19 menggunakan convolution layer dengan kernel 3x3 dan stride 1, serta pooling layer dengan 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*. *Dataset* yang digunakan merupakan *dataset* citra fundus yang telah dibedakan menjadi normal dan diabetik. Citra tersebut berjumlah total 25.563 citra. Data tersebut akan dibagi menjadi 80% data *train*, dan 20% data *test*. Model akan dilatih dengan *dataset* dan hasilnya akan dilihat akurasinya melalui *convolution matrix*.

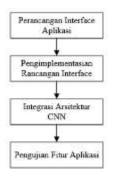


Gambar 3.2 Layer dalam arsitektur VGG-19

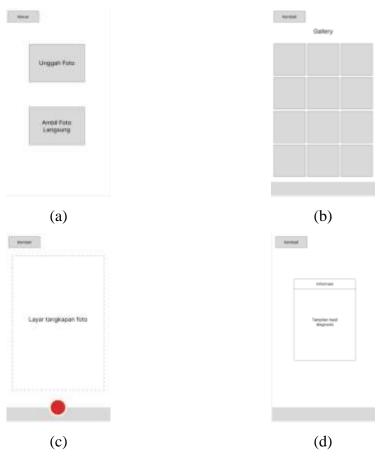
Sumber: Marcella, Yohannes, & Devella, 2022

Rancangan untuk *ResNet-50* akan memanfaatkan proses melalui lima blok *convolution*. Blok pertama akan terdapat satu *convolution 2D layer*, satu *layer* normalisasi, satu lapisan aktifasi ReLu, dan satu *maxpooling layer*. Sementara blok kedua hingga kelima akan diproses dengan beberapa blok yang lebih kecil yang disebut *residual block* yang terdiri dari bebeerapa *convolution 2D layer*, *layer* normalisasi, dan lapisan aktifasi ReLu. Selanjutnya, data akan melalui satu *average pooling layer*, satu *flatten layer*, dan satu *dense layer*.

# 3.4.2 Perancangan Aplikasi



Gambar 3.3 Alur pembuatan aplikasi android



Gambar 3.4 Rancangan *interface* a) Menu utama b) Pilihan Unggah Foto c) Pilihan Ambil Foto Langsung d) Informasi hasil diagnosis

Aplikasi akan terfokus pada kemampuan untuk mendeteksi 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 3.3 merupakan alur perancangan aplikasi.

Proses perancangan aplikasi diawali dengan pembuatan rancangan *interface* aplikasi. Rancangan *interface* kemudian diimplementasikan menggunakan Android Studio. Gambar 3.4 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.

# 3.5 Integrasi Perangkat Lunak dan Perangkat Keras

Tabel 3.1 Tabel spesifikasi smartphone

Spesifikasi	Keterangan
Layar Utama	
Jenis	AMOLED
Ukuran	6.55 inci
Resolusi	1080 x 2400 piksel
Rasio	20:9
Kerapatan	402 ppi
Processor	
Chipset	Qualcomm Snapdragon 778G
CDU	Octa-core (4x2.4 GHz Kryo 670 & 4x1.8 GHz
CIO	Kryo 670)
GPU	Adreno 642L
Memori	
RAM	8 GB
Memori Internal	256 GB
Kamera Utama	
Jumlah Kamera	3
	108 MP (wide), f/1.9
Konfigurasi	8 MP (ultrawide), f/2.2
	2 MP (macro), f/2.4
	Dual-LED dual-tone flash, HDR, panorama,
Fitur	Video: 4K@30fps, 1080p@30/60/120fps; gyro-
	EIS
D	Foto Kamera Utama: 0.6x hingga 10x
reivesaran	Video Kamera Utama: 0.6x hingga 6x
	Layar Utama Jenis Ukuran Resolusi Rasio Kerapatan Processor Chipset CPU GPU Memori RAM Memori Internal Kamera Utama Jumlah Kamera Konfigurasi

Perangkat keras yang digunakan terdiri dari *smartphone* dan kamera fundus. Kamera fundus yang digunakan merupakan kamera *smartphone* yang dilengkapi dengan oftalmoskop funduskopi. Tabel 3.1 merupakan spesifikasi dari *smartphone* yang akan digunakan dalam membuat aplikasi.



Gambar 3.5 Oftalmoskop funduskopi adaptor *smartphone* Sumber: https://www.tokopedia.com/veronikamicrosurgery

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 3.5. Tabel 3.2 merupakan spesifikasi dari lensa dan oftalmoskop funduskopi adaptor *smartphone* yang akan digunakan.

Tabel 3.2 Tabel spesifikasi oftalmoskop funduskopi adaptor smartphone

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

Gambar 3.6 merupakan rancangan dari integrasi *smartphone* dan kamera fundus. Adapun cara penggunaan dari oftalmoskop sebagai berikut:

- 1. Putar tutup lensa, masukkan lensa 20D, dan kencangkan kembali tutup lensa;
- 2. Posisikan kamera *smartphone* dan *blitz* pada posisi pusat lingkaran penjepit;

- 3. Putar knop penjepit untuk fiksasi *smartphone*;
- 4. Atur panjang lengan, biasanya dimulai dengan posisi paling panjang, fiksasi dengan mur putar;
- 5. Nyalakan mode video *smartphone* dengan *blitz* menyala;
- 6. Fiksasi fundus ke dahi pasien dengan tangan non-dominan, dengan jarak kira-kira 3 cm;
- 7. Posisikan cahaya kamera *smartphone* ke arah pusat pupil hingga tampak bayangan *red reflex*;
- 8. Mundurkan fundus secara perlahan hingga didapatkan gambaran retina yang diinginkan.



Gambar 3.6 Oftalmoskop funduskopi adaptor yang terintegrasi dengan *smartphone* a) Tampak dari dapan layar b) Tampak dari belakang layar Sumber: https://www.tokopedia.com/veronikamicrosurgery

# 3.6 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 diabetic retinopathy.

# 3.6.1 Pengujian Fitur Deteksian dari Citra *Gallery* ke Aplikasi

Pengujian menggunakan citra yang diunggah ke aplikasi akan dilakukan dengan mengambil gambar *diabetic retinopathy* di luar *dataset*. Gambar tersebut telah diberi label sehingga dapat dipantau apakah pendeteksian sudah akurat atau belum. Pengujian akan dilakukan

menggunakan 10 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 klasifikasi.

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

# 3.6.2 Pengujian Fitur Deteksi dari Citra Foto Langsung

Sebelum dilakukan pengujian secara langusng kepada subjek penelitian, pengujian akan dilakukan menggunakan kamera *smartphone* untuk mendeteksi citra penderita *diabetic retinopathy* yang sudah ada (tidak secara langsung ke subjek). Hal ini dimaksudkan untuk menguji apakah aplikasi dan pendeteksian secara langusng sudah bisa berjalan sesuai harapan atau belum.

Pengujian citra foto secara langsung akan dilakukan dengan mengambil 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 diabetic retinopathy;
- 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.

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

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 3.3.

Tabel 3.3 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 Lux, $\pm (5\% + 10 \text{ word})$
5.	Repeatability	$\pm 2\%$
6.	Daya listrik	3 x 1,5 volt AAA battery
7.	Suhu operasional	$0^{\circ}\mathrm{C}-40^{\circ}\mathrm{C}$

# 3.6.3 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 3.4

Tabel 3.4 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>

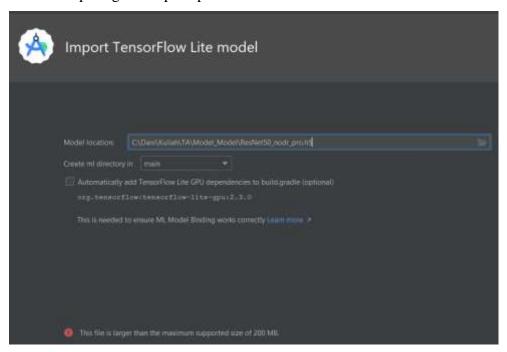
#### **BAB IV**

#### Hasil dan Pembahasan

#### 4.1 Penelitian Arsitektur Model

Pada proyek Tugas Akhir ini, penulis melakukan beberapa penelitian terkait model arsitektur CNN yang harus dibuat karena ada beberapa syarat yang harus dipenuhi. Syarat tersebut yaitu:

- a. 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.
- b. Ukuran model maksimal adalah 200 Mb. Hal ini dibuktikan dengan keadaan di mana penulis telah memasukkan model berukuran lebih dari 200 Mb dan muncul peringatan seperti pada Gambar 4.1.



