

**IMPLEMENTASI CNN TERHADAP SISTEM KLASIFIKASI PENYAKIT
PADA RETINA MATA MANUSIA MENGGUNAKAN MODEL
MOBILENETV3 BERBASIS WEBSITE**

SKRIPSI

CHALIL AL VAREEL

201401043



**PROGRAM STUDI S-1 ILMU KOMPUTER
FAKULTAS ILMU KOMPUTER DAN TEKNOLOGI INFORMASI
UNIVERSITAS SUMATERA UTARA
2024**

UNIVERSITAS SUMATERA UTARA

**IMPLEMENTASI CNN TERHADAP SISTEM KLASIFIKASI PENYAKIT
PADA RETINA MATA MANUSIA MENGGUNAKAN MODEL
MOBILENETV3 BERBASIS WEBSITE**

SKRIPSI

**Diajukan untuk melengkapi tugas dan memenuhi syarat memperoleh ijazah
Sarjana Ilmu Komputer**

CHALIL AL VAREEL

201401043



**PROGRAM STUDI S-1 ILMU KOMPUTER
FAKULTAS ILMU KOMPUTER DAN TEKNOLOGI INFORMASI
UNIVERSITAS SUMATERA UTARA
2024**

PERSETUJUAN

Judul : IMPLEMENTASI CNN TERHADAP SISTEM
 KLASIFIKASI PENYAKIT PADA RETINA MATA
 MANUSIA MENGGUNAKAN MODEL MOBILENETV3
 BERBASIS WEBSITE

Kategori : SKRIPSI

Nama : CHALIL AL VAREEL

Nomor Induk Mahasiswa : 201401043

Program Studi : SARJANA (S-1) ILMU KOMPUTER

Fakultas : ILMU KOMPUTER DAN TEKNOLOGI INFORMASI
 UNIVERSITAS SUMATERA UTARA

12 Juni 2024

Komisi Pembimbing :

Dosen Pembimbing II

Anandhini Medianty Nababan, S.Kom., M.T.
 NIP. 199304132021022001

Dosen Pembimbing I

Fanindia Purnamasari, S.TI., M.IT.
 NIP. 198908172019032023

Diketahui/disetujui oleh
Program Studi S1 Ilmu Komputer



Dr. Amalia S.T., M.T.
 NIP. 197812212014042001

PERNYATAAN**IMPLEMENTASI CNN TERHADAP SISTEM KLASIFIKASI PENYAKIT
PADA RETINA MATA MANUSIA MENGGUNAKAN MODEL
MOBILENETV3 BERBASIS WEBSITE****SKRIPSI**

Saya mengakui bahwa skripsi ini adalah hasil penelitian saya sendiri, kecuali beberapa kutipan dan ringkasan yang masing-masing telah dicantumkan sumbernya.

Medan, 7 Mei 2024



Chalil Al Vareef
201401043

PENGHARGAAN

Segala puji bagi Allah yang senantiasa melimpahkan keberkahan, rahmat, dan hidayahNya kepada penulis sehingga mampu menyelesaikan skripsi ini sebagai syarat untuk memperoleh gelar Sarjana Komputer dari Program Studi S1 Ilmu Komputer Fakultas Ilmu Komputer dan Teknologi Informasi Universitas Sumatera Utara. Shalawat beriringan salam kepada baginda Rasulullah, Nabi Muhammad Shallallahu ‘alaihi Wasallam, karena dengan syafaat beliau-lah kita bisa merasakan nikmat ilmu pengetahuan yang diridho Allah hingga saat ini.

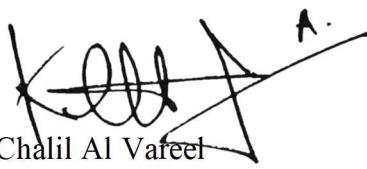
Penulis juga ingin menyampaikan rasa hormat dan terima kasih kepada:

1. Ibunda penulis Mira Suryani dan Ayahanda Faisal, A.Md.Kom. yang senantiasa memberikan motivasi, dukungan, doa, saran dan kasih sayang penuh selama menyelesaikan pendidikan.
2. Ibu Dr. Maya Silvi Lydia, B.Sc., M.Sc. selaku Dekan Fakultas Ilmu Komputer dan Teknologi Informasi Universitas Sumatera Utara.
3. Ibu Dr. Amalia S.T., M.T. selaku Ketua Program Studi S1 Ilmu Komputer Fakultas Ilmu Komputer Universitas Sumatera Utara.
4. Ibu Sri Melvani Hardi S.Kom., M.Kom. selaku Sekretaris Program Studi S1 Ilmu Komputer Universitas Sumatera Utara.
5. Ibu Fanindia Purnamasari, S.TI., M.IT. selaku dosen pembimbing I yang telah memberikan bimbingan, motivasi, kritik, dan saran serta arahan menuju kebaikan kepada penulis selama penyusunan skripsi ini.
6. Ibu Anandhini Medianty Nababan, S. Kom., M.T. selaku dosen pembimbing II yang telah memberikan bimbingan, motivasi, kritik, dan saran kepada penulis selama penyusunan skripsi ini.
7. Seluruh Bapak dan Ibu Dosen Program Studi S-1 Ilmu Komputer yang telah memberikan waktu dan pemikiran untuk mengajar dan membimbing sehingga dapat sampai kepada tahap penyusunan skripsi ini
8. Adik kandung tercinta Habil Al Variqi, Asyaqi Al Kaliffi, dan Abidzar Al Ghifari yang selalu mendukung serta mendoakan penulis dalam menjalankan aktivitas kuliah hingga akhirnya menyelesaikan tugas akhir.

9. Saudara Naynuunis Zulfachri S.Ked. yang telah mengizinkan penulis mengumpulkan informasi yang dibutuhkan dan memvalidasi hasil penelitian yang dilakukan oleh penulis.
10. Sahabat penulis Sastra Harapan Gulo, Erick Yudha Pratama Sukku dan Muhammad Iqbal Aldeena yang selalu memberikan dukungan dan masukan positif kepada penulis.
11. Teman-teman perkuliahan penulis Rezha, Wilson, Ariel, Andrew, Ewaldo, Rheza, Nico, Wilbert, Teguh, John, Rio Fransiskus, Holiness, Devi, Jessica, Ariyan, Salsabila, Yunisa, Reynold, Jeremi, Wana, Rio Hutabarat, Mustafa Kamal dan Afdoni yang memberikan penulis semangat untuk menyelesaikan skripsi.
12. Seluruh teman-teman di Ilmu Komputer Universitas Sumatera Utara yang telah memberikan motivasi dan nasihat.
13. Dan seluruh pihak yang membantu dan menyemangati yang tidak dapat penulis sebutkan satu per satu.

Semoga Allah selalu memberikan rahmat menemani langkah serta diberikan keberkahan kepada seluruh pihak yang memberikan dukungan, arahan dan tindakan selama menyelesaikan skripsi ini. Semoga skripsi ini dapat memberikan kontibusi positif bagi pribadi, keluarga, masyarakat, dan negara.

Medan, 7 Mei 2024



Chalil Al Vareef
201401043

ABSTRAK

Mata merupakan salah satu organ yang penting bagi manusia dimana mata dapat memperoleh informasi hampir sebanyak 80% hanya dengan melalui indra penglihatan. Jika mata mengalami masalah pada fungsinya, maka mata tidak dapat melihat objek yang berada di depannya dengan jelas maupun sempurna. Hal ini tentu menjadi masalah bagi manusia karena mata merupakan organ yang sangat krusial dalam melakukan aktivitas apapun sehari-hari. Apabila penyakit pada mata tidak diidentifikasi dan ditangani lebih dini, maka kondisi mata bisa saja menjadi lebih buruk bahkan dapat menyebabkan kebutaan. Oleh karena itu, diperlukan cara agar pengidentifikasi pada mata dapat dilakukan dengan lebih cepat dan akurat, penyakit mata dapat diidentifikasi melalui fundus retina mata. Salah satu solusi yang dapat digunakan adalah teknologi *Machine Learning* dengan mengandalkan algoritma *Convolutional Neural Network (CNN)* dari disiplin ilmu *Deep Learning*. CNN merupakan salah satu algoritma dalam konsep *Machine Learning* yang dapat mengolah data yang kompleks seperti gambar. Dengan perkembangan dari ilmu *Machine Learning*, CNN terus dikembangkan sehingga menghasilkan kumpulan-kumpulan model yang lebih cepat yang sudah dilatih dengan sebutan *pre-trained model*. Salah satu bentuk *pre-trained model* adalah *MobileNetV3* yang merupakan versi terbaru dari versi model sebelumnya yaitu *MobileNetV2*. CNN bersama *MobileNetV3* akan digunakan untuk mengidentifikasi jenis kondisi yang dimiliki oleh mata melalui fundus yang diberikan. Hasil akurasi yang didapatkan adalah 98% untuk data latih dan 90% untuk data validasi, implementasi akan digunakan dalam sebuah aplikasi berbasis *website*.

Kata Kunci: *Deep Learning, Convolutional Neural Network, MobileNetV3, Retina Mata*

CNN IMPLEMENTATION ON HUMAN RETINAL DISEASE CLASSIFICATION USING A WEB-BASED MOBILENETV3 MODEL

ABSTRACT

Eye is one of the most important organs for humans where the eye can obtain almost 80% of information only through the sense of sight, if the eye experiences problems in its function, then the eye cannot see the object in front of it clearly or perfectly. This will be a problem for humans because the eye is a very crucial organ in carrying out any daily activities. If eye diseases are not identified and treated early, the condition of the eyes may become worse and may even cause blindness. Therefore, a way is needed so that identification of the eye can be done more quickly and accurately, eye diseases can be identified through the retinal fundus of the eye. One solution that can be used is Machine Learning technology by relying on the Convolutional Neural Network (CNN) algorithm from the Deep Learning discipline. CNN is one of the algorithms in Machine Learning concept that can process complex data such as images. With the development of Machine Learning science, CNN continues to be developed so as to produce a collection of faster models that have been trained as pre-trained models. One form of pre-trained model is MobileNetV3 which is the latest version of the previous model version, MobileNetV2. CNN along with MobileNetV3 will be used to identify the type of condition that the eye has through the fundus given. The accuracy results obtained are 98% for training data and 90% for validation data, the implementation will be used in a web-based application.

Keywords: Deep Learning, Convolutional Neural Network, MobileNetV3, Retina.

DAFTAR ISI

PERSETUJUAN	ii
PERNYATAAN.....	iii
PENGHARGAAN.....	iv
ABSTRAK	vi
ABSTRACT.....	vii
DAFTAR ISI.....	viii
DAFTAR TABEL	x
DAFTAR GAMBAR.....	xi
DAFTAR LAMPIRAN.....	xii
BAB I PENDAHULUAN.....	1
1.1. Latar Belakang	1
1.2. Rumusan Masalah	3
1.3. Batasan Masalah.....	3
1.4. Tujuan Penelitian.....	3
1.5. Manfaat Penelitian.....	4
1.6. Metodologi Penelitian	4
1.7. Sistematika Penulisan.....	6
BAB II LANDASAN TEORI	7
2.1. Machine Learning	7
2.2. Deep Learning	8
2.3. Convolutional Neural Network (CNN)	8
2.4. <i>MobileNetV3</i>	9
2.3.1. Perbedaan <i>MobileNetV3-Small</i> dan <i>MobileNetV3-Large</i>	10
2.4. Retina Mata	12
2.4.1. Kondisi yang dapat diidentifikasi dari retina mata	14
2.5. Penggunaan Teknologi dalam Deteksi Penyakit	15
2.6. Penelitian Relevan	16
BAB III ANALISIS DAN PERANCANGAN SISTEM.....	18
3.1. Analisis Sistem.....	18
3.1.1. Analisis Masalah.....	18

3.1.2. Analisis Kebutuhan	20
3.2. Perancangan Sistem.....	21
3.3. Pemodelan Sistem	24
3.2.1. Diagram Umum Sistem.....	24
3.2.2. Use Case Diagram.....	25
3.2.3. Activity Diagram	26
3.2.4. Sequence Diagram	27
3.3. Flowchart.....	28
3.3.1. Flowchart Sistem	29
3.3.2. Flowchart Algoritma Convolutional Neural Network (CNN)	30
3.3.3. Flowchart arsitektur MobileNetV3	31
3.4. Perancangan <i>Interface</i>	32
3.4.1. Halaman Prediksi dengan Gambar.....	32
BAB IV IMPLEMENTASI DAN PENGUJIAN SISTEM.....	33
4.1. Implementasi	33
4.1.1. Landing Page	33
4.1.2. Halaman Input.....	34
4.1.3. Halaman Hasil Prediksi.....	34
4.2. Sumber Data.....	35
4.3. Pengujian Sistem.....	35
4.4. Hasil Pengujian Sistem.....	40
4.5. <i>User Acceptance Testing</i>	41
4.6. Faktor-Faktor Yang Memengaruhi Algoritma CNN.....	42
4.7. Faktor-Faktor Yang Memengaruhi Arsitektur <i>MobileNetV3</i>	43
BAB V PENUTUP.....	44
5.1. Kesimpulan.....	44
5.2. Saran.....	44
DAFTAR PUSTAKA.....	46
LAMPIRAN.....	A-1

DAFTAR TABEL

Tabel 3. 1 Augmentasi Data.....	22
Tabel 3. 2 Inisiasi MobileNetV3	22
Tabel 3. 3 Fungsi untuk Pelatihan Model	23
Tabel 4. 1 Daftar Gambar Fundus Retina mata.....	38
Tabel 4. 2 Evaluation Metrics	36
Tabel 4. 3 Nilai True Positive	38
Tabel 4. 4 Daftar Hasil Percobaan model	40
Tabel 4. 5 User Acceptance Testing.....	41

DAFTAR GAMBAR

Gambar 1. 1	Arsitektur Umum.....	5
Gambar 2. 1	Ilustrasi Machine Learning.....	7
Gambar 2. 2	Arsitektur CNN	9
Gambar 2. 3	Perbedaan Arsitektur MobileNetV3 dengan Versi Sebelumnya	10
Gambar 2. 4	Lapisan (a) MobileNetV3-Large dan (b) MobileNetV3-Small.....	11
Gambar 2. 5	Latensi MobileNetV3-Small, MobileNetV3-Large, dan MobileNetV2	12
Gambar 2. 6	Retina Mata.....	13
Gambar 2. 7	Retina Mata Normal	14
Gambar 2. 8	Retina Mata Glaucoma	14
Gambar 2. 9	Retina Mata Diabetic Retinopathy	15
Gambar 2. 10	Retina Mata Cataract	15
Gambar 3. 1	Diagram Umum Sistem	25
Gambar 3. 2	Use Case Diagram	26
Gambar 3. 3	Activity Diagram	27
Gambar 3. 4	Sequence Diagram.....	28
Gambar 3. 5	Flowchart Sistem	29
Gambar 3. 6	Flowchart Algoritma CNN	30
Gambar 3. 7	Flowchart Arsitektur MobileNetV3	31
Gambar 3. 8	Halaman Prediksi Dengan Gambar	32
Gambar 4. 1	Landing Page	33
Gambar 4. 2	Halaman Prediksi.....	34
Gambar 4. 3	Halaman Hasil Prediksi	34
Gambar 4. 4	Citra fundus retina mata	35
Gambar 4. 5	Grafik Training dan Validation Accuracy	36
Gambar 4. 6	Confusion Matrix.....	37
Gambar 4. 7	Percobaan pada sistem website	39

DAFTAR LAMPIRAN

Lampiran 1. LISTING PROGRAM.....	A-1
Lampiran 2. USER ACCEPTANCE TESTING	B-1

BAB I

PENDAHULUAN

1.1. Latar Belakang

Mata merupakan salah satu organ manusia yang penting dimana mata merupakan salah satu dari lima panca indra manusia. Dengan mata manusia dapat memperoleh informasi hampir sebanyak 80% sehingga mata merupakan organ yang sangat krusial bagi manusia (Indraswari et al, 2022). Pada tahun 2020, terdapat 35 juta masyarakat Indonesia mengalami penurunan pada kemampuan mata dan 3,7 juta diantaranya mengalami kebutaan. Dari jumlah tersebut, 30% diantaranya merupakan anak-anak dan remaja sehingga pencegahan masalah pada mata harus dilakukan lebih dini (Sumber Website *The International Agency for the Prevention of Blindness*).

Masalah pada mata dimulai dari identifikasi pada bagian retina mata, retina adalah bagian lembaran yang transparan dan tipis yang terletak pada bagian belakang bola mata. Retina bersifat sensitif akan cahaya. Retina akan bertanggung jawab terhadap kemampuan penglihatan pada manusia di mana retina akan menangkap bayangan benda dari lensa. Ketika retina rusak atau terkena penyakit, manusia akan sulit melihat dengan baik bahkan dapat menyebabkan kebutaan (Lusiani, 2019).

Penyakit pada bagian retina mata dapat dideteksi melalui citra fundus yang diperoleh menggunakan *retinal camera* ataupun *ophthalmoscope*. Citra fundus digunakan dengan alasan memakan biaya yang cukup rendah dan bersifat non-invasif sehingga cocok digunakan untuk melakukan deteksi penyakit pada mata (Indraswari et al, 2022).

