

**IMPLEMENTASI *VISUAL GEOMETRY GROUP* (VGG19) UNTUK
MENGENALI POLA CITRA DALAM MENDETEKSI PENYAKIT MATA
KATARAK**

LAPORAN TUGAS AKHIR

Laporan ini disusun guna memenuhi salah satu syarat untuk menyelesaikan program studi Teknik Informatika S-1 pada Fakultas Teknologi Industri Universitas Islam Sultan Agung



Disusun oleh :

ANI LUTFIYAH

32602000071

**FAKULTAS TEKNOLOGI INDUSTRI
UNIVERSITAS ISLAM SULTAN AGUNG
SEMARANG**

2023

***IMPLEMENTATION OF VISUAL GEOMETRY GROUP (VGG19) TO
RECOGNIZE IMAGE PATTERNS IN DETECTING CATARACT EYE
DISEASE***

*This Report is Prepared to Fulfill One of the Requirements to Obtain
Bachelor Degree (S1) in the Informatics Engineering Study Program
Faculty of Industrial Technology, Sultan Agung Islamic University*



***MAJORING OF INFORMATICS ENGINEERING
INDUSTRIAL TECHNOLOGY FACULTY
SULTAN AGUNG ISLAMIC UNIVERSITY
SEMARANG
2023***

LEMBAR PENGESAHAN PEMBIMBING

Laporan Tugas Akhir dengan judul "Implementasi VGG19 Untuk Mengenali Pola Citra Dalam Mendeteksi Penyakit Mata Katarak" ini disusun oleh :

Nama : Ani Lutfiyah

NIM : 32602000071

Program Studi : Teknik Informatika

Telah disahkan oleh dosen pembimbing pada :

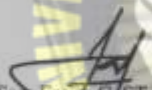
Hari : Sabtu

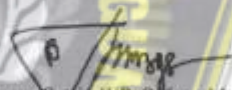
Tanggal : 04 Maret 2024

Mengesahkan,

Pembimbing I

Pembimbing II


Sam Fatma C / ST.M.Com
NIDN. 0628028602


Bagus Satrio-WP, S.Kom M.Cs
NIDN. 1027118301

Mengetahui,

Ketua Program Studi Teknik Informatika
Fakultas Teknologi Industri
Universitas Islam Sultan Agung


I. SE Mulyono, M.Eng
NIDN. 0626066601

LEMBAR PENGESAHAN PENGUJI

Laporan tugas akhir dengan judul "Implementasi VGG19 Untuk Mengenali Pola Citra Dalam Mendeteksi Penyakit Mata Katarak" ini telah dipertahankan di depan dosen penguji Tugas Akhir pada :

Hari : Senin
Tanggal : 4 Maret 2024



SURAT PERNYATAAN KEASLIAN TUGAS AKHIR

Yang bertanda tangan dibawah ini :

Nama : Ani Lutfiyah

NIM : 32602000071

Judul Tugas Akhir : Implementasi VGG19 Untuk Mengenali Pola Citra Dalam Mendeteksi Penyakit Mata Katarak

Dengan bahwa ini saya menyatakan bahwa judul dan isi Tugas Akhir yang saya buat dalam rangka menyelesaikan Pendidikan Strata Satu (S1) Teknik Informatika tersebut adalah asli dan belum pernah diangkat, ditulis ataupun dipublikasikan oleh siapapun baik keseluruhan maupun sebagian, kecuali yang secara tertulis dicatu dalam naskah ini dan disebutkan dalam daftar pustaka, dan apabila di kemudian hari ternyata terbukti bahwa judul Tugas Akhir tersebut pernah diangkat, ditulis ataupun dipublikasikan, maka saya bersedia dikenakan sanksi akademis. Demikian surat pernyataan ini saya buat dengan sadar dan penuh tanggung jawab.

Semarang, 29 Februari 2024

Yang Menyatakan,



Ani Lutfiyah

KATA PENGANTAR

Dengan mengucapkan syukur alhamdulillah atas kehadiran Allah SWT yang telah memberikan rahmat dan karuniaya kepada penulis, sehingga dapat menyelesaikan Tugas Akhir dengan judul “Implementasi *Visual Geometry Group* (VGG19) Untuk Mengenali Pola Citra Dalam Mendeteksi Penyakit Mata Katarak” ini untuk memenuhi salah satu syarat menyelesaikan studi serta dalam rangka memperoleh gelar sarjana (S-1) pada program Studi Teknik Informatika Fakultas Teknologi Industri Universitas Islam Sultan Agung Semarang.