Gambar 4.1 Peringatan batas ukuran maksimal model pada Android Studio

Berdasarkan syarat tersebut, peneliti melakukan beberapa percobaan seperti yang ditampilkan pada Tabel 4.1.

Tabel 4.1 Perbedaan percobaan pembuatan model CNN

Percobaan	Arsitektur CNN	Dataset	Modeling		
Percobaan 1	VGG-19	Data 1	learning rate = $10^{-4}$	$step \ per \ epoch = 30$	
			epoch = 10	$validate\ step = 30$	
Percobaan 2	VGG-19	Data 2	learning rate = $10^{-4}$	$step \ per \ epoch = 50$	
			epoch = 20	$validate\ step = 30$	
Percobaan 3	VGG-19	Data 3	learning rate = $10^{-4}$	$step\ per\ epoch = 40$	
			epoch = 10	$validate\ step = 30$	
Percobaan 4	ResNet-50	Data 3	learning rate = $10^{-4}$	$step\ per\ epoch = 40$	
			epoch = 10	$validate\ step = 30$	
Percobaan 5	VGG-19	Data 4	learning rate = $10^{-4}$	$step\ per\ epoch = 40$	
			epoch = 10	$validate\ step = 30$	
Percobaan 6	ResNet-50	Data 4	learning rate = $10^{-4}$	$step\ per\ epoch = 40$	
			epoch = 10	$validate\ step = 30$	
Percobaan 7	VGG-19	Data 4	learning rate = 10 <sup>-4</sup>	step per epoch = 50	
			epoch = 20	$validate\ step = 30$	
Percobaan 8	VGG-19	Data 4	learning rate = $10^{-4}$	$step\ per\ epoch = 40$	
	sederhana 1		epoch = 20	$validate\ step = 30$	
Percobaan 9	VGG-19	Data 4	learning rate = $10^{-4}$	$step\ per\ epoch = 40$	
	sederhana 2		epoch = 20	$validate\ step = 30$	

### 4.1.1 Data

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 meneteksi diabetes retino pati. Data yang digunakan merupakan citra fundus retina dari dua sumber. Namun, dalam penelitian ini data tersebut ada yang tetap dibiarkan secara original, ada yang

diambil kelas tertentunya saja, dan ada yang mendapat perlakuan khusus. Data 1, data 2, dan data 3 diambil dari satu sumber yang sama yaitu <a href="https://www.kaggle.com/competitions/diabetic-retinopathy-detection">https://www.kaggle.com/competitions/diabetic-retinopathy-detection</a>. Tetapi, data 4 berasal dari sumber yang berbeda. Untuk lebih lengkapnya terkait perbedaan keempat jenis data tersebut, dapat dilihat pada Tabel 4.2.

Tabel 4.2 Pelabelan jenis data dan perbedaannya

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

Diakses pada tanggal 19 January 2024.	
Data hanya dibagi menjadi 80% data	
train, 10% data test, dan 10% data	
validasi.	

### 4.1.1.1 Data 1

Data 1 bersumber dari <a href="https://www.kaggle.com/competitions/diabetic-retinopathy-detection">https://www.kaggle.com/competitions/diabetic-retinopathy-detection</a>. Data tersebut kemudian dibagi menjadi data *train*, *test*, dan *val* dengan rincian seperti yang tampak pada Tabel 4.3.

Tabel 4.3 Dataset 1

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

### 4.1.1.2 Data 2

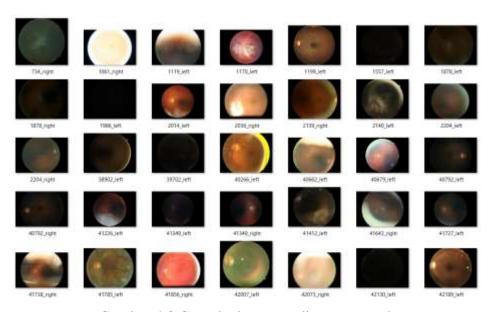


Gambar 4.2 Perbandingan retina *No DR* dengan keempat kelas lainnya Sumber: Vitriana, 2023

Data 2 merupakan Data 1 (poin 4.1.1.1), 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 dibanding kelas yang lain. Gambar 4.2 menampilkan perbedaan retina *No DR* dan keempat kelas lainnya.

### 4.1.1.3 Data 3

Data 3 merupakan Data 2 (poin 4.1.1.2) 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 4.3.



Gambar 4.3 Contoh citra yang dianggap rusak

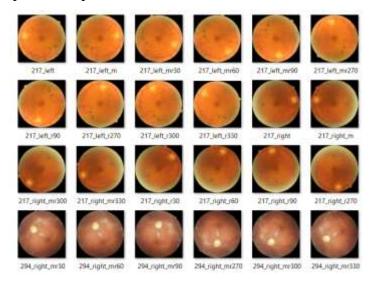
Data *No\_DR* juga kemudian dilakukan crop image dengan potongan kode yang dapat dilihat pada Lampiran 1.

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 sementara contoh hasil duplikasi dapat dilihat pada Gambar 4.4.



Gambar 4.4 Contoh citra *Proliferative DR* yang digandakan

Hasil akhir setelah mengalami pengolahan dapat dilihat pada Tabel 4.4.

Tabel 4.4 Dataset 1 yang sudah imoifikasi untuk kelas No DR dan Proliferative

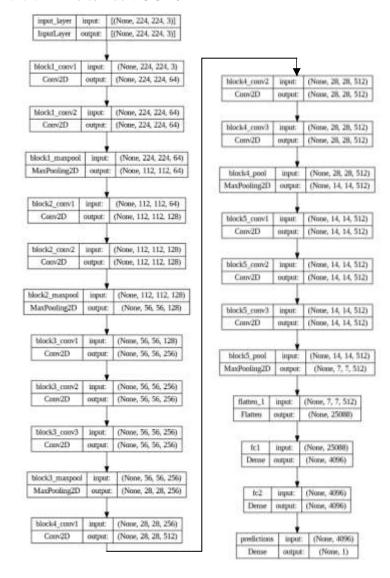
Train	Test	Validate
No DR = 7.122	No DR = 2.433	No DR = 2.375
Proliferative = 5.824	Proliferative = 837	Proliferative = 1.946

#### 4.1.1.4 Data 4

Data 4 bersumber dari <a href="https://www.kaggle.com/datasets/">https://www.kaggle.com/datasets/</a> 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*.

#### 4.1.2 Arsitektur

#### 4.1.2.1 Arsitekrut VGG-19



Gambar 4.5 Arsitektur VGG-19

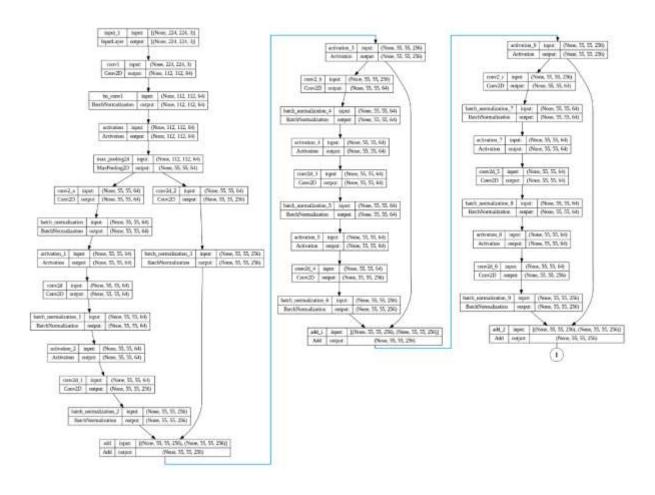
Gambar 4.5 merupakan bentuk arsitektur model VGG-19 yang telah dibuat. Pada model tersebut, citra RGB berukuran 224x224 *pixel* akan diolah melalui lima blok pengolahan yang kemudian dilakukan *flatten* dan *dense* sehingga menghasilkan satu keluaran. Masing-masing blok terdiri dari dua atau tiga *layer convolution 2D* dan diakhiri *max pooling layer* dengan dimensi yang dapat dilihat pada Gambar 4.5.