Hingga saat ini, idenifikasi penyakit mata masih menggunakan metode *Computer-aided Diagnosis* (CAD) yang masih bersifat konvensional dalam melakukan segmentasi objek-objek pada citra fundus. Metode ini memiliki kekurangan dimana perbedaan seperti letak abnormal, warna hingga ukuran yang berbeda harus disegmentasikan secara manual meskipun memiliki hasil diagnosis yang sama. Hal ini

akan membuat identifikasi menghabiskan waktu yang lebih lama sehingga diperlukan sistem yang dapat mengidentifikasi penyakit tanpa harus menerapkan segmentasi yang seharusnya tidak perlu dilakukan (Indraswari et al, 2022). Di sisi lain, pengembangan ilmu *Machine Learning* telah menghasilkan metode yang dapat mempelajari fitur data tersebut sekaligus menghilangkan dependensi terhadap segmentasi manual. Metode disebut dikenal sebagai *Convolutional Neural Network* (CNN).

Convolutional Neural Network (CNN) merupakan cabang dari implementasi *Machine Learning* dalam menganalisis fitur pada citra (Craig, 2024). Algoritma CNN sangat popular dalam disiplin ilmu Deep Learning karena kemampuan ekstraksi fitur yang mampu dilatih untuk dapat mengenal objek baru (Omori, 2020).

CNN dapat dilakukan dengan menggunakan model-model yang telah dibangun dan dilatih sebelumnya sehingga memiliki akurasi yang lebih baik dan bekerja dalam waktu yang lebih singkat dibandingkan dengan model yang dibangun dari awal. Model ini disebut sebagai *pre-trained model* (Budu, 2023).

Salah satu bentuk pre-trained model adalah *MobileNetV3*, *MobileNetV3* merupakan versi ketiga dari model *MobileNet* yang digunakan sebagai algoritma pencarian otomatis dengan akurasi tinggi dan latensi rendah untuk mendeteksi data visual. *MobileNetV3* berhasil mengurangi latensi sambil mempertahankan akurasi, memperkenalkan decoder segmentasi ringan yang terbaru, dan melakukan eksplorasi dampak non-linearitas dan komponen lain pada kinerjanya. Berdasarkan penelitian yang dilakukan oleh Howard et al (2019), *MobileNetV3* mengungguli versi model sebelumnya dalam hal akurasi dan latensi.

Salah satu penelitian tentang penggunaan CNN dengan menggunakan *pre-trained model* adalah penggunaan model *VGG-16* dalam sistem klasifikasi penyakit mata yang dilakukan oleh William dan Chairisni Lubis (2022) yang mengungkapkan bahwa citra retina mata dapat diklasifikasikan dengan akurasi sebesar 91% pada data latih. Hal ini menunjukkan bahwa *pre-trained model* layak digunakan untuk klasifikasi citra retina mata. Oleh karena itu, Peneliti bertujuan untuk mengimplementasikan model *MobileNetV3* terhadap data citra fundus retina mata guna melihat akurasi dan kelayakan dalam penggunaan model ini pada citra fundus retina mata.

Sistem klasifikasi penyakit pada retina mata manusia berbasis web dinilai mampu mendeteksi kondisi pada retina apakah retina mata terdapat penyakit atau tidak dengan mengunggah citra fundus sebagai data input (Indraswari et al, 2022). Data tersebut diproses menggunakan pendekatan CNN dengan model *MobileNetV3* untuk menghasilkan data keluaran yaitu kondisi retina mata. Kondisi mata yang akan dikeluarkan sebagai output adalah *cataract*, *diabetic retinopathy*, *glaucoma*, dan normal.

1.2. Rumusan Masalah

Penggunaan sistem CAD membuat identifikasi penyakit pada mata menjadi lebih sulit dan memakan waktu yang lama. Oleh karena itu, sistem klasifikasi kondisi retina mata berbasis web diperlukan untuk mengidentifikasi secara dini guna menghindari mata dari masalah yang lebih serius.

1.3. Batasan Masalah

Batasan masalah yang digunakan pada penelitian ini adalah:

1. Metode yang digunakan adalah Convolutional Neural Network (CNN) dengan menggunakan model *MobileNetV3*.
2. Data yang digunakan adalah fundus mata dengan format *jpg*, *jpeg*, dan *png* dari hasil *fundoscopy* yang dikumpulkan dari situs dataset *Kaggle*.
3. Jumlah gambar yang diperlukan adalah 4180 gambar dengan 2926 gambar sebagai data latih (*training data*) dan 1254 gambar sebagai data validasi (*validation data*).
4. Hasil klasifikasi yang akan diidentifikasi ada 4, yaitu normal, *cataract*, *glaucoma*, dan *diabetic retinopathy*.

1.4. Tujuan Penelitian

Tujuan penelitian ini yaitu menerapkan CNN dan arsitektur *MobileNeV3* yang dapat mengklasifikasikan kondisi mata pasien berdasarkan citra fundus mata yang diberikan untuk mengidentifikasi masalah mata pasien dengan lebih akurat.

1.5. Manfaat Penelitian

Manfaat yang akan diperoleh melalui penelitian ini adalah:

1. Penggunaan sistem website ini akan mengurangi dampak *human error* yang dihasilkan dari kamera fundus.
2. Identifikasi jenis penyakit mata yang dialami oleh pasien dapat dilakukan dengan lebih dini sehingga dapat menentukan penanganan lebih lanjut dengan lebih cepat.
3. Hasil penelitian yang diperoleh dapat digunakan sebagai rujukan pengembangan sistem klasifikasi CNN terhadap citra fundus mata dengan menggunakan model *MobileNetV3*.

1.6. Metodologi Penelitian

Metode penelitian yang akan digunakan dalam penelitian ini adalah sebagai berikut:

1. Studi Pustaka

Penelitian dimulai dengan menemukan referensi yang berasal dari berbagai sumber terpercaya seperti buku-buku, jurnal, *e-book*, artikel ilmiah, makalah ataupun skripsi yang berhubungan dengan system identifikasi penyakit pada retina mata.

2. Pengumpulan data

Tahap pengumpulan adalah tahap dimana penulis mengumpulkan data yang diperlukan, dara tersebut adalah gambar dari setiap hasil prediksi yang ingin ditentukan.

3. Analisis dan Perancangan

Tahap dimana penulis melakukan analisis terhadap kebutuhan penlitian dan hasilnya dirancang dalam sebuah diagram alir (*flowchart*).

4. Implementasi

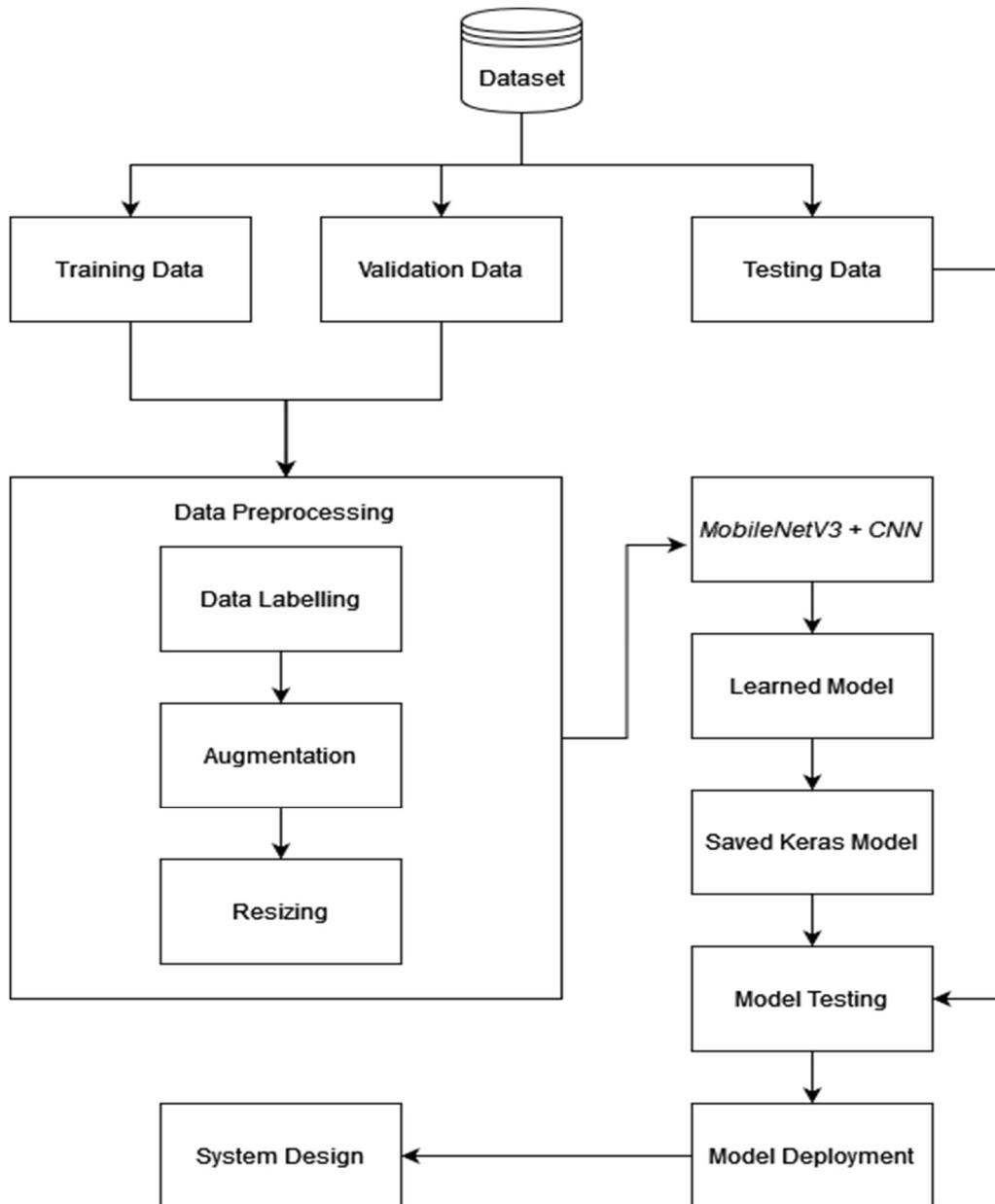
Tahap ini akan menjadi tahap pembuatan sistem (*coding*) sesuai dengan *flowchart* yang telah dirancang, sistem yang dirancang adalah model dan *website*.

5. Pengujian

Sistem yang telah dirancang akan diuji coba dengan mengunggah data yang baru untuk melihat apakah model dapat memprediksi data yang diberikan.

6. Arsitektur umum

Berikut merupakan arsitektur sederhana yang akan digunakan:



Gambar 1. 1 Arsitektur Umum

7. Dokumentasi

Penelitian yang dilakukan akan didokumentasikan dalam bentuk skripsi mulai dari bagian analisa sampai kepada tahap pengujian.

1.7. Sistematika Penulisan

Sistematika penulisan terdiri dari lima bab, yakni:

BAB I PENDAHULUAN

Semua yang berkaitan mengenai latar belakang dari permasalahan yang akan diteliti yakni tujuan penelitian, rumusan masalah, metode penelitian, batasan masalah, manfaat penelitian, dan sistematika penulisan akan dibahas pada bab ini.

BAB II LANDASAN TEORI

Tinjauan pustaka yang berhubungan dengan *Convolutional Neural Network* dan *MobileNetV3* akan dijelaskan pada bab ini.

BAB III ANALISIS PERANCANGAN

Analisis masalah dan sistem yang dibangun akan dibahas pada bab ini. Selanjutnya merancang sistem dengan menggunakan Algoritma CNN dan Model *MobileNetV3*.

BAB IV IMPLEMENTASI DAN PENGUJIAN SISTEM

Membahas tentang Implementasi dan pengujian sistem berdasarkan tahapan analisis dan perancangan yang telah disebutkan di atas.

BAB V PENUTUP

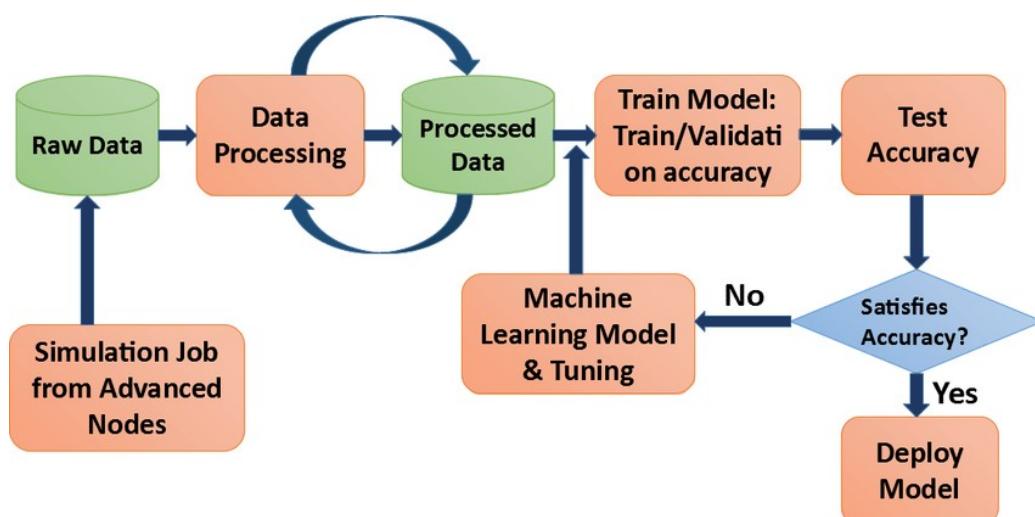
Membahas hasil esimpulan dan saran yang didapatkan dari hasil penelitian yang dilakukan akan dibahas pada bab ini.

BAB II

LANDASAN TEORI

2.1. Machine Learning

Machine Learning adalah disiplin ilmu dalam ilmu komputer yang berkaitan dengan pengembangan algoritma dan model statistik yang memungkinkan komputer untuk memperoleh pengetahuan atau keterampilan dari data yang ada. Beberapa disiplin ilmu seperti matematika, statistika, dan komputasi digunakan untuk merancang algoritma yang mampu mengidentifikasi pola-pola dalam data, mengekstraksi fitur, membuat prediksi, dan menghasilkan keputusan tanpa harus diprogram secara langsung untuk setiap tugas yang dihadapi. (Bi et al., 2019).



Gambar 2. 1 Ilustrasi Machine Learning
 (Sumber: Yogi, 2020)

Tujuan utama dari *machine learning* adalah memberikan kemampuan pada sistem komputer untuk mempelajari dan mengasimilasi informasi dari data yang diberikan dalam bentuk model. Dengan pendekatan ini, komputer akan belajar dan beradaptasi dengan data yang berubah, serta meningkatkan efisiensinya dalam

menyelesaikan tugas yang diberikan. Metode yang digunakan dalam *machine learning* meliputi pengenalan pola, analisis regresi, dan klasifikasi. *machine learning* memungkinkan sistem komputer untuk mengekstraksi fitur yang berharga dari data yang kompleks dan besar. Dengan kemampuan ini, *machine learning* memiliki aplikasi yang luas seperti sistem pengenalan wajah, sistem rekomendasi, analisis pasar keuangan, dan masih banyak lagi.

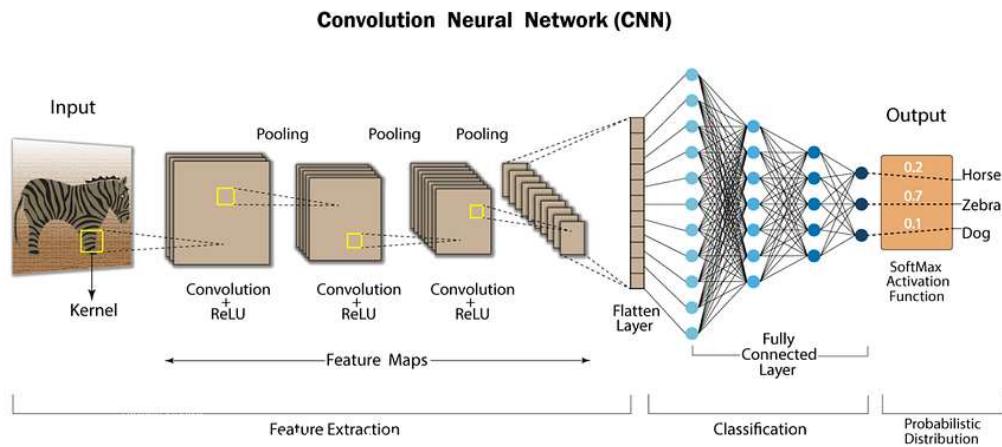
2.2. Deep Learning

Deep Learning adalah subdisiplin ilmu dalam *machine learning* yang digunakan untuk mengekstraksi fitur dari data yang lebih kompleks di mana komputasi *deep learning* dilakukan melalui jaringan saraf berlapis-lapis (*multi-layered neural networks*), *deep learning* terinspirasi oleh pola konektivitas dari otak manusia (khususnya korteks visual) yang memainkan peran penting dalam memahami dan memproses rangsangan visual (Sarker, 2021). *Deep learning* memiliki kemampuan untuk mengekstrak fitur-fitur dari data yang sangat bermanfaat, untuk mendeteksi penyakit dengan arsitektur yang berlapis dan jumlah saraf yang lebih banyak (Horry et al., 2020).