#### 4.1.2.2 Arsitektur *ResNet-50*

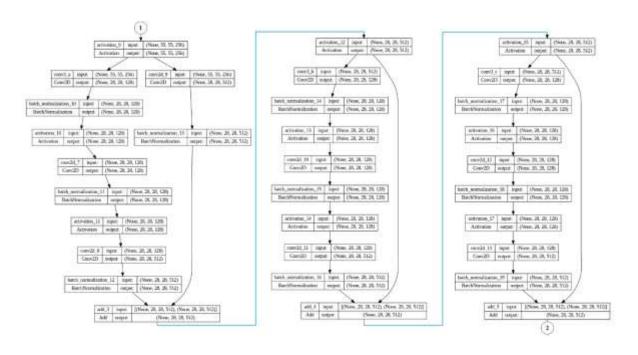
Gambar 4.6 hingga Gambar 4.9 merupakan bentuk arsitektur model *ResNet-50* yang telah dibuat. Pada model tersebut, citra RGB berukuran 224x224 *pixel* akan diproses melalui lima blok *convolution*. Pada blok pertama, data akan melalui satu *convolution 2D layer*, satu *layer* normalisasi, satu lapisan aktifasi 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 aktifasi ReLu. Selanjutnya, data akan melalui satu *average pooling layer*, satu *flatten layer*, dan satu *dense layer*.

Blok *convolution* kedua memiliki total tiga residual blok. Blok *convolution* ketiga memiliki total empat residual blok. Blok *convolution* keempat memiliki total enam residual blok. Blok *convolution* kelima memiliki total tiga residual blok. 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.

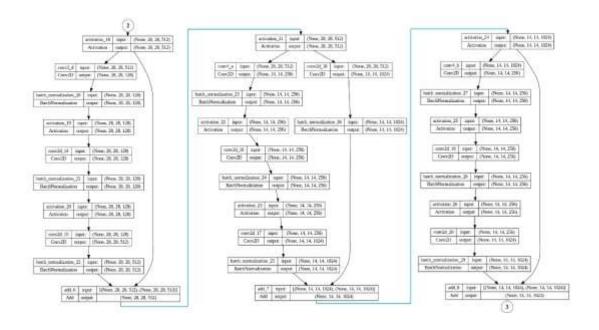
Gambar 4.6 merupakan gambar untuk blok *convolution* pertama hingga blok *convolution* kedua. Gambar 4.7 merupakan gambar untuk blok *convolution* ketiga untuk blok residual pertama hingga ketiga. Gambar 4.8 merupakan gambar untuk blok *convolution* ketiga untuk blok residual keempat dan dilanjut blok *convolution* keempat-blok residual pertama hingga kedua. Gambar 4.9 merupakan gambar untuk blok *convolution* keempat untuk blok residual ketiga hingga keenam. Gambar 4.10 merupakan gambar untuk blok *convolution* kelima untuk blok residual pertama hingga ketiga dan dilanjutkan dengan tiga layer penutup.



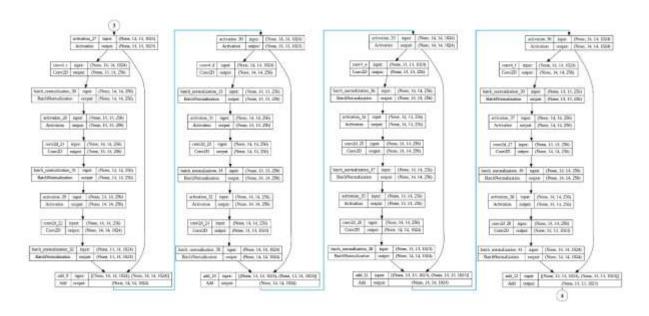
Gambar 4.6 Arsitektur ResNet-50 bagian 1



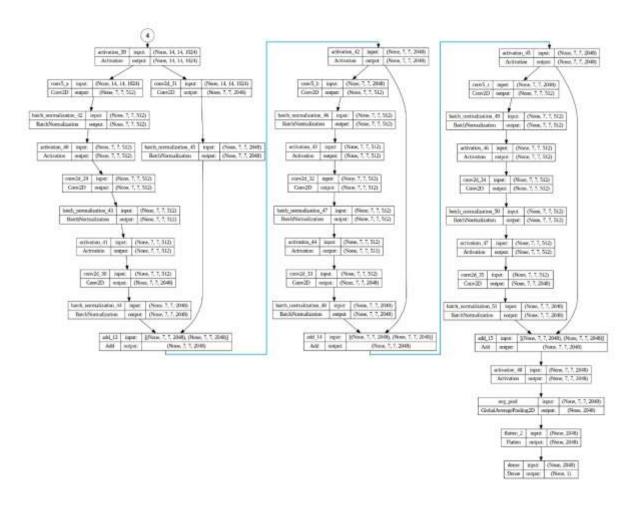
Gambar 4.7 Arsitektur ResNet-50 bagian 2



Gambar 4.8 Arsitektur ResNet-50 bagian 3



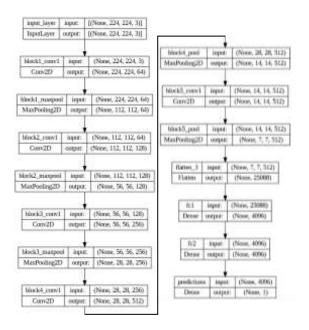
Gambar 4.9 Arsitektur ResNet-50 bagian 4



Gambar 4.10 Arsitektur ResNet-50 bagian 5

## 4.1.2.3 Arsitektur VGG-19 sederhana 1

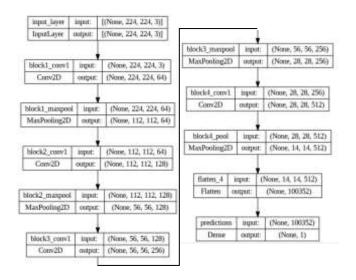
Gambar 4.11 merupakan bentuk arsitektur model VGG-19 sederhana 1 yang telah dibuat. Pada model tersebut, citra RGB berukuran 224x224 *pixel* akan diolah melalui lima blok pengolahan yang kemudian dilakukan *flatten* dan *dense* sehingga menghasilkan satu keluaran. Masing-masing blok terdiri dari satu *layer convolution 2D* dan diakhiri *max pooling layer* dengan dimensi yang dapat dilihat pada Gambar 4.11.



Gambar 4.11 Arsitektur VGG-19 sederhana 1

#### 4.1.2.4 Arsitektur VGG-19 sederhana 2

Gambar 4.12 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 4.12.



Gambar 4.12 Arsitektur VGG-19 sederhana 2

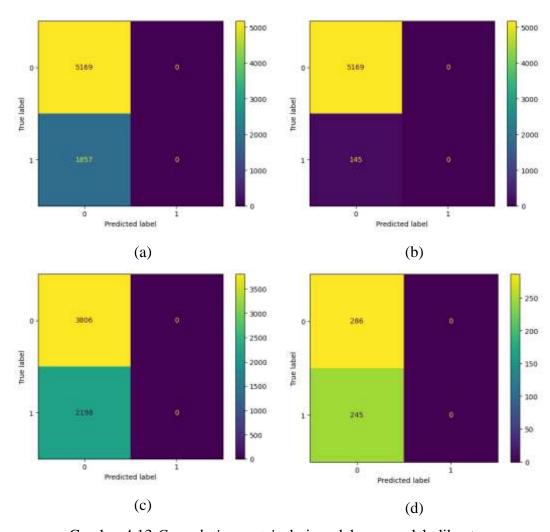
# 4.2 Hasil Percobaan Model

Tabel 4.5 Perbedaan hasil percobaan pembuatan model CNN

Percobaaan Akurasi dan Loss		Waktu Learning	Ukuran Model
Percobaan 1 (VGG19.h5)	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
Percobaan 2 (VGG19_nodr_prodr .h5)	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
Percobaan 3 (VGG19_nodr_prodr _nogray.h5)	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
Percobaan 4 (ResNet50_nodr_pro .h5)	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
Percobaan 5 (VGG19_nodr_prodr _data4.h5)	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
Percobaan 6 (ResNet50_nodr_pro _data4.h5)	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
Percobaan 7 (VGG19_nodr_prodr _data4_try2.h5)	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

Percobaan 8 (VGG19_nodr_prodr _data4_try3.h5)	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
Percobaan 9 (Try_Model_4.h5)	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 4.13 menampilkan hasil dari convolution matrix dari percobaan pembuatan model deteksi yang sudah dibuat. Gambar 4.13 poin (a) merupakan percobaan 1, poin (b) untuk percobaan 2, poin (c) untuk percobaan 3 dan 4, sementara poin (d) untuk percobaan 5 hingga 9.