Tidak seperti *machine learning* yang memerlukan data yang lebih sedikit dan sederhana, *Deep learning* memerlukan data yang berbentuk kompleks dan jumlah yang lebih banyak untuk membangun sebuah model. Oleh karena itu, model *deep learning* dalam pembangunannya membutuhkan waktu yang lebih banyak dengan kemampuan komputasi yang lebih mumpuni dibandingkan dengan model *machine learning* yang lebih sederhana.

2.3. Convolutional Neural Network (CNN)

Convolutional Neural Network (CNN) adalah sebuah kategori model *deep learning* yang cocok untuk menganalisis data visual dengan cara mengekstraksi fitur dan mengidentifikasi pola gambar (Craig, 2023). Saraf buatan dalam CNN disusun untuk menginterpretasikan informasi visual secara efisien, sehingga memungkinkan model ini memproses data gambar.



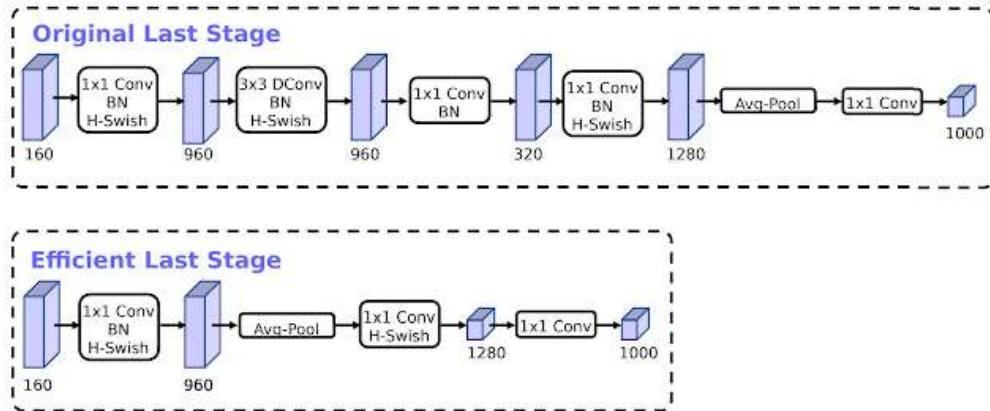
Gambar 2. 2 Arsitektur CNN
 (Sumber: Shahriar, 2023)

Algoritma CNN menggunakan operasi konvolusi untuk mengidentifikasi pola data yang dinilai kompleks dan terus meningkatkan kemampuannya untuk menghasilkan model yang akurat. CNN terdiri dari beberapa lapisan (*layer*) yang umumnya dibedakan menjadi 3 bagian, yaitu lapisan untuk masukan (*input layer*), lapisan tengah atau tersembunyi (*hidden layer*), dan lapisan untuk keluaran (*output layer*). Beberapa jenis lapisan juga digunakan untuk membantu model mengekstraksi fitur secara lebih baik pada data seperti *pooling*, normalisasi, *dropout*, dan sebagainya.

Meskipun CNN sebagian besar digunakan untuk memproses gambar, CNN juga dapat diadaptasi untuk bekerja dengan audio dan data sinyal lainnya (Craig, 2023).

2.4. *MobileNetV3*

MobileNetV3 adalah versi ketiga dari arsitektur *MobileNet* yang mendukung kemampuan analisis gambar dari banyak aplikasi seluler populer. Arsitektur ini juga telah digabungkan dalam framework populer seperti *TensorFlow Lite*. Google secara teratur telah merilis pembaruan pada arsitektur MobileNet yang menggabungkan beberapa ide paling baru di bidang deep learning (Rodriguez, 2019).



Gambar 2. 3 Perbedaan Arsitektur MobileNetV3 dengan Versi Sebelumnya
 (Sumber: Howard, 2019)

Perkembangan yang menarik dari *MobileNetV3* adalah hasil desain ulang dari beberapa lapisan dalam arsitektur. Beberapa lapisan pada *MobileNetV2* dihilangkan karena menimbulkan masalah pada tingkat latensi meskipun dianggap sangat penting bagi akurasi mode. Dengan menggabungkan beberapa optimasi dasar, *MobileNetV3* mampu menghapus tiga lapisan tersebut dari arsitektur pendahulunya tanpa mengurangi akurasi dengan latensinya.

2.3.1. Perbedaan *MobileNetV3-Small* dan *MobileNetV3-Large*

MobileNetV3 terdiri dari dua jenis model, yaitu *MobileNetV3-Small* dan *MobileNetV3-Large*. Meskipun keduanya merupakan versi lanjutan dari *MobileNetV2*, kedua model ini memiliki beberapa perbedaan yang dalam kinerjanya, perbedaan tersebut adalah:

1. Struktur lapisan

Lapisan pada *MobileNetV3-Large* memiliki beberapa lapisan *bottleneck* yang lebih banyak dibandingkan dengan *MobileNetV3-Small* sehingga *MobileNetV3-Large* dapat digunakan terhadap data yang lebih kompleks atau besar apabila penggunaan *MobileNetV3-Small* dinilai masih belum sempurna. Perbedaan struktur lapisan kedua model terdapat pada Gambar 2.4 di bawah ini.

Input	Operator	exp size	#out	SE	NL	s	Input	Operator	exp size	#out	SE	NL	s
$224^2 \times 3$	conv2d	-	16	-	HS	2	$224^2 \times 3$	conv2d, 3x3	-	16	-	HS	2
$112^2 \times 16$	bneck, 3x3	16	16	-	RE	1	$112^2 \times 16$	bneck, 3x3	16	16	✓	RE	2
$112^2 \times 16$	bneck, 3x3	64	24	-	RE	2	$56^2 \times 16$	bneck, 3x3	72	24	-	RE	2
$56^2 \times 24$	bneck, 3x3	72	24	-	RE	1	$28^2 \times 24$	bneck, 3x3	88	24	-	RE	1
$56^2 \times 24$	bneck, 5x5	72	40	✓	RE	2	$28^2 \times 24$	bneck, 5x5	96	40	✓	HS	2
$28^2 \times 40$	bneck, 5x5	120	40	✓	RE	1	$14^2 \times 40$	bneck, 5x5	240	40	✓	HS	1
$28^2 \times 40$	bneck, 5x5	120	40	✓	RE	1	$14^2 \times 40$	bneck, 5x5	240	40	✓	HS	1
$28^2 \times 40$	bneck, 3x3	240	80	-	HS	2	$14^2 \times 40$	bneck, 5x5	120	48	✓	HS	1
$14^2 \times 80$	bneck, 3x3	200	80	-	HS	1	$14^2 \times 48$	bneck, 5x5	144	48	✓	HS	1
$14^2 \times 80$	bneck, 3x3	184	80	-	HS	1	$14^2 \times 48$	bneck, 5x5	288	96	✓	HS	2
$14^2 \times 80$	bneck, 3x3	184	80	-	HS	1	$7^2 \times 96$	bneck, 5x5	576	96	✓	HS	1
$14^2 \times 80$	bneck, 3x3	480	112	✓	HS	1	$7^2 \times 96$	bneck, 5x5	576	96	✓	HS	1
$14^2 \times 112$	bneck, 3x3	672	112	✓	HS	1	$7^2 \times 96$	conv2d, 1x1	-	576	✓	HS	1
$14^2 \times 112$	bneck, 5x5	672	160	✓	HS	2	$7^2 \times 576$	pool, 7x7	-	-	-	-	1
$7^2 \times 160$	bneck, 5x5	960	160	✓	HS	1	$1^2 \times 576$	conv2d 1x1, NBN	-	1024	-	HS	1
$7^2 \times 160$	bneck, 5x5	960	160	✓	HS	1	$1^2 \times 576$	conv2d 1x1, NBN	-	k	-	-	1
$7^2 \times 160$	conv2d, 1x1	-	960	-	HS	1	$1^2 \times 1024$	conv2d 1x1, NBN	-	k	-	-	1
$7^2 \times 960$	pool, 7x7	-	-	-	1								
$1^2 \times 960$	conv2d 1x1, NBN	-	1280	-	HS	1							
$1^2 \times 1280$	conv2d 1x1, NBN	-	k	-	1								

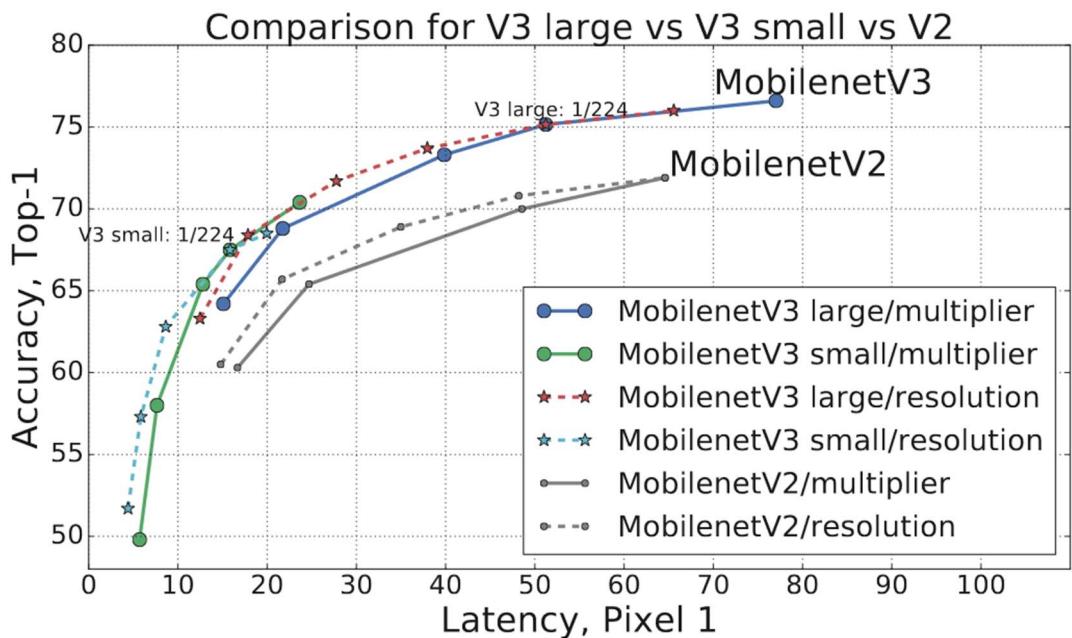
(a)

(b)

Gambar 2. 4 Lapisan (a) MobileNetV3-Large dan (b) MobileNetV3-Small
(Sumber: Howard, 2019)

2. Latensi

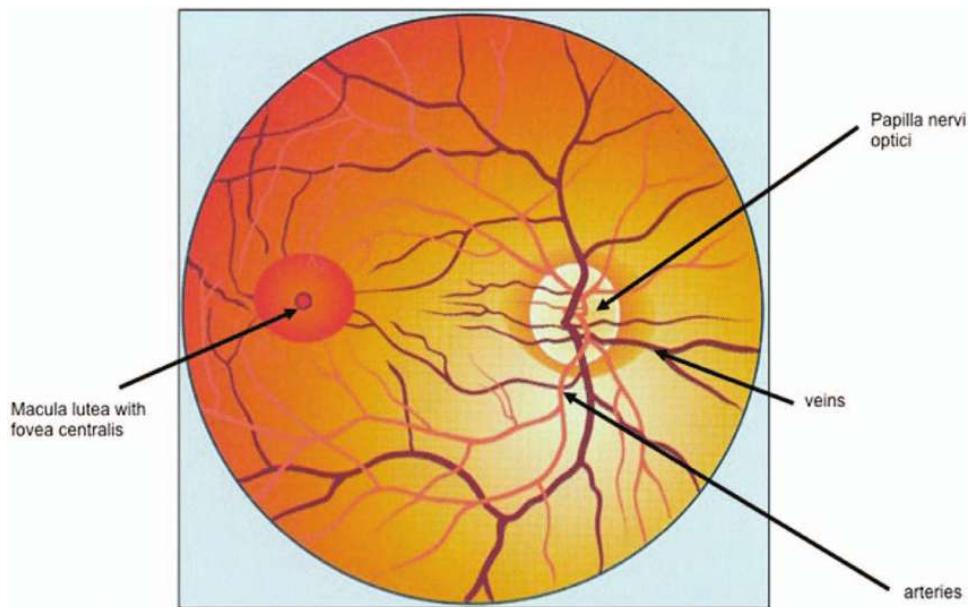
Latensi merupakan jumlah waktu yang diperlukan oleh sebuah arsitektur untuk menghasilkan prediksi untuk sebuah sampel masukan (Elbrag, 2023). Meskipun *MobileNetV3-Large* dikenal dengan kemampuan memprediksi input dengan akurasi yang lebih tinggi, jumlah lapisan yang lebih banyak dan kompleks menyebabkan latensi pada *MobileNetV3-Large* menjadi lebih tinggi daripada *MobileNetV3-Small*. Oleh karena itu, *MobileNetV3-Large* membutuhkan waktu yang sedikit lebih banyak dalam pemrosesan data dibandingkan dengan *MobileNetV3-Small*. Perbedaan latensi kedua model terdapat pada Gambar 2.5 di bawah ini.



Gambar 2. 5 Latensi MobileNetV3-Small, MobileNetV3-Large, dan MobileNetV2
(Sumber: Howard, 2019)

2.4. Retina Mata

Retina (*Tunika nervosa*) adalah lapisan yang bersifat peka cahaya yang berada dalam mata. Retina memiliki struktur yang kompleks dengan ketebalan yang hanya sekitar 0,5 mm, tetapi kaya akan jaringan saraf, pembuluh darah, dan sel-sel fotoreseptor yang memungkinkan kita untuk menangkap cahaya dan mengubahnya menjadi sinyal-sinyal yang bisa diproses oleh otak. Lapisan yang tipis ini terletak pada bagian belakang mata dan bertanggung jawab untuk menangkap cahaya. Apabila lapisan ini terganggu, maka akan mempengaruhi sistem penglihatan pada manusia (Noor, 2024).



Gambar 2. 6 Retina Mata
(Sumber: Hönenmann, 2015)

Gejala penyakit retina terjadi ada berbagai macam tergantung penyebabnya. Gejala penyakit retina yang dialami secara umum adalah sebagai berikut.

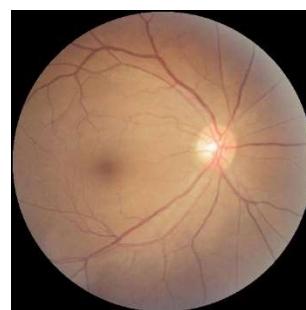
1. Luas pandangan terbatas: Penglihatan menjadi terbatas atau tidak mampu melihat objek yang terlalu jauh. Hal ini dapat mengganggu aktivitas seperti membaca, mengemudi, dan berkendara.
2. Pandangan kabur: Penglihatan terganggu, seperti pandangan yang kabur atau hilang sehingga individu tidak dapat melihat objek dengan jelas.
3. Melihat kilatan-kilatan cahaya atau photopsi: Penglihatan menjadi terganggu dengan kilatan-kilatan cahaya yang seharusnya tidak ada. Cahaya ini muncul dan menghilang secara tiba-tiba bahkan dalam keadaan gelap sekalipun.
4. Melihat bayangan benda yang berukuran kecil: Penglihatan terganggu dengan bayangan benda berukuran kecil, seperti bintik-bintik hitam kecil. Hal ini dapat membuat individu sulit untuk fokus pada objek tertentu.
5. Sensitif terhadap cahaya: Mata menjadi lebih sensitif terhadap cahaya seperti merasa silau atau tidak nyaman dengan cahaya yang lembut sekalipun. Individu

akan mencari perlindungan ke tempat yang memiliki cahaya yang lebih sedikit atau tidak terlalu terang.

6. Terganggunya kemampuan membedakan warna: Gangguan pada retina juga dapat menyebabkan kemampuan membedakan warna tertentu menjadi terganggu bahkan dapat kehilangan kemampuan untuk melihat warna secara keseluruhan.

2.4.1. Kondisi yang dapat diidentifikasi dari retina mata

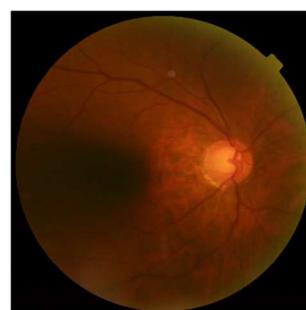
1. Normal



Gambar 2. 7 Retina Mata Normal

Pada Gambar 2.7, dapat dilihat bahwa retina mata yang normal memiliki pembuluh darah yang tampak jelas dan tidak ada tanda-tanda gangguan seperti pembengkakan ataupun bintik-bintik pada mata.

2. *Glaucoma*



Gambar 2. 8 Retina Mata *Glaucoma*

Pada Gambar 2.8, Retina mata yang mengalami *Glaucoma* akan terlihat berbeda dimana pembuluh darah pada bagian *optic disk* terlihat terjepit dan saraf optik tampak lebih pucat.

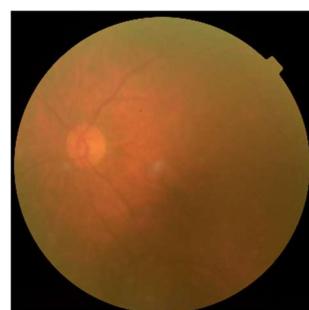
3. *Diabetic Retinopathy*



Gambar 2. 9 Retina Mata *Diabetic Retinopathy*

Pada Gambar 2.9, Retina mata yang mengalami *Diabetic Retinopathy* mengalami perubahan seperti pembengkakan, pendarahan, atau pertumbuhan abnormal pembuluh darah kecil yang disebut dengan neovaskularisasi. Hal ini disebabkan oleh penyakit diabetes.