Gambar 4.13 Convolution matrix dari model yang sudah dibuat

Berdasarkan hasil yang ditunjukkan pada Tabel 4.5, dapat disimpulkan bahwa dalam pembuatan model deteksi, yang memiliki akurasi paling tinggi ditunjukkan oleh percobaan 2 disusul percobaan 8 dan 9. Berdasarkan kecepatan proses pembelajaran maka urutannya adalah percobaan 8 disusul percobaan 6 dan 9. Sementara itu, berdasarkan ukuran model yang paling ringan maka urutannya adalah percobaan 9 disusul percobaan 4 dan 6. Namun, untuk matrik konvolusi, tidak menampilkan perbedaan dalam melakukan pendeteksian dan semua dideteksi sebagai No\_DR. untuk memastikan kemampuan model, maka model dipanggil untuk menguji beberapa citra dan didapatkan hasil seperti pada Tabel 4.6.

Tabel 4.6 Perbandingan hasil prediksi model CNN

Keterangan:

T1 : Data 4 valid DR T5 : Data 3 valid No\_DR

T2 : Data 4 valid No\_DR T6 : Data 3 test No\_DR

T3 : Data 3 valid Prolifer DR T7 : Data 4 test No\_DR

T4: Data 3 test Prolifer DR

11. Data 5 test 110mer DR								
Percobaaan		Hasil Pendeteksian						
	T1	No_DR - 113	DR - 0	T5	No_DR - 200	DR - 0		
Percobaan 1	T2	No_DR - 118	DR - 0	T6	No_DR - 200	DR - 0		
(VGG19.h5)	Т3	No_DR - 200	DR - 0	T7	No_DR - 200	DR - 0		
	T4	No_DR - 200	DR - 0					
Percobaan 2	T1	No_DR - 113	DR - 0	T5	No_DR - 200	DR - 0		
	T2	No_DR - 118	DR - 0	T6	No_DR - 200	DR - 0		
(VGG19_nodr_	Т3	No_DR - 200	DR - 0	T7	No_DR - 200	DR - 0		
prodr.h5)	T4	No_DR - 200	DR - 0					
Percobaan 3	T1	No_DR - 113	DR - 0	T5	No_DR - 200	DR - 0		
	T2	No_DR - 118	DR - 0	T6	No_DR - 200	DR - 0		
(VGG19_nodr_	Т3	No_DR - 200	DR - 0	T7	No_DR - 200	DR - 0		
prodr_nogray.h5)	T4	No_DR - 200	DR - 0					
Percobaan 4	T1	No_DR - 113	DR - 0	T5	No_DR - 200	DR - 0		
	T2	No_DR - 118	DR - 0	T6	No_DR - 200	DR - 0		
(ResNet50_nodr	Т3	No_DR - 200	DR - 0	Т7	No_DR - 200	DR - 0		
_pro.h5)	T4	No_DR - 200	DR - 0					

Danashaan 5	T1	No_DR - 13	DR - 100	T5	No_DR - 164	DR - 36
Percobaan 5	T2	No_DR - 113	DR - 5	T6	No_DR - 159	DR - 41
(VGG19_nodr_	Т3	No_DR - 123	DR - 77	T7	No_DR - 190	DR - 10
prodr_data4.h5)	T4	No_DR - 163	DR - 37			
Percobaan 6	T1	No_DR - 113	DR - 0	T5	No_DR - 200	DR - 0
(ResNet50_nodr_	T2	No_DR - 118	DR - 0	T6	No_DR - 200	DR - 0
	Т3	No_DR - 200	DR - 0	T7	No_DR - 200	DR - 0
pro_data4.h5)	T4	No_DR - 200	DR - 0			
Percobaan 7	T1	No_DR - 21	DR - 92	T5	No_DR - 179	DR - 21
(VGG19_nodr_	T2	No_DR - 116	DR - 2	T6	No_DR - 187	DR - 13
prodr_data4_	Т3	No_DR - 174	DR - 26	T7	No_DR - 195	DR - 5
try2.h5)	T4	No_DR - 186	DR - 14			
Percobaan 8	T1	No_DR - 14	DR - 99	T5	No_DR - 136	DR - 64
(VGG19_nodr_	T2	No_DR - 107	DR - 11	T6	No_DR - 136	DR - 64
prodr_data4_	Т3	No_DR - 116	DR - 84	T7	No_DR - 175	DR - 25
try3.h5)	T4	No_DR - 129	DR - 71			
Percobaan 9	T1	No_DR - 14	DR - 99	T5	No_DR - 180	DR - 20
	T2	No_DR - 109	DR - 9	T6	No_DR - 180	DR - 20
(Try_Model_	Т3	No_DR - 133	DR - 67	T7	No_DR - 186	DR - 14
4.h5)	T4	No_DR - 158	DR - 42			

Berdasarkan Tabel 4.6, percobaan 1, 2, 3, 4, dan 6 mendeteksi citra No\_DR sepenuhnya. Sementara untuk percobaan lainnya, dapat dilihat bahwa data sebagian besar menebak sebagai No\_DR meski citra yang ditampilkan merupakan citra DR. Namun, jika diperhatikan berdasarkan hasil, pengujian dari T1 untuk percobaan 5, 7, 8, dan 9 berhasil melakukan prediksi dengan benar lebih dari 80% sementara untuk T3 dan T4 keberhasilannya kurang dari 50%. Melihat gagalnya proses pembuatan model percobaan 3 dan 4, menunjukkan bahwa kurang teraturnya Data 3 dapat menyebabkan kegagalan pembuatan model maupun pengujian akhir. Data 3 sendiri merupakan data yang telah diproses lebih lanjut dari Data 2 dan Data 1. Maka dari itu, untuk menghindari kegagalan, lebih baik dianjurkan menggunakan model dan melakukan pengujian dengan Data 4 sebagai acuan.

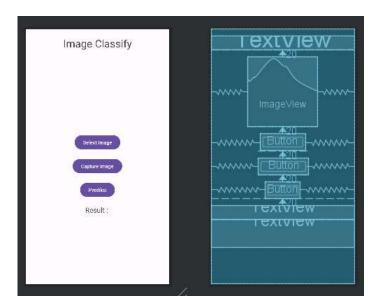
Tabel 4.7 Rata-rata prediksi benar dari model yang dipilih

Percobaan 5	T1	No_DR - 13	DR - 100	Benar: 88,495%				
(VGG19_nodr_	T2	No_DR - 113	DR - 5	Benar : 95,763%				
prodr_data4.h5)	T7	No_DR - 190	DR - 10	Benar : 95,000%				
prodr_data (me)	Rat	Rata-rata benar = 93,086%						
Percobaan 7	T1	No_DR - 21	DR - 92	Benar: 81,416%				
(VGG19_nodr_	T2	No_DR - 116	DR - 2	Benar : 98,305%				
prodr_data4_try2.h5)	T7	No_DR - 195	DR - 5	Benar : 97,500%				
Francisco,	Rata-rata benar = 92,407%							
Percobaan 8	T1	No_DR - 14	DR - 99	Benar: 87,611%				
(VGG19_nodr_	T2	No_DR - 107	DR - 11	Benar : 90,678%				
prodr_data4_try3.h5)	T7	No_DR - 175	DR - 25	Benar: 87,500%				
r	Rat	a-rata benar = 88	8,596%					
	T1	No_DR - 14	DR - 99	Benar: 87,611%				
Percobaan 9	T2	No_DR - 109	DR - 9	Benar : 92,373%				
(Try_Model_4.h5)	T7	No_DR - 186	DR - 14	Benar: 83,000%				
	Rat	a-rata benar = 8°	7,661%					

Berdasarkan hasil pendeteksian dari percobaan 5, 7, 8, dan 9, didapati bahwa model dengan akurasi tertinggi dimiliki oleh percobaan 5. Namun, karena batasan model yang bisa diterima oleh Android Studio, maka dipilihlah model dari percobaan 9 yang memiliki selisih akurasi sebesar ±6% dibanding model percobaan 5. Meski demikian, model dari percobaan 9 masih dianggap layak sebab memiliki persentase keberhasilan pendeteksian diatas 85%.