4. *Cataract*



Gambar 2. 10 Retina Mata *Cataract*

Pada Gambar 2.10. Retina mata yang mengalami *Cataract* dapat dilihat dengan lensa mata yang keruh. Hal ini dapat menjelaskan alasan mengapa penglihatan pada penderita akan mengalami buram atau kabur.

2.5. Penggunaan Teknologi dalam Deteksi Penyakit

Penggunaan teknologi dalam deteksi penyakit merupakan metode yang mulai dikembangkan untuk mendapatkan hasil diagnosis dengan guna sebagai informasi

mengenai kondisi pasien. Harapan dari penggunaan teknologi di bidang ini ialah mendapatkan hasil diagnosis yang lebih cepat dan presisi dibandingkan dengan hanya mengandalkan kemampuan manusia.

Sebagai contoh, penggunaan *deep learning* dalam mendeteksi penyakit mata dikembangkan untuk mempermudah Ophthalmologist dalam mengidentifikasi penyakit mata pasien melalui data yang diperoleh, *deep learning* telah dikenal secara luas kemampuannya dalam mendapatkan hasil diagnosis terhadap penyakit umum pada mata yang mengancam kemampuan penglihatan seperti *glaucoma*, *age-related macular degeneration (AMD)*, *retinopathy of prematurity*, dan *diabetic retinopathy*.

Hingga saat ini, integrasi antara disiplin ilmu *deep learning* dengan praktisi Ophthalmologist dilakukan dengan harapan memberikan inovasi terbaru dalam meningkatkan kemampuannya dalam mengenali jenis penyakit pada mata secara lebih dini dan mendapatkan hasil prediksi yang lebih baik (Nuzzi et al, 2021).

2.6. Penelitian Relevan

Berikut adalah beberapa penelitian terdahulu yang berkaitan sistem pendekripsi penyakit pada retina mata manusia menggunakan model MobileNetV3 berbasis website:

1. Penelitian yang dilakukan oleh Rarasmaya Indraswari et al., (2022) yang berjudul Deteksi Penyakit Mata Pada Citra Fundus Menggunakan *Convolutional Neural Network (CNN)* diperoleh bahwa CNN mampu mendekripsi ketidaknormalan pada mata dengan akurasi sebesar 72% menggunakan MobileNetV2 tanpa perlu melakukan *fine tuning*. Kelas data yang digunakan adalah Normal dan Abnormal (kumpulan penyakit mata seperti *cataract*, *glaucoma*, dan *retinal disease*)
2. Penelitian yang dilakukan oleh Dewi Marcella et al., (2022) yang berjudul Klasifikasi Penyakit Mata Menggunakan *Convolutional Neural Network* Dengan Arsitektur *VGG-19* diperoleh bahwa model yang digunakan sebaiknya adalah model yang berbeda dengan tambahan preprocessing seperti grayscale, blood vessel, dan sebaginya. Kelas data yang digunakan adalah Normal, *Cataract*, *Glaucoma*, dan *Retinal Disease* dengan jumlah data sebanyak 601 citra.

3. Penelitian yang dilakukan oleh William dan Chairisni Lubis, (2022) dengan judul *Klasifikasi Penyakit Mata Menggunakan CNN* diperoleh Klasifikasi penyakit pada mata dapat dilakukan menggunakan arsitektur CNN dengan VGG16 sebagai alat untuk ekstraksi fitur citra fundus. Kelas data yang digunakan adalah Normal, Diabetik, Glaukoma, katarak, dan Hipertensi.
4. Penelitian yang dilakukan oleh Swasono et al (2019) dengan judul *Retinal Diseases Classification Using Levenberg-Marquath (LM) Learning Algorithm for Pi Sigma Network (PSN) and Principal Component Analysis (PCA) Methods* diperoleh bahwa studi ini berhasil mencapai akurasi dan validasi sebesar 100% dengan metode PCA. Kelas data yang digunakan adalah *Age-related Macular Disease* (AMD), *Diabetic Retinopathy* (DR), Normal, dan *Retinal Detachment* (RD) dengan jumlah total data sebanyak 60 citra.
5. Penelitian yang dilakukan oleh Putri Sarah Fransisca & Nurhafifah Matondang, (2023) dengan judul *Deteksi Citra Digital Penyakit Cacar Monyet menggunakan Algoritma Convolutional Neural Network dengan Arsitektur MobileNetV2* diperoleh bahwa model MobileNetV2 dengan teknik transfer learning mampu mendeteksi citra digital penyakit cacar monyet dengan akurasi yang tinggi, yaitu 94% untuk data latih dan 92% untuk data validasi. Kelas data yang digunakan adalah Normal, *Monkeypox*, dan *Measles* dengan jumlah total data sebanyak 135.

BAB III

ANALISIS DAN PERANCANGAN SISTEM

3.1. Analisis Sistem

Analisis sistem adalah salah satu tahapan dalam penelitian yang memiliki tujuan untuk menguraikan komponen-komponen yang dibutuhkan untuk dapat bekerja di dalam sistem. Analisis sistem dibagi ke dalam dua jenis yakni analisis masalah yang berguna untuk melakukan identifikasi penyebab dan dampak dari sebuah permasalahan dan analisis kebutuhan yang dilakukan untuk menguraikan data dan proses yang diperlukan dalam perancangan sebuah sistem.

3.1.1. Analisis Masalah

Dalam penelitian, analisis masalah bertujuan untuk mengetahui penyebab dari permasalahan kemudian dilakukan pengkajian lebih dalam pada masalah yang akan diselesaikan untuk menghasilkan sistem yang baik. Dalam penelitian ini, permasalahan yang dianalisis adalah bagaimana algoritma CNN dengan arsitektur *MobileNetV3* dapat mengidentifikasi penyakit pada retina mata.

Untuk mempermudah analisis masalah, sebuah metode yang dapat digunakan adalah metode *5-Whys*. Metode *5-Whys* adalah pendekatan yang digunakan dalam analisis masalah untuk menggali akar penyebab masalah dengan bertanya "Mengapa?" secara berulang. Metode ini bertujuan untuk menemukan penyebab mendasar dari suatu masalah dengan mengidentifikasi dan mengeksplorasi berbagai faktor yang berkontribusi terhadap masalah tersebut.

Berikut adalah beberapa pertanyaan yang digunakan untuk mengeksplorasi lebih lanjut masalah penelitian diatas:

1. Mengapa identifikasi penyakit pada retina mata penting?
- Identifikasi penyakit retina mata sangat penting karena retina mata merupakan bagian penting dari sistem visual manusia. Penyakit retina mata seperti retinopati

diabetik dapat menyebabkan kerusakan permanen pada penglihatan jika tidak diidentifikasi dan diobati dengan cepat. Dengan meningkatnya prevalensi penyakit mata akibat faktor gaya hidup dan faktor genetik, deteksi dini melalui teknologi seperti MobileNetV3 dapat membantu dalam mendiagnosis penyakit retina mata secara lebih efisien.

2. Mengapa *Convolutional Neural Network* (CNN) dipilih sebagai metode untuk identifikasi penyakit retina mata?
 - CNN adalah sebuah algoritma *Machine Learning* yang handal dalam hal pengolahan data dalam bentuk gambar dan mencari pola dari gambar-gambar yang diberikan. CNN memiliki proses ekstraksi gambar yang baik sehingga dapat melakukan pengolahan fitur-fitur yang terdapat pada sebuah gambar dan sesuai dengan masalah yang ingin diselesaikan yakni penyakit pada retina mata.
3. Mengapa *MobileNetV3* diintegrasikan dalam studi kasus ini untuk prediksi penyakit retina mata berbasis CNN?
 - Integrasi *MobileNetV3* dalam studi kasus ini dipilih karena kemampuannya dalam mengoptimalkan kinerja CNN dalam mengenali pola-pola kompleks pada gambar retina mata. *MobileNetV3* menawarkan keunggulan dalam efisiensi komputasi tanpa mengorbankan akurasi, sehingga cocok digunakan dalam aplikasi medis yang memerlukan prediksi cepat dan akurat.
4. Mengapa menggunakan *Flask* dalam sistem berbasis *website* untuk sistem klasifikasi penyakit pada retina mata?
 - *Flask* dipilih untuk sistem berbasis *website* karena kemudahan penggunaannya dalam membangun aplikasi web yang ringan dan skalabel. Dengan *Flask*, fitur-fitur seperti pengelolaan pengguna, visualisasi data, dan integrasi dapat dilakukan dengan lebih cepat dengan model klasifikasi penyakit retina mata. Selain itu, *Flask* juga mendukung pengembangan berbasis *Python* sehingga memudahkan penggunaan algoritma klasifikasi berbasis Python seperti *MobileNetV3*.

5. Mengapa diperlukan studi kasus untuk menguji pendekatan ini dalam situasi nyata?
 - Dengan adanya studi kasus, akan memberikan pemahaman lebih mendalam tentang CNN dan *MobileNetV3*. Hasil dari studi kasus ini diharapkan dapat menjadi referensi dan tambahan wawasan mengenai implementasi CNN ataupun *MobileNetV3* di masa yang akan datang.

3.1.2. Analisis Kebutuhan

Analisis kebutuhan adalah langkah penting untuk mengenali data dan prosedur yang diperlukan dalam merancang sistem. Dalam proses perancangan, analisis kebutuhan dapat mencakup persyaratan fungsional dan non fungsional yang perlu terpenuhi supaya sistem yang dikembangkan dapat mencapai tujuannya.

3.1.2.1. Kebutuhan Fungsional

Kebutuhan Fungsional adalah proses yang akan dilakukan oleh sistem untuk memenuhi tujuannya. Kebutuhan Fungsional pada penelitian ini adalah sebagai berikut:

1. *Input*

Sistem dapat membaca *input* berupa gambar dari pengguna.

2. Pengenalan data

Sistem dapat mengenali gambar berupa retina mata ketika dilakukan proses prediksi oleh model.

3. Hasil Prediksi

Sistem dapat memberikan hasil prediksi kondisi retina mata yang dimasukkan

4. Keakuratan

Sistem dapat memberikan nilai *confidence* (keakuratan) terhadap hasil yang didapatkan

3.1.2.2. Kebutuhan Non Fungsional

Kebutuhan non-fungsional adalah kebutuhan pelengkap yang mencakup atribut, fitur, pembatasan layanan, atau aspek-aspek lainnya dari sistem. Di bawah ini terdapat daftar kebutuhan fungsional yang diperlukan dalam sistem:

1. Ekstensi

Batasan ekstensi gambar yang dapat diterima sebagai *input* adalah .JPG, .PNG, dan .JPEG.

2. Output

Batasan kondisi retina mata yang dapat diprediksi sebagai keluaran (*output*) adalah *Glaucoma*, *Diabetic Retinopathy*, *Cataract*, dan Normal.

3. Internet

Memiliki koneksi internet untuk terhubung dengan sistem.

4. Durasi

Sistem dirancang untuk dapat mendeteksi kondisi mata dalam waktu kurang dari 10 detik.

5. Deteksi kesalahan

Memiliki kemampuan untuk mendeteksi kesalahan atau *error* apabila pengguna saat melakukan *input data* yang tidak sesuai.

3.2. Perancangan Sistem

Langkah selanjutnya adalah perancangan sistem, model akan dibangun dengan melalui proses yang dijabarkan sebagai berikut:

1. Augmentasi data

Augmentasi data adalah tahapan proses pengubahan data bagi dari segi bentuk maupun ukuran dengan tujuan mendapatkan bentuk data yang lebih variatif. Dengan proses augmentasi data, model yang akan digunakan dapat memprediksi data *testing* dengan akurasi yang lebih baik. Jenis augmentasi data yang akan digunakan terdapat dalam Tabel 3.1 di bawah ini.

Tabel 3. 1 Augmentasi Data

No.	Data Augmentasi	Value
1.	<i>Random Flip</i>	<i>Horizontal and Vertical</i>
2.	<i>Random Rotation</i>	0.4
3.	<i>Random Zoom</i>	0.2
4.	<i>Random Seed</i>	123
5.	<i>Image Size</i>	224 X 224

2. *Modelling*

Tahap selanjutnya adalah *modelling* di mana model akan dibentuk dari data yang telah melalui tahap augmentasi data, tahap *modelling* akan dimulai dengan inisiasi model *MobileNetV3* yang akan digunakan sebagai berikut.

Tabel 3. 2 Inisiasi MobileNetV3

No.	<i>MobileNetV3</i>	Value
1.	Jenis Model	Large
2.	<i>input shape</i>	224 X 224 X 3
3.	<i>trainable</i>	True

Selanjutnya adalah penyusunan lapisan model secara sekuesial untuk membangun model dengan susunan sebagai berikut.

- Lapisan *input* yang akan menerima data input dengan dimensi 224 x 224 piksel dan 3 saluran warna (RGB).
- Lapisan augmentasi yang akan mengolah data *train* yang digunakan.
- Lapisan *MobileNetV3-Large* sebagai model *pre-trained* yang akan mengolah data untuk mendapatkan fitur-fitur yang diperlukan dalam sistem.
- Lapisan *Batch Normalization* yang diterapkan pada output lapisan *pre-trained* model untuk mempercepat proses pelatihan model sekaligus meningkatkan kinerja model.

- e. Lapisan *Dropout* yang akan menonaktifkan sebagian unit pada model selama pelatihan dengan tujuan untuk mengurangi overfitting, nilai yang digunakan pada penelitian ini adalah 0,5.
- f. Lapisan prediksi yang tersusun dari 4 neuron, fungsi aktivasi *softmax*, dan juga regularisasi kernal L2 dengan nilai 0,05 untuk mengurangi kompleksitas pada model.
- g. Lapisan *output* yang digunakan untuk mendapatkan *output* dari lapisan prediksi.

Setelah lapisan untuk *Modelling* disusun, model akan dilakukan *compiling* dengan fungsi sebagai berikut.

Tabel 3. 3 Fungsi untuk Pelatihan Model

No.	Fungsi	Value
1.	<i>Optimizer</i>	Lion
2.	<i>Learning Rate (Optimizer)</i>	0,00001
3.	<i>Loss</i>	<i>Categorical Crossentropy</i>
4.	<i>Metric</i>	<i>Accuracy</i>
5.	Epochs	50

3. *Evaluation Metric*

Tahap terakhir adalah evaluasi mode dengan menentukan *evaluation metric*, evaluasi model merupakan proses evaluasi model dengan memperhitungkan performa model selama pembangunan model berlangsung dan *evaluation metric* adalah pengukuran yang digunakan pada model. Pada tahap evaluasi model, pengukuran utama yang akan digunakan pada sistem ini adalah akurasi.

4. *Confusion Matrix*

Evaluasi model juga akan menggunakan *confusion matrix* untuk melihat performa model dalam memprediksi data validasi yang telah ditentukan. *Confusion matrix* terdiri dari 2 bagian nilai, yaitu nilai *predicted* yang merupakan nilai yang didapatkan saat prediksi dan nilai *actual* yang merupakan nilai asli dari data yang

diberikan. Dari kedua nilai ini akan didapatkan empat penilaian yang akan memastikan model telah memprediksi dengan baik, nilai tersebut adalah:

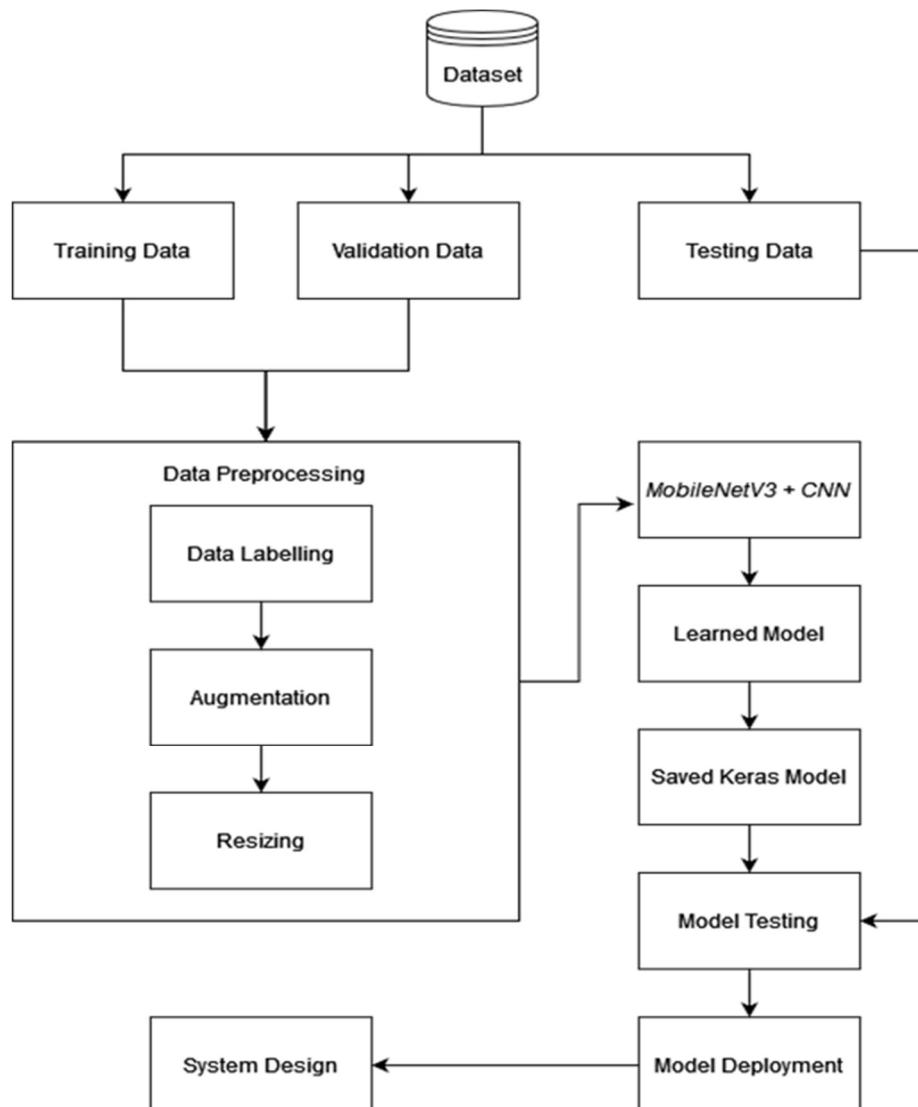
- **True Positive** yang menghitung total data positif dengan benar
- **True Negative** yang menghitung total data negatif dengan benar
- **False Positive** yang menghitung total data positif dengan salah
- **False Negative** yang menghitung total data negatif dengan salah

3.3. Pemodelan Sistem

Pemodelan sistem menjadi proses penguraian langkah-langkah interaksi antara pengguna dengan aplikasi yang dibangun, sehingga sistem dapat beroperasi secara efisien. Umumnya, pemodelan sistem diwujudkan dalam format *Unified Modelling Language* (UML). Dalam penelitian ini, format UML yang akan digunakan adalah *Activity Diagram*, *Use Case Diagram*, *Sequence Diagram*, dan Diagram umum sistem.

3.2.1. Diagram Umum Sistem

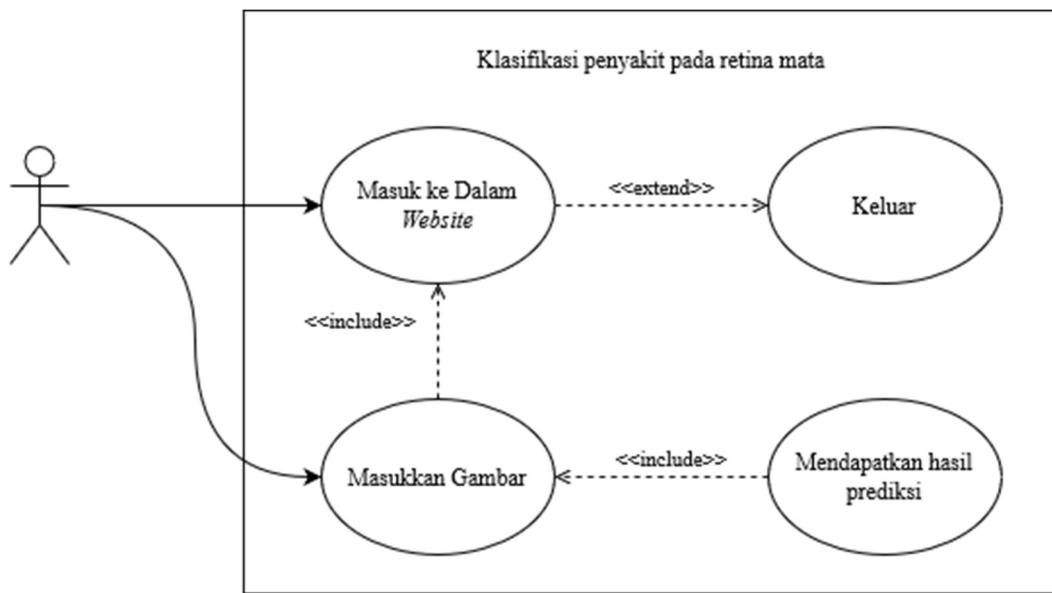
Diagram umum pada sistem merupakan gambaran visual tentang bagaimana sistem berjalan dengan memperlihatkan bagaimana interaksi, aliran, dan proses setiap komponen-komponen dalam sistem berlangsung. Seluruh desain aplikasi ini akan dijelaskan secara rinci dalam ilustrasi di bawah ini:



Gambar 3. 1 Diagram Umum Sistem

3.2.2. Use Case Diagram

Use Case Diagram merupakan bagian dari *Unified Modeling Language* (UML) yang berfungsi menggambarkan interaksi antara sistem dengan pelaku yang berinteraksi dengan sistem tersebut. Pelaku adalah entitas luar yang berperan dalam berinteraksi dengan sistem, seperti manusia, perangkat keras, atau sistem lainnya. Berikut adalah *use case diagram* yang digunakan pada sistem:

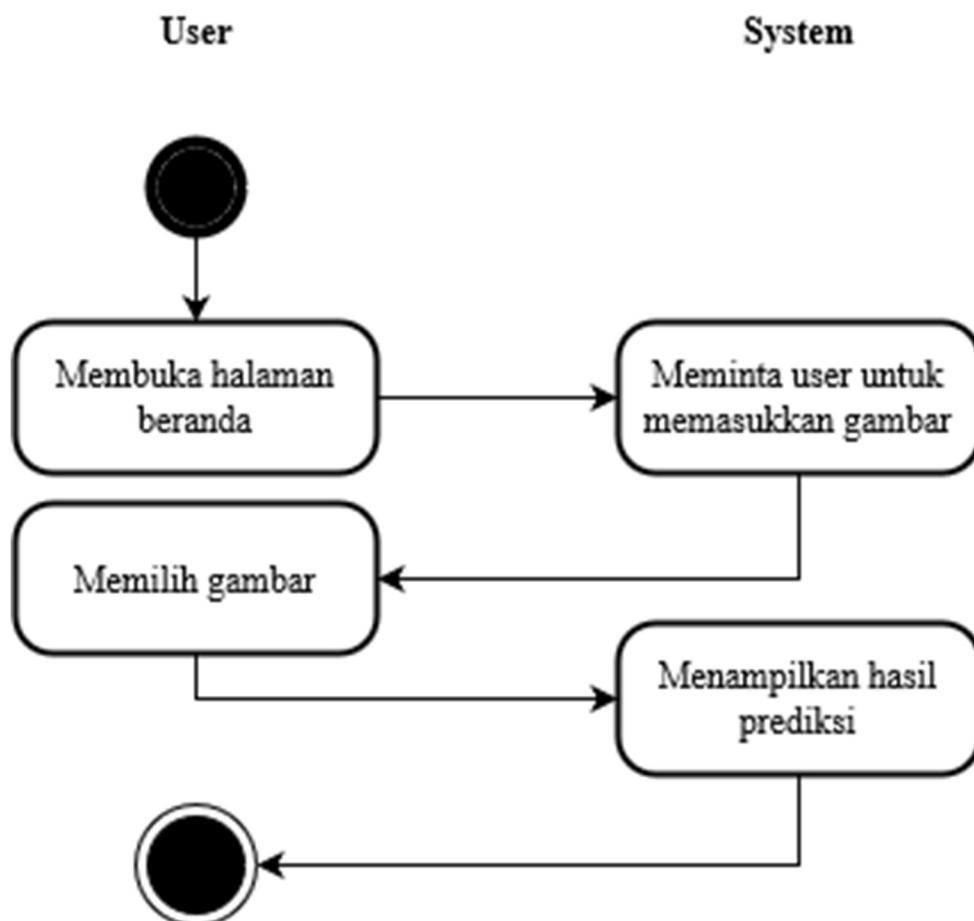


Gambar 3. 2 Use Case Diagram

Pada Gambar 3.2. *User* akan memulai interaksi dengan masuk ke bagian beranda aplikasi dan dapat langsung masuk ke inti program. *User* akan memasukkan sebuah gambar untuk diprediksi mengenai kondisi retina mata. Setelah itu, *user* akan menerima hasil prediksi dari data yang diberikan.

3.2.3. Activity Diagram

Activity Diagram merupakan bagian dari *UML* yang bertujuan sebagai representasi visual urutan alur kerja atau proses kegiatan pada sistem dimulai dari langkah awal hingga langkah akhir. Berikut adalah *Activity Diagram* yang dipakai dalam sistem:



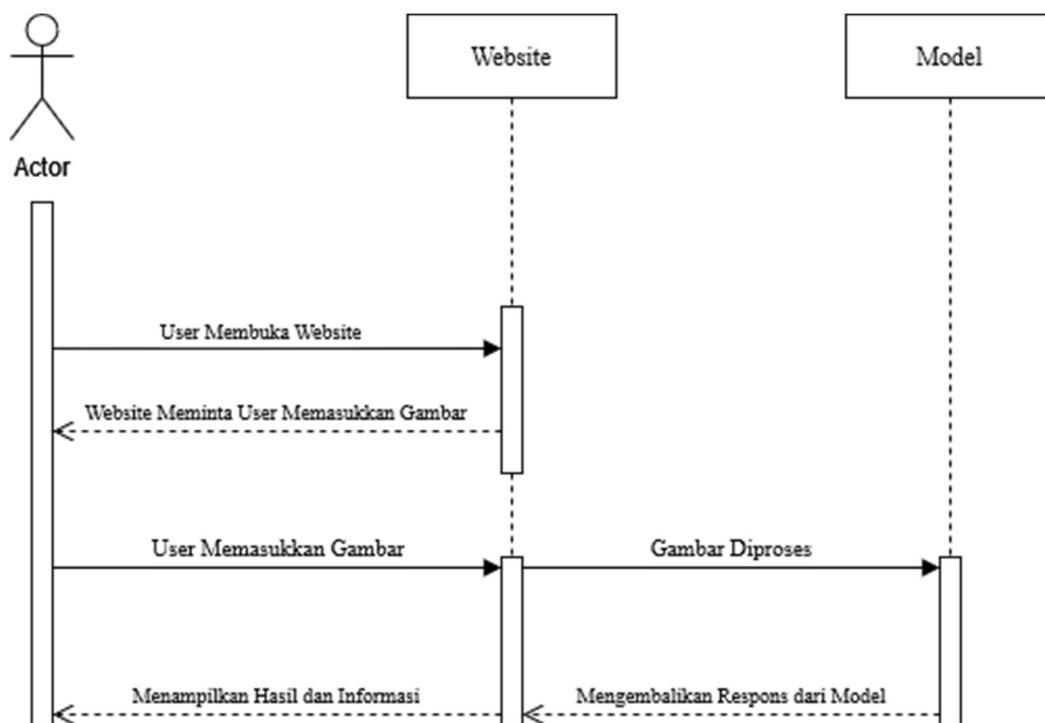
Gambar 3.3 Activity Diagram

Gambar 3.3 menunjukkan *activity diagram* dari sistem. Dimulai dari user yang membuka halaman beranda aplikasi. Apabila *User* memilih gambar maka sistem akan meminta kembali kepada *User* untuk melakukan *input* gambar. *User* kemudian melakukan *input* gambar dan sistem akan mengembalikan hasil prediksi tersebut kepada *User*.

3.2.4. Sequence Diagram

Sequence Diagram adalah bagian dari *Unified Modeling Language* (UML) yang digunakan untuk memvisualisasikan interaksi antara *user* dengan objek atau komponen dalam sistem dalam urutan waktu tertentu. Diagram ini bertujuan untuk memvisualisasikan bagaimana objek-objek berinteraksi terhadap satu sama lain dalam

konteks sebuah skenario atau proses tertentu. Berikut adalah *Sequence Diagram* yang digunakan pada sistem:



Gambar 3.4 Sequence Diagram

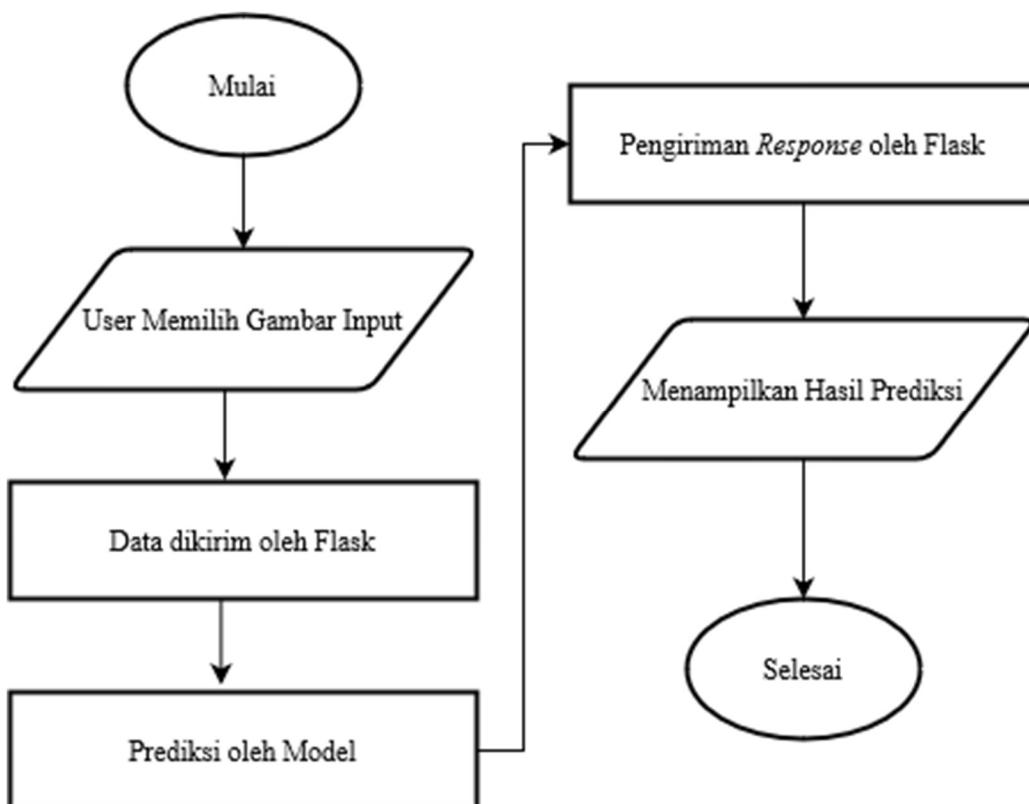
Pada *Sequence Diagram* diatas, terdapat 3 komponen yang saling berinteraksi, yaitu aktor (*user*), *Website*, dan *Model*. Diagram diatas menjelaskan proses dari awal *user* membuka halaman *website* hingga permintaan *user* selesai dilaksanakan dan dikembalikan.

3.3. Flowchart

Flowchart (Diagram Alir) merupakan representasi visual dari kumpulan langkah-langkah yang akan dijalankan oleh sebuah algoritma secara berurutan. Setiap tahapan prosesnya direpresentasikan dengan simbol yang berbeda dan dilengkapi dengan penjelasan untuk setiap tahapnya.

3.3.1. Flowchart Sistem

Berikut adalah *flowchart* sederhana dari sistem yang digunakan:

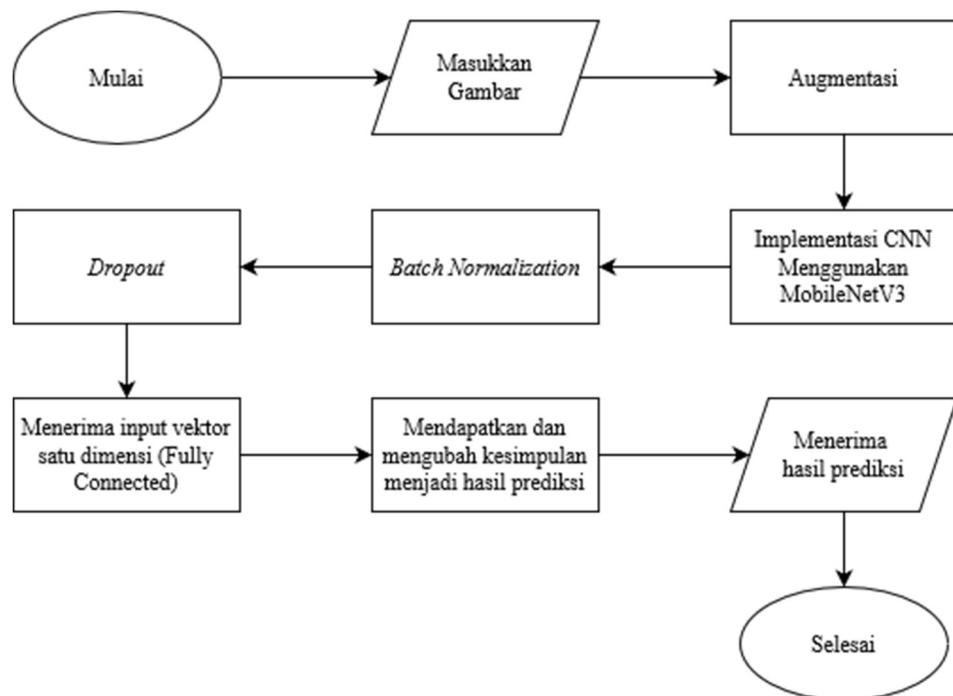


Gambar 3. 5 Flowchart Sistem

Pada Gambar 3.5 dapat dilihat bahwa alur kerja dari sistem yang pertama yakni *user* memilih gambar yang akan digunakan. Kemudian, gambar akan dikirim oleh Flask kepada model. Model akan memprediksi dan mengembalikan *response* berdasarkan hasil prediksi kepada *user*. Flask kemudian menampilkan hasil dan informasi relevan terhadap prediksi yang dilakukan oleh model.