### 4.3 Pembuatan Aplikasi Android

Aplikasi android dibuat menggunakan Android Studio. Desain pembuatan tampilan dari aplikasi dapat dilihat pada Gambar 4.14.



Gambar 4.14 Tampilan menu utama dari aplikasi android

Terdapat beberapa fungsi penting untuk mengatur interaksi tombol dan tampilan lainnya pada aplikasi tersebut. Beberapa fungsi tersebut dapat dilihat pada Gambar 4.15.

```
064
    // Set button to active gallery view click action,
    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) {
              Intent intent = new Intent(Intent.ACTION PICK,
079
              MediaStore.Image.Media.EXTERNAL CONTENT URI);
              startActivityForResult(intent, 12);
080
081
082
    });
```

Gambar 4.15 Potongan kode fungsi interaksi tombol dan tampilan

Potongan kode pada Gambar 4.15 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 4.15. Fungsi tersebut akan diaktifkan jika pengguna menekan tombol "Capture Image". Selanjutnya, dalam kedua fungsi tersebut terdapat pemanggilan fungsi startActivityForResult dan mengirimkan requestCode yang berbeda. Fungsi ini digunakan untuk mengaktifasi perintah apa yang akan dijalankan selanjutnya jika tombol bersangkutan ditekan. Fungsi startActivityForResult dapat dilihat pada Gambar 4.16. 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.

```
public
              void
                      onActivityResult(int
                                               requestCode,
    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
                                      MediaStore.Images.Media.get
                        bitmap
                                 =
                        Bitmap(this.getContentResolver(), uri);
192
                                 =
                                       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
                                  ThumbnailUtils.extractThumbnail
                   bitmap
                   (bitmap, dimension, dimension);
204
                   bitmap
                           =
                                Bitmap.createScaledBitmap(bitmap,
                   224, 224, false);
205
                   imageView.setImageBitmap(bitmap);
206
              }
207
          }
208
```

Gambar 4.16 Potongan kode fungsi onActivityResult

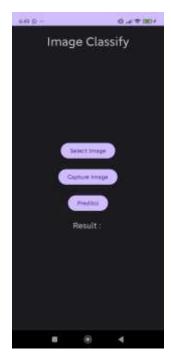
Gambar 4.17 menampilkan fungsi yang akan aktif saat tombol "Prediksi" dipilih. Baris 76 merupakan proses pemanggilan model CNN. Baris 79 hingga 97 merupakan proses persiapan buffer untuk mengolah data gambar RGB yang diterima model. Baris 99 hingga 101 merupakan proses pendeteksian citra. Baris 103 hingga 111 merupakan bagian untuk menentukan indeks tertinggi dan menentukan label citra dan tingkat akurasinya. Baris 113 dan 115 berfungsi menampilkan hasil dari pendeteksian.

```
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
               // This is for try catch if there was an image or it
092
              wasn't
093
               try {
                     pred = bitmap.getByteCount();
094
095
               }
096
              catch (Exception e) {
097
                     result.setText("Choose the image");
098
               }
099
100
              if (pred > 0) {
                     \ensuremath{//} Detection process
101
102
                     try {
103
                                                ModelFix.newInstance
                          ModelFix
                                    model
                          (MainActivity.this);
104
105
                          // Creates inputs for reference
106
                          TensorBuffer
                                        inputFeature0
                                                               Tensor
                          Buffer.createFixedSize(new
                                                         int[]{1,224,
                          224, 3}, DataType.FLOAT32);
107
108
                          // Prepare for buffer image
109
                          ByteBuffer byteBuffer
                                                          ByteBuffer.
                          allocateDirect(4 * 224 * 224 * 3);
110
                          byteBuffer.order(ByteOrder.nativeOrder());
111
112
                          // Match the data pixel into the buffer
113
                          int [] intValues = new int [224 * 224];
114
                          bitmap.getPixels(intValues, 0, bitmap.get
                          Width(), 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));
```

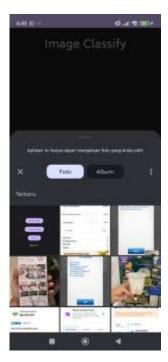
```
121
                                  byteBuffer.putFloat((val & 0xFF) *
                                  (1.f / 1));
122
                               }
123
                         }
124
125
                         inputFeature0.loadBuffer(byteBuffer);
126
127
                         // Runs model inference and gets result.
128
                         ModelFix.Outputs outputs = model.process
                         (inputFeature0);
129
                         TensorBuffer outputFeature0 = outputs.
                         getOutputFeatureOAsTensorBuffer();
130
                         // Get prediction and confidence
131
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
                                                         * 100)).
                         (String.valueOf(maxConfidence
                         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 4.17 Potongan kode fungsi prediksi dan pemanggilan model

Hasil dari tampilan aplikasi yang sudah dibaut dapat dilihat pada Gambar 4.18 hingga Gambar 4.21.



Gambar 4.18 Tampilan menu utama aplikasi



Gambar 4.19 Tampilan aplikasi tombol "Select Image"





Gambar 4.20 Tampilan aplikasi tombol "*Capture Image*" (a) sebelum ambil gambar (b) setelah ambil gambar



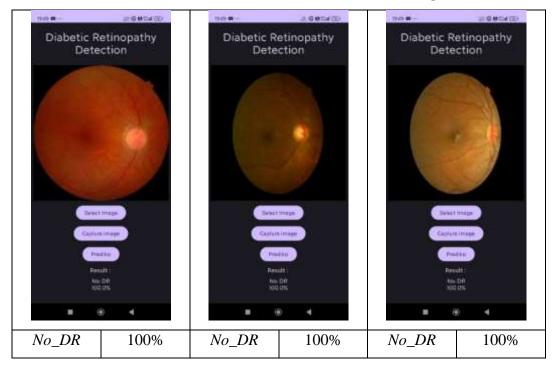
Gambar 4.21 Tampilan aplikasi tombol "Prediksi"

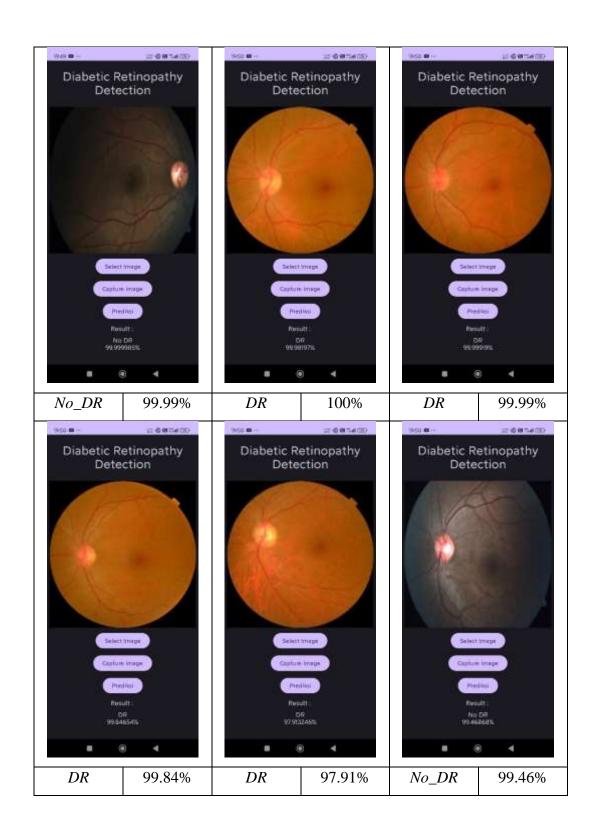
# 4.4 Pengujian Deteksi dengan Aplikasi Android

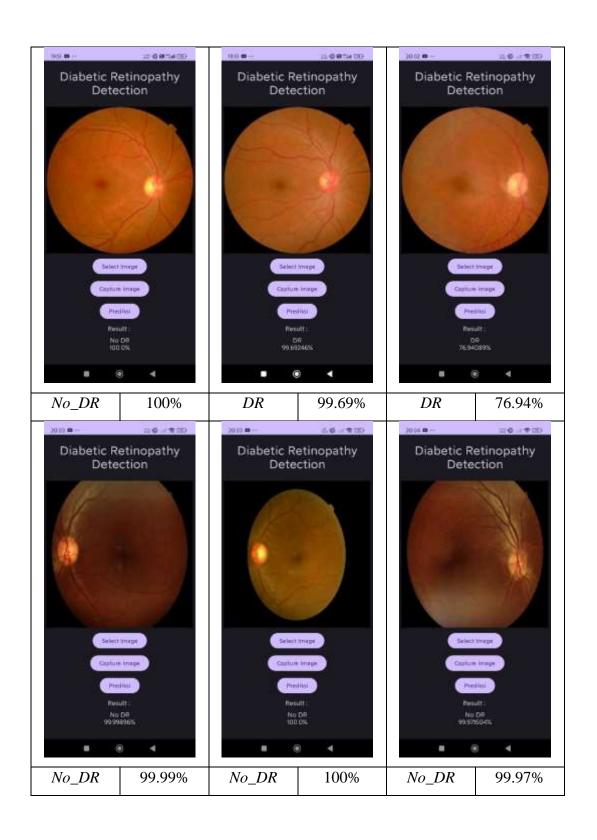
# 4.4.1 Pengujian Deteksi Menggunakan Citra dari Gallery

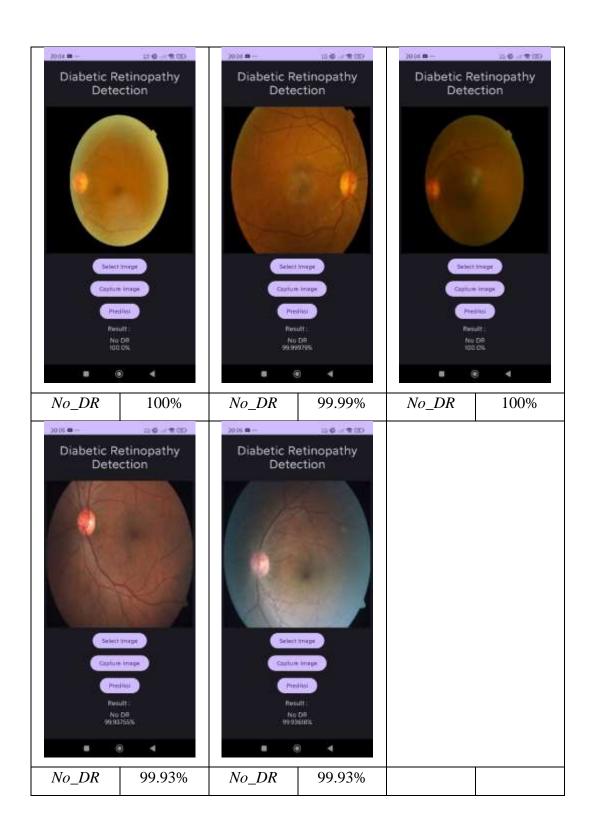
Tabel 4.8 dan Tabel 4.9 merupakan tabel hasil pendeteksian diabetic retinopathy melalui aplikasi meggunakan citra dari Data 4 Test yang diunggah ke aplikasi. Tabel 4.8 menunjukkan data citra *No\_DR* dengan hasil pendeteksian dan akurasinya. Tabel 4.9 menunjukkan data citra *DR* dengan hasil pendeteksian dan akurasinya.

Tabel 4.8 Hasil deteksi citra No\_DR dari memori handphone

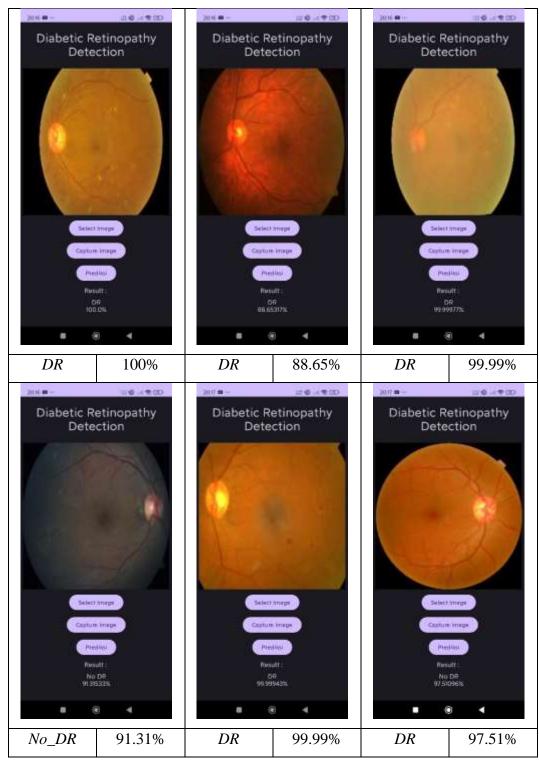


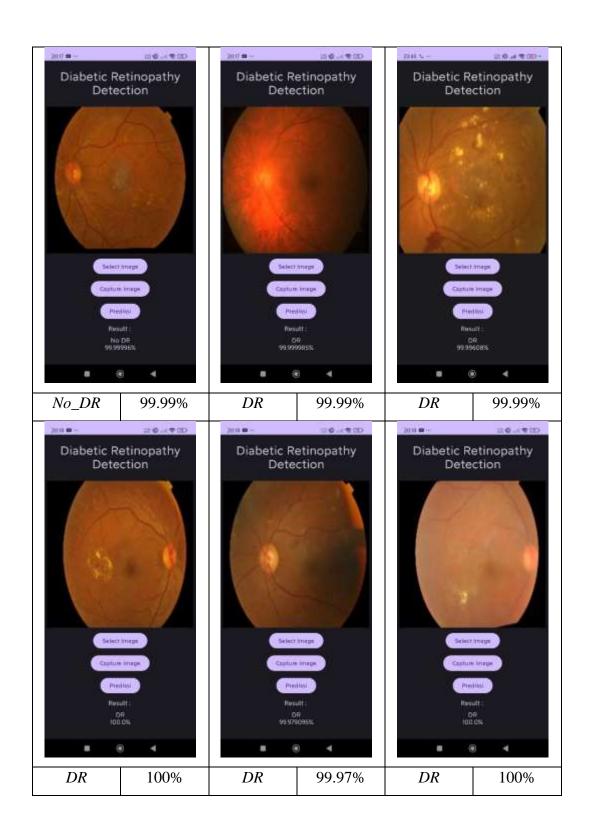


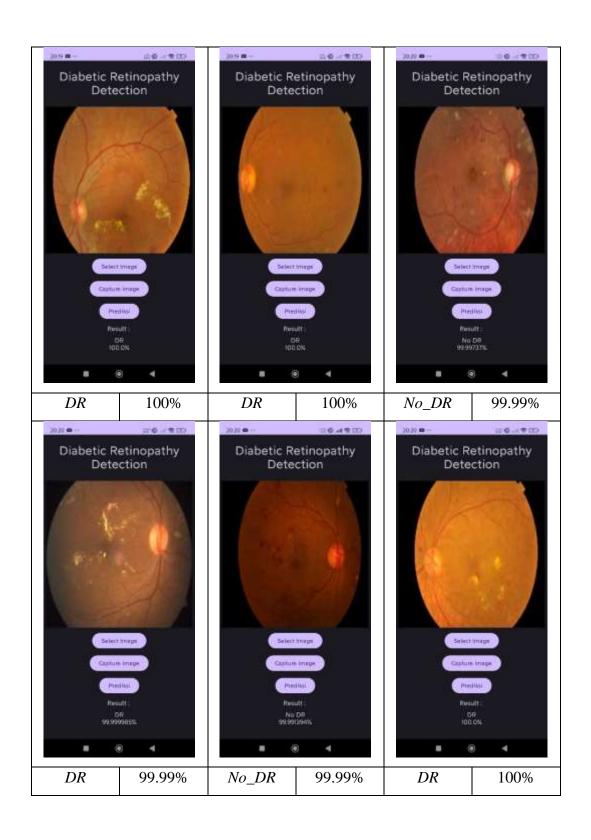


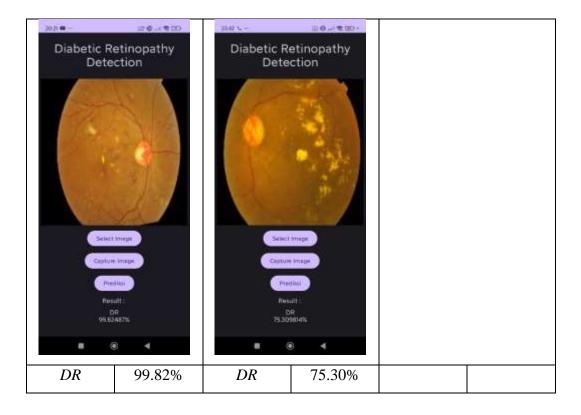


Tabel 4.9 Hasil deteksi citra DR dari memori handphone









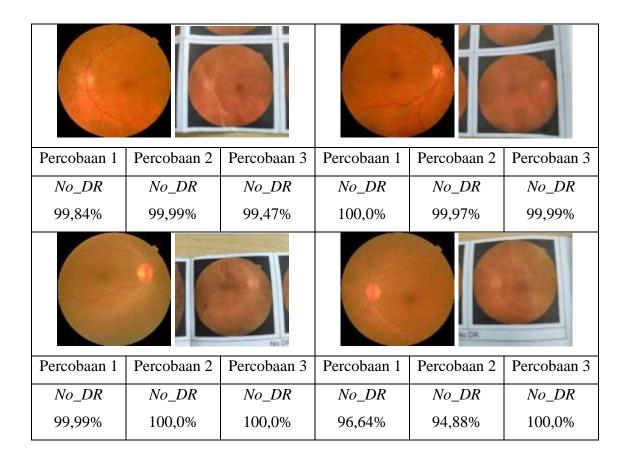
Berdasarkan Tabel 4.8, dari 20 citra  $No\_DR$  terdapat 6 citra yang dideteksi sebagai DR. Data tersebut menunjukkan keberhasilan untuk mendeteksi citra  $No\_DR$  sebesar  $\frac{14}{20} \times 100\% = 70\%$ . Sementara itu, Berdasarkan Tabel 4.9, dari 20 citra DR terdapat 4 citra yang dideteksi sebagai  $No\_DR$ . Data tersebut menunjukkan keberhasilan untuk mendeteksi citra DR sebesar  $\frac{16}{20} \times 100\% = 80\%$ .

### 4.4.2 Pengujian Deteksi dengan Kamera pada Citra Gambar Cetak

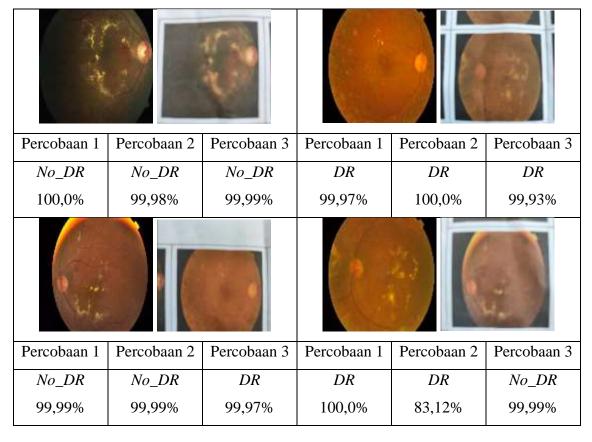
Tabel 4.10 dan Tabel 4.11 merupakan tabel hasil pendeteksian diabetic retinopathy melalui aplikasi meggunakan citra dari Data 4 Test yang diunggah ke aplikasi. Tabel 4.10 menunjukkan data citra *No\_DR* dengan hasil pendeteksian dan akurasinya. Tabel 4.11 menunjukkan data citra *DR* dengan hasil pendeteksian dan akurasinya.

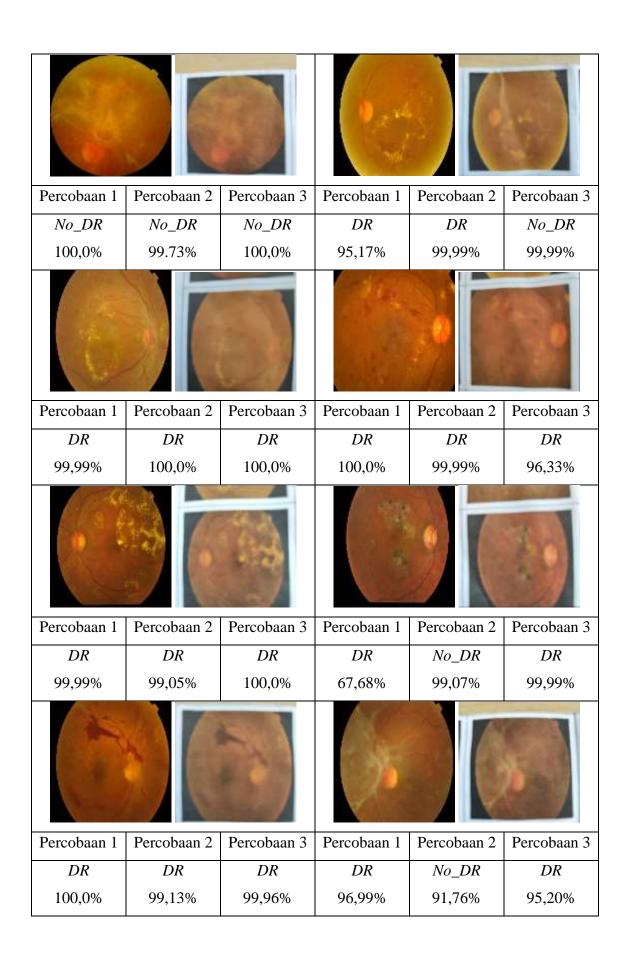
Tabel 4.10 Hasil deteksi citra  $No\_DR$  dari kamera pada gambar cetak

			dan kamera pada gambar eetak		
Percobaan 1	Percobaan 2	Percobaan 3	Percobaan 1	Percobaan 2	Percobaan 3
No_DR	DR	No_DR	No_DR	No_DR	No_DR
99,96%	58,76%	100,0%	100,0%	99,99%	100,0%
Percobaan 1	Percobaan 2	Percobaan 3	Percobaan 1	Percobaan 2	Percobaan 3
No_DR	DR	DR	No_DR	No_DR	No_DR
100,0%	96,53%	82,81%	100,0%	99,99%	52,72%
Percobaan 1	Percobaan 2	Percobaan 3	Percobaan 1	Percobaan 2	Percobaan 3
No_DR	No_DR	No_DR	No_DR	No_DR	No_DR
100,0%	99,99%	100,0%	99,99%	99,96%	100,0%
Percobaan 1	Percobaan 2	Percobaan 3	Percobaan 1	Percobaan 2	Percobaan 3
Percobaan 1  No_DR	Percobaan 2  No_DR	Percobaan 3  No_DR	Percobaan 1  No_DR	Percobaan 2  No_DR	Percobaan 3  No_DR



Tabel 4.11 Hasil deteksi citra DR dari kamera pada gambar cetak





Pengujian dilakukan sebanyak tiga kali untuk masing-masing citra. Maka dari itu, data akan mengikuti hasil pendeteksian yang diakui setidaknya dua kali. Berdasarkan data pada Tabel 4.10, 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 4.11, 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 Pengujuan Deteksi dengan Kamera pada Citra Retina Asli

Pada tahap ini, peneliti melakukan penerapan rancangan dengan desain yang tampak pada Gambar 4.22.