3.3.2. Flowchart Algoritma Convolutional Neural Network (CNN)

Berikut adalah *flowchart* sederhana dari algoritma CNN:



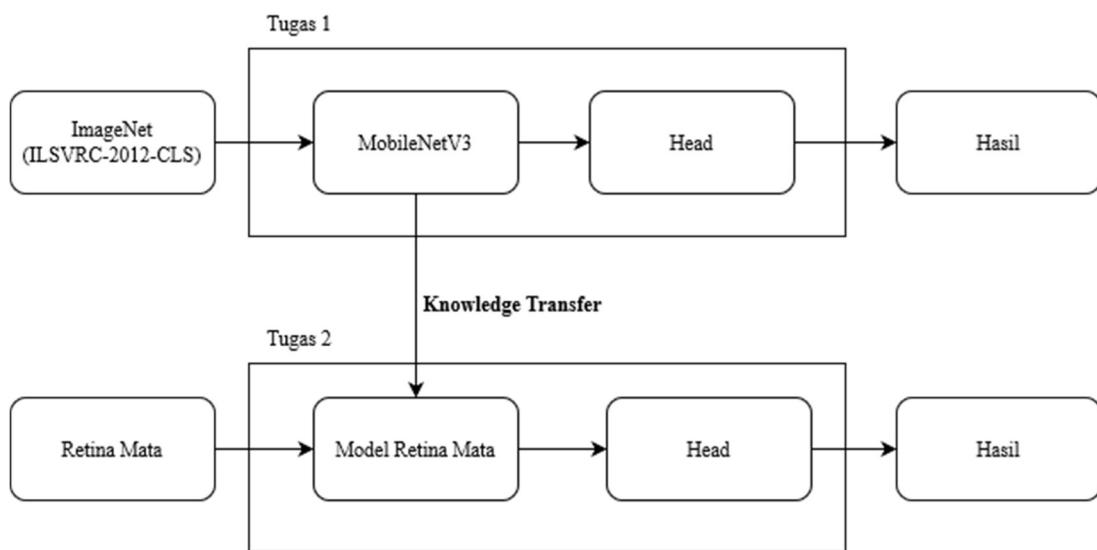
Gambar 3. 6 Flowchart Algoritma CNN

Gambar 3.6 menggambarkan bagaimana cara kerja dari algoritma CNN pada penelitian kali ini. Dimulai dengan menyediakan dan memasukkan gambar yang telah dilakukan proses augmentasi, selanjutnya data diterima oleh model diteruskan menuju lapisan konvolusi pada model *MobileNetV3*. Setelah data diproses dengan arsitektur *MobileNetV3*, *Batch Normalization* dilakukan untuk membuat pelatihan model lebih cepat dan stabil dengan mengurangi rata-rata *batch* pelatihan lalu dibagi dengan standar deviasi dari *batch* tersebut. Selanjutnya, lapisan model diberikan nilai *Dropout* yang akan menonaktifkan *input unit* secara random untuk menghindari *overfitting*. Selanjutnya, teknik *flatten* dilakukan untuk mengubah fitur menjadi vektor satu dimensi

untuk dimasukkan ke lapisan *fully connected* yang hanya bisa menerima data *input* dalam bentuk vektor satu dimensi. Setelah itu, lapisan *fully connected* akan menarik kesimpulan berdasarkan semua data yang diterima dan mengembalikan hasil prediksi dari gambar tersebut.

3.3.3. Flowchart arsitektur MobileNetV3

Berikut adalah *flowchart* sederhana dari arsitektur *MobileNetV3*:



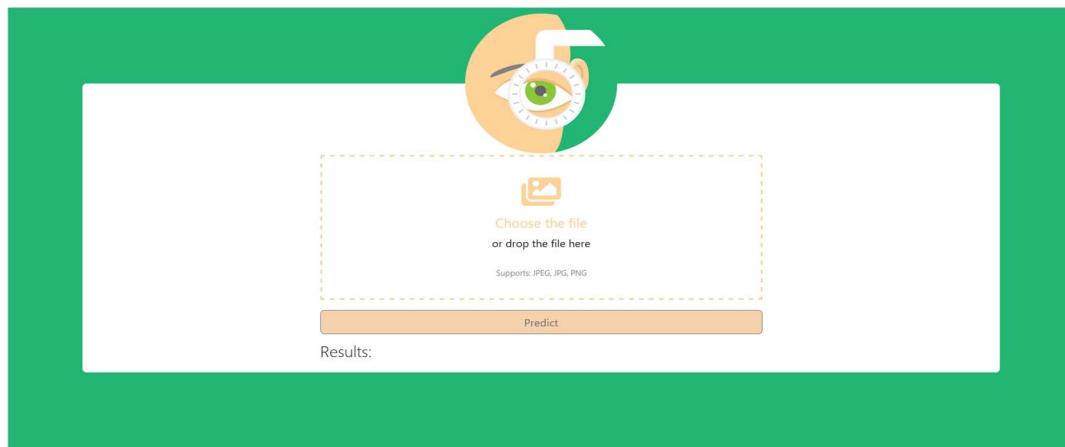
Gambar 3. 7 Flowchart Arsitektur MobileNetV3

Pada gambar 3.7 dapat dilihat bagaimana cara kerja dari arsitektur *MobileNetV3*. Model mengambil *knowledge* dari model *MobileNetV3* yang telah dilatih sebelumnya menggunakan dataset *ImageNet (ILSVRC-2012-CLS)*. Hasil dari pelatihan model tersebut akan dikirimkan ke model tujuan untuk digunakan terhadap masalah-masalah tertentu, hal ini disebut dengan *Knowledge Transfer*. Setelah proses *knowledge transfer* dilakukan, model akan dilatih kembali dengan data yang digunakan dalam penelitian yaitu penyakit retina mata. Hasil dari prediksi tersebut akan digunakan untuk membentuk sebuah model yang dapat memprediksi data baru untuk selanjutnya.

3.4. Perancangan *Interface*

Perancangan *Interface* adalah fase penciptaan desain sistem yang akan diaplikasikan oleh model. Perancangan *Interface* menjadi langkah yang diperlukan untuk memastikan bahwa seluruh proses pembuatan sistem telah berjalan dengan lebih efisien sesuai dengan panduan yang telah didefinisikan dalam desain *Interface* tersebut.

3.4.1. Halaman Prediksi dengan Gambar



Gambar 3. 8 Halaman Prediksi Dengan Gambar

Halaman diatas menunjukkan *Interface* dari halaman prediksi dengan gambar akan ditampilkan. Pada halaman ini, *user* akan memasukkan gambar untuk diprediksi, kemudian gambar tersebut akan ditampilkan dan pengguna dapat menekan tombol *predict* untuk mengetahui hasil prediksi dari gambar tersebut.

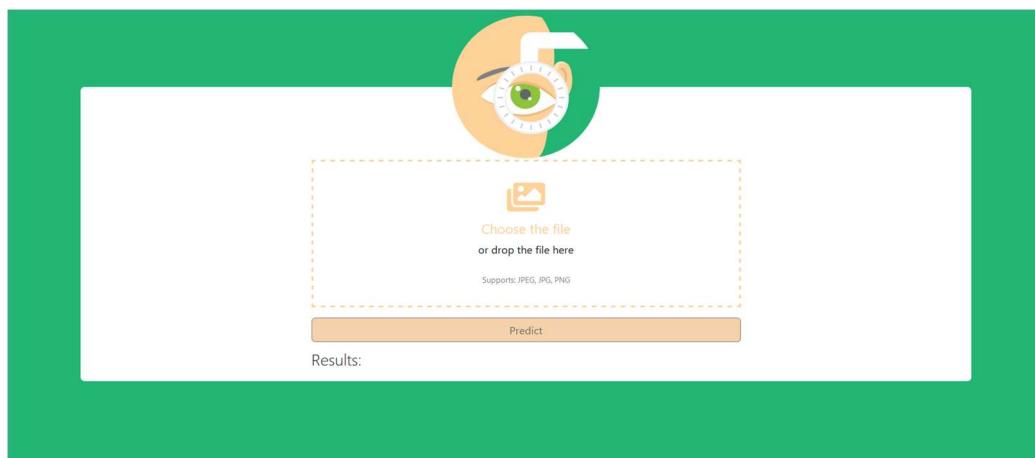
BAB IV

IMPLEMENTASI DAN PENGUJIAN SISTEM

4.1. Implementasi

Pada penelitian ini, sistem akan dibangun dengan memanfaatkan algoritma CNN dan arsitektur *transfer learning*. Sistem akan dibangun dengan dua bagian, yakni *front-end* yang akan dibangun menggunakan JavaScript (JS) dan *back-end* yang dibangun menggunakan bantuan bahasa pemrograman *python* dengan bantuan sebuah *framework* yaitu Flask.

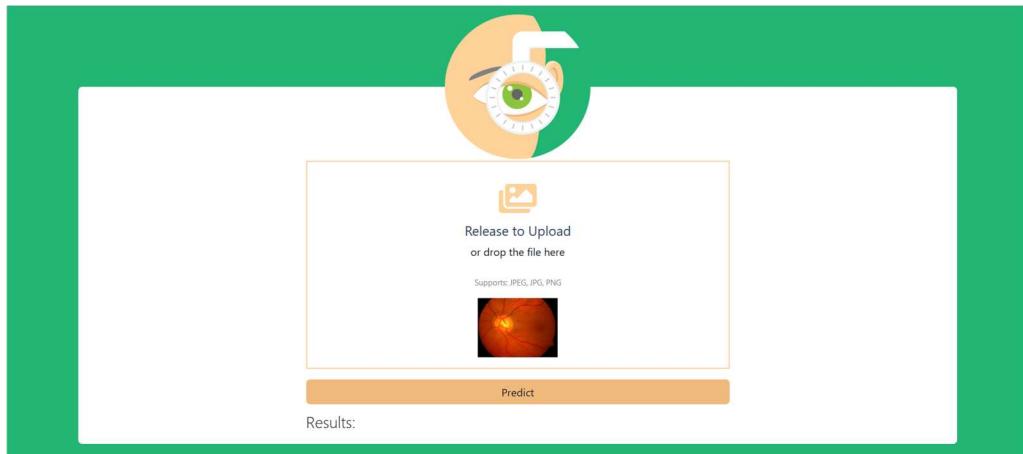
4.1.1. Landing Page



Gambar 4. 1 Landing Page

Gambar 4.1 merupakan bagian *landing page* pada sistem yang dibangun, bagian ini akan menjadi halaman pertama yang dilihat oleh user saat sistem diakses. *User* akan diminta untuk memasukkan sebuah gambar untuk diprediksi oleh web.

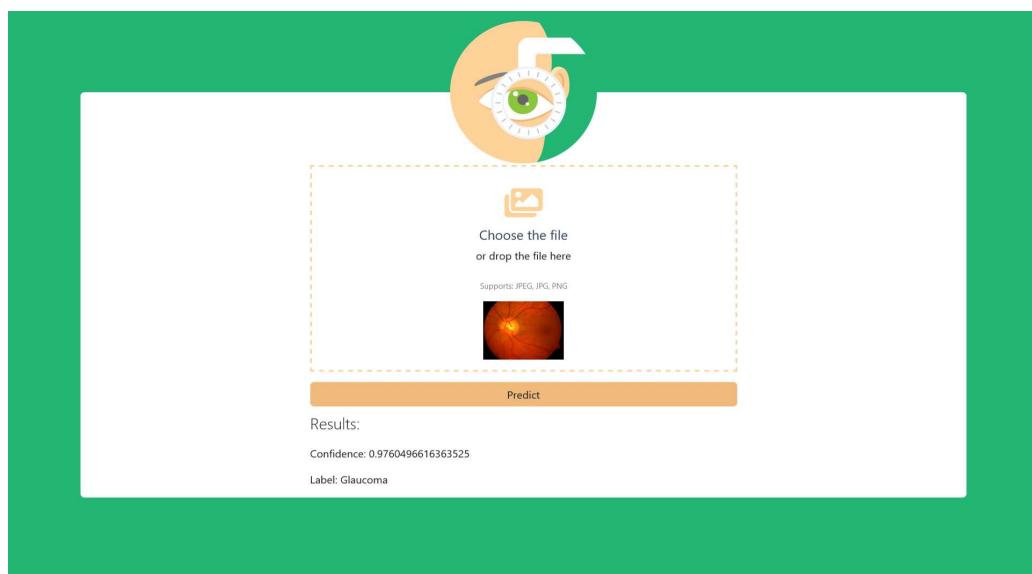
4.1.2. Halaman Input



Gambar 4. 2 Halaman Prediksi

Pada Gambar 4.2 di atas dapat dilihat bahwa halaman prediksi telah menerima input. Pada halaman ini, *user* akan melakukan *input* data berupa gambar yang dapat diambil dari perangkat mereka dengan cara mengunggah (*upload*) atau mengarahkan foto yang ingin digunakan ke dalam kotak kuning pada halaman website tersebut (*drag and drop*). Prediksi akan dilakukan apabila user telah menekan tombol *Predict*.

4.1.3. Halaman Hasil Prediksi

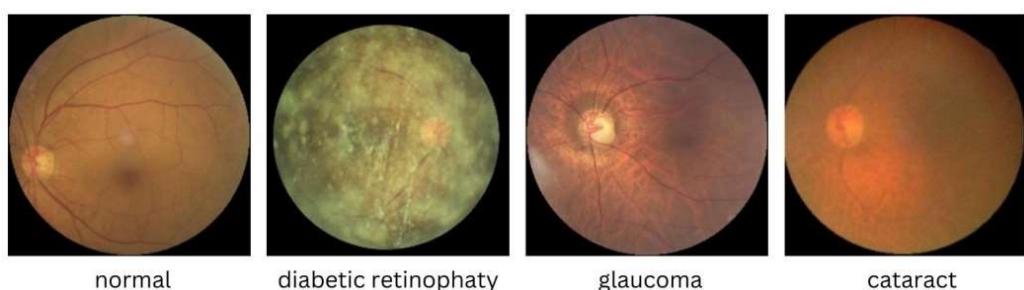


Gambar 4. 3 Halaman Hasil Prediksi

Pada Gambar 4.3 dapat dilihat halaman prediksi telah memberikan hasil prediksi terhadap gambar. Sistem akan memberikan hasil prediksi berupa label yang merupakan hasil pemrosesan antara model dengan data yang diberikan, sistem juga akan memberikan nilai *Confidence* untuk mengindikasikan seberapa tepat model ini dalam memprediksi gambar yang dimasukkan berdasarkan data yang telah dipelajari.

4.2. Sumber Data

Data citra fundus retina mata diperoleh melalui situs Kaggle (Doddi, 2022). Data yang digunakan berasal dari kumpulan-kumpulan sumber data yang valid seperti IDRiD (Indian Diabetic Retinopathy Image Dataset) dan sebagainya. Data tersebut dibagi menjadi 4 kelas yang terdiri dari *cataract*, *glaucoma*, *diabetic retinopathy*, dan normal. Total data yang diperoleh adalah 4180 dengan jumlah citra pada kelas *cataract* sebanyak 1006, *glaucoma* sebanyak 1007, *diabetic retinopathy* sebanyak 1098, dan normal sebanyak 1074. Sampel citra dari masing-masing kelas ditampilkan pada Gambar 4.4.



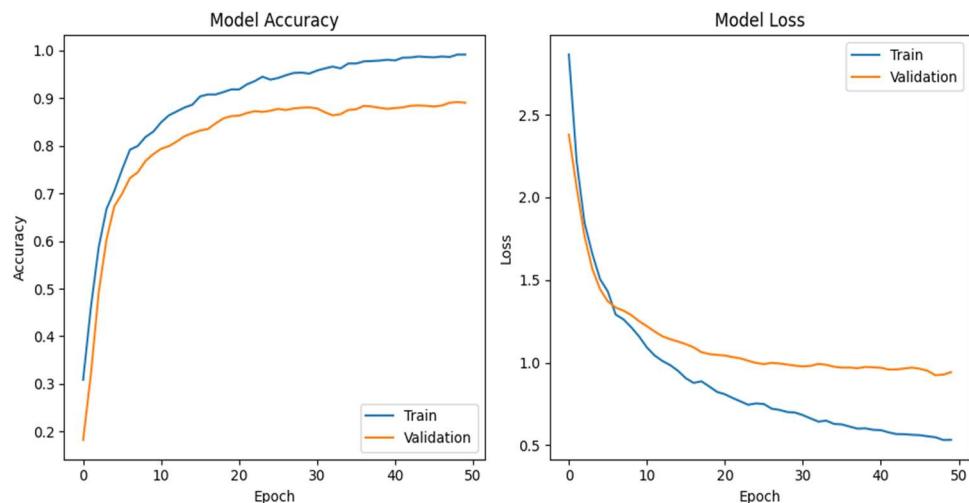
Gambar 4. 4 Citra fundus retina mata

4.3. Pengujian Sistem

Pengujian sistem adalah tahapan yang akan dilakukan setelah tahapan implementasi arsitektur berhasil dilakukan, pengujian sistem bertujuan membuktikan bahwa sistem yang telah dibangun dapat berjalan dengan benar dalam memprediksi penyakit mata melalui fundus retina mata. Pada tahap pengujian, terdapat 3 jenis penyakit yang akan diprediksi. Penyakit tersebut adalah, *Glaucoma*, *Diabetic*

Retinopathy, dan *Cataract*. Selain itu, model juga akan memprediksi bentuk retina mata yang normal. Berikut adalah langkah-langkah yang dilakukan dalam pengujian sistem.

a. Accuracy Plot



Gambar 4.5 Grafik Training dan Validation Accuracy

Grafik pada Gambar 4.5 merupakan hasil perhitungan terhadap akurasi dan loss dari data pelatihan dan data validasi. Pada grafik ini, kedua garis terlihat saling berdampingan yang menunjukkan bahwa model berhasil melakukan prediksi dengan baik menggunakan data latihan dan data validasi.

b. Evaluation Metrics

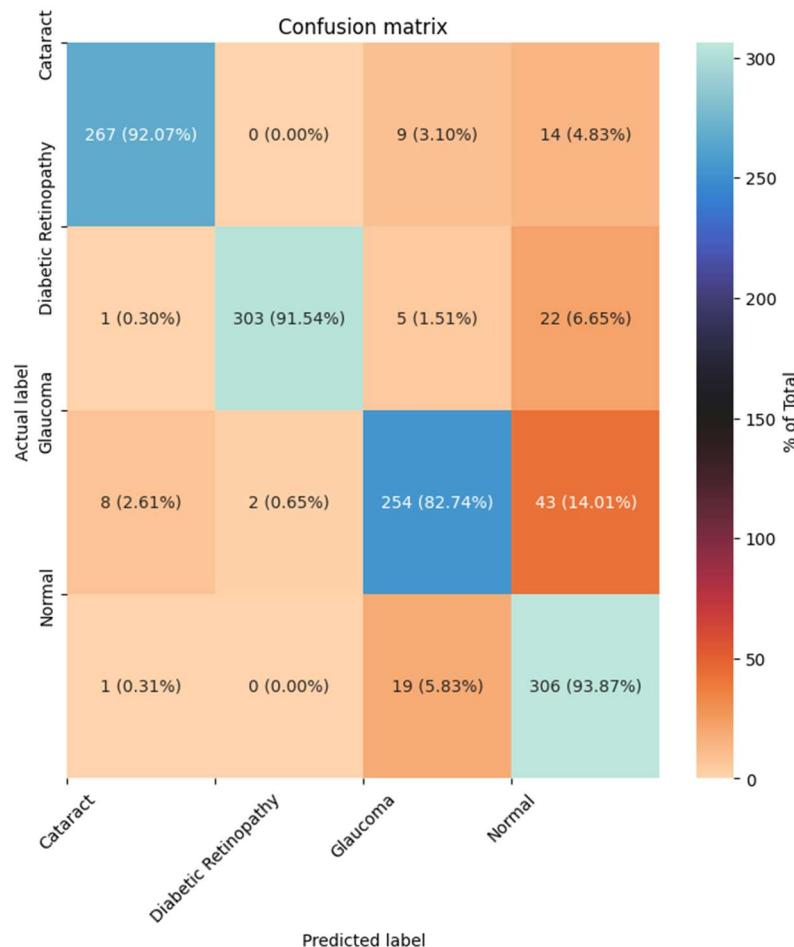
Tabel 4.1 Evaluation Metrics

No.	Evaluasi	Nilai
1.	<i>Accuracy</i>	90,11%
2.	<i>Precision</i>	90,18%
3.	<i>Recall</i>	90.11%
4.	<i>Area under curve</i>	97,56%
5.	<i>Categorical Crossentropy</i>	38,62%

No.	Evaluasi	Nilai
6.	<i>Mean Squared Error</i>	4.15%

Tabel 4.1 merupakan hasil evaluasi model dengan melihat *evaluation metric* yang telah ditentukan. Dengan akurasi sebesar 90,11%, dapat dipastikan bahwa model telah mencapai performa yang baik dan dapat digunakan untuk memprediksi jenis penyakit pada retina mata.

c. *Confusion Matrix*



Gambar 4. 6 *Confusion Matrix*

Dari hasil *Confusion Matrix* pada Gambar 4.6 di atas, dapat disimpulkan bahwa model telah berhasil memprediksi data sesuai dengan nilai aktual dari data

yang diberikan. Persentase *True Positive (TN)* dari tiap-tiap kelas adalah sebagai berikut.

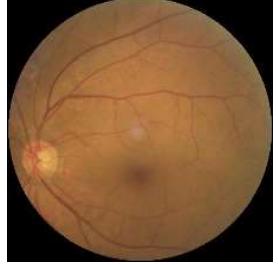
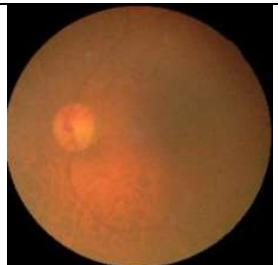
Tabel 4.2 Nilai True Positive

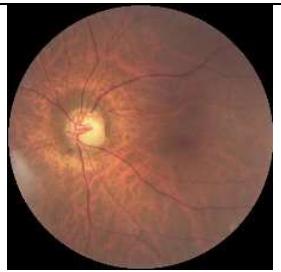
No.	Kelas	Persentase <i>True Positive (TP)</i>
1	<i>Cataract</i>	92,07
2	<i>Diabetic Retinopathy</i>	91,54
3	<i>Glaucoma</i>	82,74
4	<i>Normal</i>	93,87

d. *Testing Data*

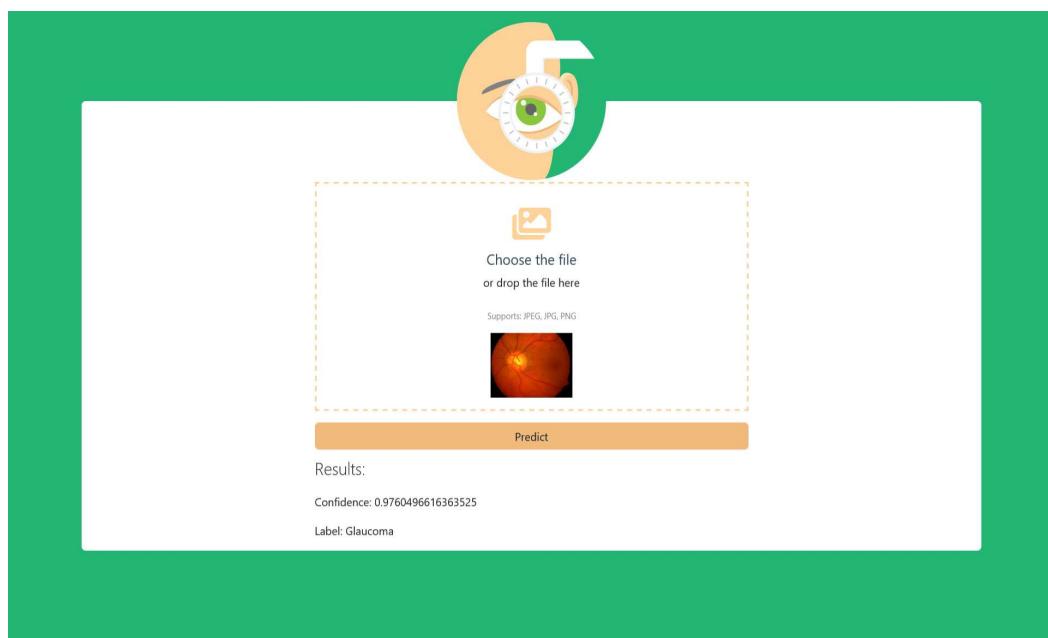
Tahap selanjutnya adalah menentukan testing data yang digunakan untuk *model testing*, data yang digunakan adalah data yang tidak termasuk dalam data latih maupun data validasi. Data yang digunakan dapat dilihat pada Tabel 4.3 di bawah ini.

Tabel 4.3 Daftar Gambar Fundus Retina mata

No.	Nama File	Gambar
1.	normal.jpg	
2.	cataract.jpg	

No.	Nama File	Gambar
3.	glaucoma.jpg	
4.	diabetic retinopathy.jpg	

e. *System Design*



Gambar 4. 7 Percobaan pada sistem website

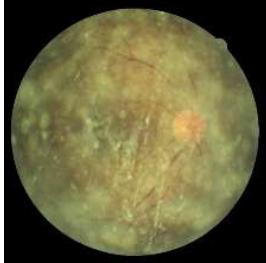
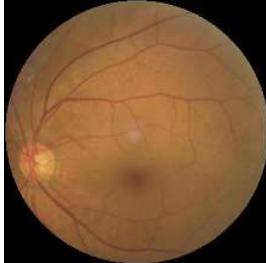
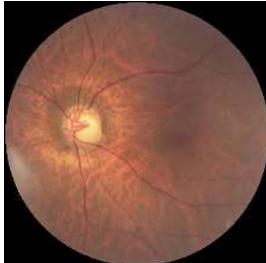
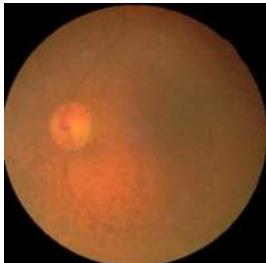
Tahap terakhir adalah percobaan yang dilakukan terhadap model menggunakan *platform* berupa *website*. Setelah data dimasukkan, *user* akan menekan tombol *Predict* untuk mendapatkan hasil uji dari model yang digunakan berupa label hasil uji dan *confidence* untuk memberikan informasi seberapa akurat

model ini dalam memprediksi data uji yang diberikan. Hasil percobaan dapat dilihat pada Gambar 4.7 yang menandakan bahwa sistem telah berjalan dengan baik.

4.4. Hasil Pengujian Sistem

Setelah dilakukan pengujian sistem, hasil yang didapat dengan mencoba semua fundus mata yang telah dijelaskan sebelumnya adalah:

Tabel 4.4 Daftar Hasil Percobaan model

No.	Gambar	Hasil Prediksi	Hasil Aktual
1.		Diabetic Retinopathy	Diabetic Retinopathy
2.		Normal	Normal
3.		Glaucoma	Glaucoma
4.		Cataract	Cataract

Dari hasil Tabel 4.4 di atas, dapat disimpulkan bahwa model yang dibangun telah dapat mengklasifikasi kondisi retina mata sesuai dengan kondisi aslinya.

4.5. User Acceptance Testing

User Acceptance Testing (UAT) merupakan pengujian akhir yang dilakukan oleh pengguna pada sistem yang dirancang untuk memvalidasi bahwa sistem dapat diterima dan memenuhi kebutuhan pengguna. UAT disusun dalam bentuk tabel yang berisi pernyataan dan akan diisi oleh pakar sebagai penguji dan mengevaluasi sistem klasifikasi penyakit pada retina mata yang telah dibangun. Hasil dari UAT dapat dilihat pada Tabel 4.5 di bawah ini.

Tabel 4.5 User Acceptance Testing

Pernyataan	Sangat Tidak Setuju	Tidak Setuju	Netral	Setuju	Sangat Setuju
Sistem tidak memiliki kesulitan dalam penggunaanya					✓
Sistem klasifikasi menghasilkan diagnosa penyakit dalam waktu yang lebih singkat			✓		
Sistem klasifikasi penyakit pada retina mata menghasilkan diagnosa penyakit sesuai dengan diagnosa pakar				✓	
Sistem klasifikasi penyakit pada retina mata layak digunakan sesuai dengan kebutuhan medis				✓	
Sistem layak dikembangkan untuk dapat mendiagnosis jenis penyakit mata yang lebih banyak lagi					✓

Dari hasil Tabel 4.5, dapat dinyatakan bahwa sistem klasifikasi penyakit pada retina mata tidak memiliki kesulitan yang mengganggu dalam penggunaanya, menghasilkan diagnosa penyakit yang sesuai dengan diagnosa pakar, layak digunakan sesuai kebutuhan medis, dan layak dikembangkan untuk dapat mendiagnosis jenis penyakit mata yang lebih banyak lagi. Hanya saja, Pakar menyarankan agar jenis penyakit yang diklasifikasikan dapat dikembangkan dan dibagi sesuai dengan tingkat keparahannya sehingga pakar mendapatkan hasil diagnosis jenis penyakit sekaligus dengan jenisnya dalam waktu yang lebih cepat karena tentunya pakar dapat mengetahui jenis penyakit tetapi pakar akan memerlukan waktu yang lebih untuk memastikan tipe maupun tingkat keparahan jenis penyakit tersebut.

4.6. Faktor-Faktor Yang Memengaruhi Algoritma CNN

Berdasarkan hasil penelitian yang dilakukan, Faktor-faktor yang memengaruhi algoritma CNN selama penelitian berlangsung adalah:

1. **Data:** Apabila *data* berada dalam jumlah yang besar baik dari segi kuantitas, ukuran, maupun resolusi pada gambar, maka waktu perhitungan pada model akan menjadi lebih lama dibandingkan dengan data dalam jumlah yang lebih sedikit ataupun ringan.
2. **Jumlah Lapisan:** Model CNN yang memiliki lapisan yang lebih banyak akan menghasilkan kompleksitas yang lebih tinggi daripada model yang memiliki jumlah lapisan lebih sedikit.
3. **Runtime:** Penggunaan *runtime* seperti GPU membantu mengurangi waktu perhitungan dan meningkatkan efisiensi pada algoritma dengan lebih baik dibandingkan dengan hanya menggunakan CPU.
4. **Fungsi Aktivasi:** Penggunaan fungsi aktivasi yang tepat dapat mempengaruhi efisiensi model dalam mempelajari data.
5. **Pooling:** Operasi *pooling* dapat digunakan untuk mengurangi dimensi dari data sehingga dapat mengurangi waktu perhitungan pada model.

6. **Regularisasi:** Penggunaan teknik regularisasi dapat mempengaruhi kompleksitas terhadap model. Meskipun regularisasi dapat membantu menghindari *overfitting*, regularisasi juga dapat menambah kompleksitas dari model tersebut.
7. **Batch Normalization:** *Batch normalization* dapat membantu mempercepat pelatihan model, tetapi *batch normalization* juga dapat meningkatkan kompleksitas pada model.
8. **Ukuran Model:** Apabila sebuah model CNN memiliki banyak parameter, tentunya akan menambah waktu dalam melakukan perhitungan sehingga dapat memperbesar ukuran model.

4.7. Faktor-Faktor Yang Memengaruhi Arsitektur *MobileNetV3*

Berdasarkan hasil penelitian yang dilakukan, ada beberapa faktor yang memengaruhi performa dari *MobileNetV3-Large*, faktor-faktor tersebut adalah:

1. **Jumlah Lapisan:** Seperti *pre-trained model* pada umumnya, *MobileNetV3* juga memiliki jumlah lapisan yang banyak sehingga membuat model ini memiliki arsitektur yang kompleks.
2. **Ukuran Model:** Apabila parameter yang digunakan pada sebuah model berjumlah lebih banyak, maka ukuran model dapat menjadi lebih besar.
3. **Data:** Data seperti gambar dengan ukuran atau resolusi tinggi dapat mempengaruhi arsitektur *MobileNetV3* daripada gambar dengan resolusi rendah atau sedang karena memerlukan waktu lebih lama selama perhitungan berlangsung.

BAB V

PENUTUP

5.1. Kesimpulan

Kesimpulan yang diperoleh berdasarkan analisis, perancangan, implementasi dan pengujian Algoritma CNN dan Arsitektur *MobileNetV3* dalam prediksi penyakit pada mata melalui fundus retina mata adalah sebagai berikut.

1. Algoritma *Convolutional Neural Network* bersama Arsitektur *MobileNetV3* dapat digunakan untuk memprediksi penyakit pada mata melalui fundus retina mata dengan akurasi sebesar 98% terhadap data latih dan 90% terhadap data validasi.
2. Jumlah data dan pembagian data yang digunakan pada saat pelatihan sangat mempengaruhi Algoritma CNN untuk mendapatkan model yang stabil.
3. Ukuran gambar, filter, jumlah lapisan, fungsi aktivasi, *epoch* dan *runtime* sangat mempengaruhi kompleksitas model.
4. Menggunakan *MobileNetV3* dapat mempercepat proses pelatihan model sekaligus memperluas *knowledge* dari model target.
5. Prediksi penyakit pada mata melalui fundus retina mata dapat menjadi acuan dalam memeriksa kondisi mata dengan lebih akurat.

5.2. Saran

Setelah melakukan penelitian ini, saran yang dapat dijadikan bahan pertimbangan untuk penelitian selanjutnya adalah:

1. Memperbanyak kuantitas data yang digunakan agar akurasi model menjadi lebih sempurna.
2. Membagi data dengan tingkat keparahan yang dialami apabila data yang dikumpulkan dinilai cukup untuk dibagi berdasarkan tingkat keparahan penyakit yang dialami

3. Memperluas domain jenis penyakit yang diperlukan untuk diprediksi apabila data yang dikumpulkan dinilai cukup untuk dimasukkan ke dalam model.
4. Diharapkan kedepannya dapat bekerja sama dengan instansi yang berkaitan untuk memperkuat sistem agar berguna bagi petugas medis ataupun optikan yang menggunakan sistem ini.
5. Diharapkan kedepannya implementasi dapat diperluas sehingga penggunaan sistem tidak hanya melalui aplikasi *web*, melainkan penggunaan yang bersifat *multiplatform*.