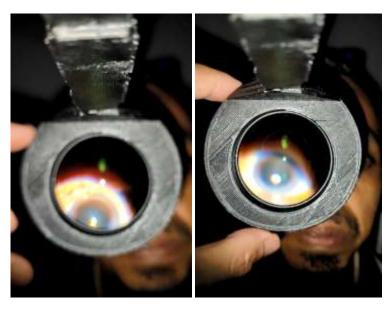


Gambar 4.22. Penerapan perangkat keras *smartphone* dan lensa 20D

Pada Gambar 4.22. Bagian A meerupakan lensa 20D, bagian B merupakan perangkat *smartphone*, dan bagian C merupakan adaptor dengan panjang lengan maksimal 25cm.

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.
- 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 ±8cm. 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 4.23 Penangkapan citra fundus dengan lensa 20D, adaptor, dan *smartphone* 

 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 ±8cm. 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 4.23.

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 4.24 Penangkapan citra fundus dengan lensa 20D dan *smartphone* 



Gambar 4.25 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 ±4cm dan jarak lensa ke kamera dimulai dari ±15cm dijauhkan perlahan hingga ±30cm. Namun, pada metode ini fundus juga

masih belum berhasil didapatkan. Gambar 4.24 menunjukkan hasil dari pengambilan citra, sementara Gambar 4.25 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 ±20cm 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 4.26 merupakan kondisi pengambilan citra fundus menggunakan direct oftalmoskop.

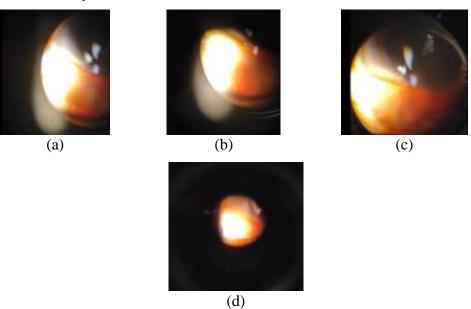


Gambar 4.26 Penangkapan citra fundus dengan *direct oftalmoskop* dan *smartphone* 

#### 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 ±7cm 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 4.27 merupakan hasil pengambilan citra fundus sementara Gambar 4.28 merupakan kondisi pengambilan citra fundus menggunakan lensa 78D, *slit lamp*, dan kamera *smartphone*.



Gambar 4.27 Penangkapan citra fundus dengan lensa 78D, *slit lamp* dan *smartphone* 



Gambar 4.28 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 sudadh diatur pada < 60 lux dan dengan penambahan *flash smartphone*. Pada tahap ini, peneliti menggunakan beberapa variabel dengan variasi nilai yang terdapat pada Tabel 4.12:

Tabel 4.12 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 smartphone	20 cm, 30 cm, 40 cm, 50 cm
Perbesaran kamera smartphone	1x, 2x, 5x, 10x

Berdasarkan penelitian yang dilakukan, hasil yang sempat berhasil mendapatkan citra terdapat pada Tabel 4.13:

Tabel 4.13 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 smartphone	±25 cm hingga ±30 cm
Perbesaran kamera smartphone	5x

Dengan demikian, untuk menangkap citra fundus secara optimal saat menggnuakan lensa 20D, dapat digunakan kondisi seperti yang terdapat pada Tabel 4.13.

#### **BAB V**

### **Penutup**

### 5.1 Kesimpulan

Arsitektur VGG-19 sederhana sudah dibuat dengan *dataset* dari <a href="https://www.kaggle.com/datasets/pkdarabi/diagnosis-of-diabetic-retinopathy">https://www.kaggle.com/datasets/pkdarabi/diagnosis-of-diabetic-retinopathy</a> 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.

Aplikasi android untuk meneteksi 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%.

#### 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 *venous*, *cotton wool spots*, *mikroaneurisma*, *hard exedute*, 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, <a href="https://www.acerid.com/laptop-swift-3-acer-day-edition-sf314-56g/">https://www.acerid.com/laptop-swift-3-acer-day-edition-sf314-56g/</a>.
- 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
- 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.
- 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 Muhammadiah 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: <a href="https://www.ncbi.nlm.nih.gov/books/NBK585111">https://www.ncbi.nlm.nih.gov/books/NBK585111</a>
- 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 <a href="https://p2ptm.kemkes.go.id/tag/diabetes-penderita-di-indonesia-bisa-mencapai-30-juta-orang-pada-tahun-2030">https://p2ptm.kemkes.go.id/tag/diabetes-penderita-di-indonesia-bisa-mencapai-30-juta-orang-pada-tahun-2030</a>.
- Pradeep. (2023, Mei 23). What is the difference between VGG16 and VGG19 neural network? Retrieved Oktober 13, 2023, from www.quora.com: <a href="https://www.quora.com/What-is-the-difference-between-VGG16-and-VGG19-neural-network">https://www.quora.com/What-is-the-difference-between-VGG16-and-VGG19-neural-network</a>
- 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.
- 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