DAFTAR PUSTAKA

- Ardyansyah, M. A., & Gunawansyah. (2023). Sistem Deteksi Level Diabetic Retinopathy Melalui Citra Fundus Mata dengan Menggunakan Metode CNN (Convolutional Neural Network). *G-Tech: Jurnal Teknologi Terapan*, 7(4), 1673–1682. <https://doi.org/10.33379/gtech.v7i4.3332>
- Bi, Q., Goodman, K. E., Kaminsky, J., & Lessler, J. (2019). What is Machine Learning? A Primer for the Epidemiologist. *American Journal of Epidemiology*, 188(12), 2222–2239. <https://doi.org/10.1093/aje/kwz189>
- Budu, E. (2023). What does pre-training a neural network mean? *Baeldung on Computer Science*. <https://www.baeldung.com/cs/neural-network-pre-training>
- Country Map & Estimates of Vision Loss Indonesia*. (2020). Retrieved from The International Agency for the Prevention of Blindness (IAPB): <https://www.iapb.org/learn/vision-atlas/magnitude-and-projections/countries/indonesia>
- Craig, L., & Awati, R. (2024). convolutional neural network (CNN). *Enterprise AI*. <https://www.techtarget.com/searchenterpriseai/definition/convolutional-neural-network>
- Doddi, G. V. (2022). *Eye Disease Classification*. Retrieved from Kaggle: <https://www.kaggle.com/datasets/gunavenkatdoddi/eye-diseases-classification>
- Elbrag, Y. (2023, February 1). *Measuring Neural Network Performance: Latency and Throughput on GPU*. Retrieved from Medium: <https://younsess-elbrag.medium.com/measuring-neural-network-performance-latency-and-throughput-on-gpu-5d54657871f0>
- Fransisca, P. S., & Matondang, N. (2023). Deteksi Citra Digital Penyakit Cacar Monyet menggunakan Algoritma Convolutional Neural Network dengan Arsitektur MobileNetV2. *Jurnal Ilmu Komputer Dan Agri-Informatika*, 10(2), 200–211. <https://doi.org/10.29244/jika.10.2.200-211>

- Hönemann, C. (2015, July). *Valsalva retinopathy - a rare complication following general anesthesia.* Retrieved from ResearchGate: https://www.researchgate.net/figure/Normal-ocular-fundus_fig3_280574401
- Horry, M. J., Chakraborty, S., Paul, M., Ulhaq, A., Pradhan, B., Saha, M., & Shukla, N. (2020). COVID-19 Detection Through Transfer Learning Using Multimodal Imaging Data. *IEEE Access*, 8, 149808–149824. <https://doi.org/10.1109/ACCESS.2020.3016780>
- Indraswari, R., Herulambang, W., & Rokhana, R. (2022). Deteksi Penyakit Mata Pada Citra Fundus Menggunakan Convolutional Neural Network (CNN). *Techno.Com*, 21(2), 378–389. <https://doi.org/10.33633/tc.v21i2.6162>
- Lusiani, M. Y. (2019, July 10). *Definisi dan Fungsi Retina Serta Bermacam Penyakit Retina.* Retrieved from KMN EyeCare: <https://www.klinikmatanusantara.com/id/ketahui-lebih-lanjut/info-kesehatan-mata-dari-kmn-eyecare/artikel/definisi-dan-fungsi-retina-serta-bermacam-penyakit-retina/>
- Marcella, D., Yohannes, Y., & Devella, S. (2022). Klasifikasi Penyakit Mata Menggunakan Convolutional Neural Network Dengan Arsitektur VGG-19. *Jurnal Algoritme*, 3(1), 60–70. <https://doi.org/10.35957/algoritme.v3i1.3331>
- Noor, A. (2024, January 23). *Gangguan Retina Mata dan Diagnosisnya.* Retrieved from Ciputra SMG EYE Clinic: <https://ciputrasmgeyeclinic.com/gangguan-retina-mata-dan-diagnosisnya/>
- Nuzzi, R., Boscia, G., Marolo, P., & Ricardi, F. (2021). The Impact of Artificial Intelligence and Deep Learning in Eye Diseases: A Review. *Frontiers in medicine*, 8, 710329. <https://doi.org/10.3389/fmed.2021.710329>
- Omori, Yuki. (2020). Image Augmentation for Eye Contact Detection Based on Combination of Pre-trained Alex-Net CNN and SVM. *Journal of Computers.* 15. 85-97. 10.17706/jcp.15.3.85-97
- Rodriguez, J. (2019, December 2). *Google open sources MobileNetV3 with new ideas to improve mobile computer vision models.* Retrieved from KDnuggets:

<https://www.kdnuggets.com/2019/12/google-open-sources-mobilenetv3-improve-mobile-computer-vision-models.html>

Sarker, I. H. (2021). Deep Learning: A Comprehensive Overview on Techniques, Taxonomy, Applications and Research Directions. *SN Computer Science*, 2(6), 1–20. <https://doi.org/10.1007/s42979-021-00815-1>

Shahriar, N. (2023, February 1). *What is Convolutional Neural Network — CNN (Deep Learning)*. Retrieved from Medium:

<https://nafizshahriar.medium.com/what-is-convolutional-neural-network-cnn-deep-learning-b3921bdd82d5>

Swasono, S., Damayanti, A., & Pratiwi, A. B. (2019). Retinal Diseases Classification Using Levenberg-Marquath (LM) Learning Algorithm for Pi Sigma Network (PSN) and Principal Component Analysis (PCA) Methods. *Journal of Physics: Conference Series*, 1306(1). <https://doi.org/10.1088/1742-6596/1306/1/012048>

William, W., & Lubis, C. (2022). Klasifikasi Penyakit Mata Menggunakan Cnn. *Jurnal Ilmu Komputer Dan Sistem Informasi*, 10(1), 1–4. <https://doi.org/10.24912/jiksi.v10i1.17834>

Yogi, M. (2020, July 9). *Implementing Machine Learning on Avocado Data Set*. Retrieved from Medium: <https://medium.com/analytics-vidhya/implementing-machine-learning-on-avocado-data-set-7217943cd7fd>

LAMPIRAN LISTING PROGRAM

App.py

```
from PIL import Image
import numpy as np
import io
from flask import Flask, render_template, request, jsonify
import tensorflow as tf
import tensorflow_hub as hub

app = Flask(__name__)

# Load your TensorFlow model
model = tf.keras.models.load_model('model_1.keras',
custom_objects={'KerasLayer': hub.KerasLayer()})

class_labels = {
    0: 'Cataracs',
    1: 'Diabetic Retinopathy',
    2: 'Glaucoma',
    3: 'Normal'
}

def preprocess_image(image_file):
    # Read the image from the file-like object
    image = Image.open(io.BytesIO(image_file.read()))

    # Resize the image
    image = image.resize((224, 224))

    # Convert the image to a numpy array and normalize it
    image_array = tf.keras.utils.img_to_array(image)

    # Add a batch dimension because the model expects batches
    # of images
    image_array = np.expand_dims(image_array, axis=0)

    return image_array
```

```

def postprocess_prediction(prediction):
    # Assuming the prediction is a single class probability
    # distribution
    predicted_class_index = np.argmax(prediction, axis=1)[0]
    predicted_class_label =
    class_labels[predicted_class_index]

    # Convert to a readable format
    response = {
        # 'predicted_class': int(predicted_class_index),
        'confidence': float(np.max(prediction)),
        'label': predicted_class_label
    }

    return response

@app.route('/predict', methods=['POST'])
def predict():
    if request.method == 'POST':
        image = request.files.get('file')
        if image:
            # Preprocess the image
            processed_image = preprocess_image(image)

            # Run model prediction
            prediction = model.predict(processed_image)

            # Process the prediction and return response
            response = postprocess_prediction(prediction)
            return jsonify(response)
        else:
            return jsonify({'message': 'No file provided'})
    return jsonify({'message': 'Invalid request method'})

@app.route('/', methods=['GET'])
def index():
    return render_template('index.html')

if __name__ == '__main__':
    app.run(debug=True)

```

ML Model.ipynb

```

import os
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
%matplotlib inline
from matplotlib.image import imread
import tensorflow as tf
from tensorflow.keras.preprocessing.image import
ImageDataGenerator
from tensorflow.keras.models import Sequential
from tensorflow.keras.callbacks import EarlyStopping
from tensorflow.keras.layers import Dense, BatchNormalization,
Dropout
from tensorflow.keras.losses import CategoricalCrossentropy
from tensorflow.keras.optimizers import Adam
import tensorflow_hub as hub
from tensorflow.keras.preprocessing import image

%pip install gdown
!gdown --fuzzy
"https://drive.google.com/file/d/1PdwPKo91TizcS6Lv_aB9SPhHxpP04b
OS/view?usp=sharing"
!unzip 'Penyakit Retina Mata 3.zip'

data_directory = './Penyakit Retina Mata 3/'
training_set_1 =
tf.keras.preprocessing.image_dataset_from_directory(
    data_directory,
    color_mode='rgb',
    batch_size=128,
    image_size=((224, 224)),
    shuffle=True,
    label_mode='categorical',
    seed=123,
    validation_split=0.3,
    subset='training'
)
validation_set_1 =
tf.keras.preprocessing.image_dataset_from_directory(
    data_directory,
    color_mode='rgb',
    batch_size=128,

```

```

        image_size=((224, 224)),
        shuffle=True,
        label_mode='categorical',
        seed=123,
        validation_split=0.3,
        subset='validation'
    )

augmentation_layers = tf.keras.Sequential([
    tf.keras.layers.RandomFlip("horizontal_and_vertical"),
    tf.keras.layers.RandomRotation(0.4),
    tf.keras.layers.RandomZoom(0.2)
])

get_mobilenet_v3_pretrained_model =
"https://tfhub.dev/google/imagenet/mobilenet_v3_large_100_224/feature_vector/5"
mobilenet_v3_pretrained_model =
hub.KerasLayer(get_mobilenet_v3_pretrained_model,
input_shape=(224, 224, 3), trainable=True)

inputs_1 = tf.keras.Input(shape=(224, 224, 3))
x_1 = augmentation_layers(inputs_1)
x_1 = mobilenet_v3_pretrained_model(x_1)
x_1 = tf.keras.layers.BatchNormalization()(x_1)
x_1 = tf.keras.layers.Dropout(0.5)(x_1)
prediction_layer_1 = tf.keras.layers.Dense(4,
activation='softmax',
kernel_regularizer=tf.keras.regularizers.l2(0.05))
outputs_1 = prediction_layer_1(x_1)
model_1 = tf.keras.Model(inputs_1, outputs_1)

model_1.compile(optimizer=Lion(learning_rate=0.00001),
loss='categorical_crossentropy', metrics=['accuracy'])
history_1 = model_1.fit(training_set_1,
validation_data=validation_set_1, epochs=50)

def acc_visual(history):
    plt.figure(figsize=(10, 5))
    plt.subplot(1, 2, 1)
    plt.plot(history.history['accuracy'])
    plt.plot(history.history['val_accuracy'])
    plt.title('Model Accuracy')
    plt.xlabel('Epoch')
    plt.ylabel('Accuracy')
    plt.legend(['Train', 'Validation'], loc='lower right')

    # Plot training & validation loss values

```

```

plt.subplot(1, 2, 2)
plt.plot(history.history['loss'])
plt.plot(history.history['val_loss'])
plt.title('Model Loss')
plt.xlabel('Epoch')
plt.ylabel('Loss')
plt.legend(['Train', 'Validation'], loc='upper right')

plt.tight_layout()
plt.show()

acc_visual(history_1)

model_1.save('model_1.keras')

test_model = tf.keras.models.load_model('model_1.keras',
custom_objects={'KerasLayer':hub.KerasLayer})

label = {
    "0": "Cataract",
    "1": "Diabetic Retinopathy",
    "2": "Glaucoma",
    "3": "Normal"
}

def predict(image_directory, model):
    image = tf.keras.utils.load_img(image_directory,
target_size=(224,224))
    input_arr = tf.keras.utils.img_to_array(image)
    input_arr = np.array([input_arr])
    predictions = np.argmax(model.predict(input_arr))
    return predictions

!gdown --fuzzy
"https://drive.google.com/file/d/1yw3jjTQfdornZRlI5SJqzbem3UcvIsY4/view?usp=sharing"
!unzip "upload test-20240502T062928Z-001.zip"

for i in os.listdir('/content/upload test'):

    print(f"Class: {str(i)}, prediction:
{label[str(predict(os.path.join('./upload test', i),
test_model))]}")

# Import Metric, precision, dan recall
from tensorflow.keras import metrics
from sklearn.metrics import confusion_matrix, precision_score,
recall_score

METRICS = [

```

```

    metrics.CategoricalCrossentropy(name='categorical cross
entropy'),
    metrics.MeanSquaredError(name='mean squared error'),
    metrics.TruePositives(name='true positive'),
    metrics.FalsePositives(name='false positive'),
    metrics.TrueNegatives(name='true negative'),
    metrics.FalseNegatives(name='false negative'),
    metrics.CategoricalAccuracy(name='accuracy'),
    metrics.Precision(name='precision'),
    metrics.Recall(name='recall'),
    metrics.AUC(name='auc'),
    metrics.AUC(name='precision-recall curve', curve='PR')
]

# Compile ulang dengan loss dan optimizer yang sama
test_model.compile(optimizer=model_1.optimizer, # use the same
optimizer
                    loss=model_1.loss, # use the same loss
                    metrics=METRICS)

# Evaluasi model dengan data validation
val_data = list(validation_set_1.take(len(validation_set_1)))
val_images, val_labels = zip(*[(x.numpy(), y.numpy()) for x, y
in val_data])
val_images = np.concatenate(val_images)
val_labels = np.concatenate(val_labels)

test_predictions_baseline = test_model.predict(val_images,
batch_size=128)

# Evaluasi model dengan metric yang baru
baseline_results = test_model.evaluate(val_images, val_labels,
batch_size=128, verbose=0)

# Output hasil evaluasi dalam bentuk persentase
for name, value in zip(test_model.metrics_names,
baseline_results):
    if name in ["true positive", "true negative", "false
positive", "false negative"]:
        print(f"{name}: {value * 100:.2f}%")
    else:
        print(f"{name}: {value * 100:.2f}%")
print()

# menghitung precision dan recall

```

```

val_labels_argmax = np.argmax(val_labels, axis=1)
test_predictions_argmax = np.argmax(test_predictions_baseline,
axis=1)

precision = precision_score(val_labels_argmax,
test_predictions_argmax, average='weighted')
recall = recall_score(val_labels_argmax,
test_predictions_argmax, average='weighted')

print(f"Precision (sklearn): {precision * 100:.2f}%")
print(f"Recall (sklearn): {recall * 100:.2f}%")

# Fungsi untuk visualisasi confusion matrix
def plot_cm(label_matrix, predictions, class_names):
    preds = np.argmax(predictions, axis=1)
    labels_ = np.argmax(label_matrix, axis=1)
    cm = confusion_matrix(labels_, preds,
                           labels=np.arange(len(class_names)))
    cm_normalized = cm.astype('float') / cm.sum(
        axis=1)[:, np.newaxis] * 100
    annot = np.empty_like(cm).astype(str)
    nrows, ncols = cm.shape
    for i in range(nrows):
        for j in range(ncols):
            annot[i, j] = f'{cm[i, j]} ({cm[i, j] / cm.sum(
                axis=1)[j]:.2f}%)'
    plt.figure(figsize=(8, 8))
    sns.heatmap(cm, annot=annot, fmt="", cmap='copper',
                cbar_kws={'label': '% of Total'})
    indices = np.arange(len(class_names))
    plt.xticks(indices, class_names, rotation=45)
    plt.yticks(indices, class_names)
    plt.title('Confusion matrix')
    plt.ylabel('Actual label')
    plt.xlabel('Predicted label')

class_names = ["Cataract", "Diabetic Retinopathy", "Glaucoma",
               "Normal"]
plot_cm(val_labels, test_predictions_baseline, class_names)

```

USER ACCEPTANCE TESTING

Implementasi CNN Terhadap Sistem Klasifikasi Penyakit Pada Retina Mata Manusia
Menggunakan Model MobileNetV3 Berbasis Website

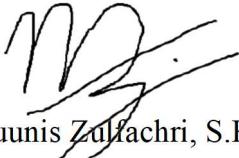
Nama : Naynuunis Zulfachri, S.Ked

Tempat : Daring

Pernyataan	Sangat Tidak Setuju	Tidak Setuju	Netral	Setuju	Sangat Setuju
Sistem tidak memiliki kesulitan dalam penggunaanya					✓
Sistem klasifikasi menghasilkan diagnosa penyakit dalam waktu yang lebih singkat		✓			
Sistem klasifikasi penyakit pada retina mata menghasilkan diagnosa penyakit sesuai dengan diagnosa pakar				✓	
Sistem klasifikasi penyakit pada retina mata layak digunakan sesuai dengan kebutuhan medis				✓	
Sistem layak dikembangkan untuk dapat mendiagnosis jenis penyakit yang lebih banyak					✓

Banda Aceh, 29 Mei 2024

Responden



(Naynuunis Zulfachri, S.Ked.)