```
def crop_image_from_gray(img,tol=7):
        if img.ndim ==2:
            mask = img>tol
            return img[np.ix_(mask.any(1),mask.any(0))]
        elif img.ndim==3:
            gray_img = cv2.cvtColor(img, cv2.COLOR_RGB2GRAY)
            mask - gray img>tol
            check_shape = img[:,:,0][np.ix_(mask.any(1),mask.any(0))].shape[0]
            if (check_shape == 0): # image is too dark so that we crop out everything,
                return img # return original image
                img1=img[:,:,0][np.ix_(mask.any(1),mask.any(0))]
                img2=img[:,:,1][np.ix_(mask.any(1),mask.any(0))]
img3=img[:,:,2][np.ix_(mask.any(1),mask.any(0))]
                img = np.stack([img1,img2,img3],axis--1)
            return img
19 def copy_crop_nodr(img):
        im = Image.open(img)
        numpy_array = np.array(im)
        im - crop_image_from_gray(numpy_array)
       data = Image.fromarray(im)
```

### Lampiran 2 Kode untuk Menggandakan Data No DR

```
1 def generate data moderate(image, directory, namefile len):
      img = Image.open(image)
      numpy_array = np.array(img)
      im - crop_image_from_gray(numpy_array)
     data = Image.fromarray(im)
      os.chdir(directory)
      filename = image[namefile_len:-5] + png
      data.save(filename)
      mirror(img, directory, (image namefile len:-5 + ' m.png'))
      rotate_image(img, 30, directory, (image namefile_len:-5] + '_r30.png'))
      rot_mir(img, 30, directory, (image namefile_len:-5] + '_mr30.png'))
      rotate image(img, 60, directory, (image namefile len:-5] + ' r60.png'))
      rot_mir(img, 60, directory, (image[namefile_len:-5] + '_mr60.png'))
      rotate_image(img, 90, directory, (image namefile_len:-5 + ' r90.png'))
      # 8. SAVE THE MIRROR - ROTATE 90
      rot_mir(img, 90, directory, (image nameFile_len:-5] + '_mr90.png'))
      rotate image(img, 330, directory, (image[namefile_len:-5] + "r330.png"))
      rot_mir(img, 330, directory, (image[namefile_len:-5] + '_mr330.png'))
      rotate_image(img, 300, directory, (image[namefile_len:-5] + '_r300.png'))
      rot_mir(img, 300, directory, (image[namefile_len:-5] + '_mr300.png'))
      rotate_image(img, 270, directory, (image[namefile_len:-5] + '_r270.png'))
      rot_mir(img, 270, directory, (image namefile_len: 5] + '_mr270.png'))
      print(image[namefile_len:-5] + ' finish')
```