Tugas Akhir ini disusun dan dibuat dengan adanya bantuan dari berbagai pihak, materi maupun teknis, oleh karena itu saya selaku penulis mengucapkan terima kasih kepada :

1. Rektor UNISSULA Bapak Prof. Dr. H. Gunarto, S.H., M.H yang mengizinkan penulis menimba ilmu dikampus ini.
2. Dekan Fakultas Teknologi Industri Ibu Dr. Ir. Novi Marlyana, S.T., M.T., IPU., ASEAN. Eng
3. Dosen Pembimbing I penulis Bapak Sam Farisa Chaerul Haviana, S.T., M.Kom yang memberikan banyak nasehat dan saran.
4. Dosen Pembimbing II penulis Bapak Bagus Waluyo Putro, S.Kom., M.Cs
5. Orang tua penulis, Mustahil saya mampu melewati semua permasalahan yang penulis alami selama ini jika tanpa doa, ridha dan dukungan dari beliau. Terima kasih bapak ibuk, berkatmu tugas akhir ini berjalan dengan lancar.
6. Kepada cinta kasih saudara-saudara saya, terutama kakak saya Sus Nafi'ah. Terima kasih telah memberikan semangat, dukungan dan motivasi serta terima kasih telah setia meluangkan waktunya untuk menjadi tempat dan pendengar terbaik bagi penulis sampai akhirnya penulis dapat menyelesaikan tugas akhir ini.
7. Untuk Dewi Netta Febrianti S.Pd. Sahabat penulis yang selalu mendengarkan cerita dan memberikan semangat bahkan nasehat sampai akhirnya penulis mampu bangkit kembali menyelesaikan tugas akhir.

8. Untuk Rakhmatulfitria Mekacahyani, Sahabat penulis selama menempuh pendidikan di dunia perkuliahan, terima kasih karna sudah membantu penulis dalam mengerjakan program tugas akhir. *See you on the next top*
9. Dan kepada semua pihak yang tidak dapat saya sebutkan satu persatu.

Dengan segala kerendahan hati, penulis menyadari masaih terdapat banyak kekurangan dari segi kualitas atau kuantitas maupun dari ilmu pengetahuan dalam penyusunan laporan, sehingga penulis mengharapkan adanya saran dan kritikan yang bersifat membangun demi kesempurnaan laporan ini di masa mendatang.

Semarang, Februari 2024



Ani Lutfiyah



DAFTAR ISI

COVER.....	ii
LEMBAR PENGESAHAN PEMBIMBING.....	iii
LEMBAR PENGESAHAN PENGUJI.....	iv
SURAT PERNYATAAN KEASLIAN TUGAS AKHIR.....	v
LEMBAR PERSETUJUAN PUBLIKASI KARYA ILMIAH.....	vi
KATA PENGANTAR.....	vii
DAFTAR ISI.....	ix
DAFTAR GAMBAR.....	xi
DAFTAR TABEL.....	xii
ABSTRAK.....	xiii
BAB I PENDAHULUAN.....	1
1.1 LATAR BELAKANG.....	1
1.2 RUMUSAN MASALAH.....	2
1.3 BATASAN MASALAH.....	2
1.4 TUJUAN.....	3
1.5 MANFAAT.....	3
1.6 SISTEMATIKA PENULISAN.....	3
BAB II TINJAUAN PUSTAKA DAN DASAR TEORI.....	5
2.1 Tinjauan Pustaka.....	5
2.2 Dasar Teori.....	7
2.2.1 Katarak.....	7
2.2.2 Penyebab Katarak.....	8
2.2.3 Convolutional Neural Networks.....	8
2.2.4 VGG19.....	10
2.2.5 Python.....	12
2.2.6 TensorFlow.....	13
2.2.7 Keras.....	13
2.2.8 Streamlit.....	14
2.2.9 Jupyter Notebook.....	14
2.2.10 Flowchart.....	15

BAB III METODE PENELITIAN.....	17
3.1 Metode Penelitian.....	17
3.2 Identifikasi Masalah	17
3.3 Studi Literatur.....	19
3.4 Pengumpulan Data.....	19
3.5 Perancangan Sistem.....	20
3.6 Implementasi	29
3.7 Pengujian	30
3.8 Evaluasi Sistem	34
BAB VI HASIL DAN ANALISIS PENELITIAN.....	36
4.1 <i>Deployment</i>	36
4.2 Hasil Model Layer.....	39
4.3 Hasil <i>Visualisasi</i> dan <i>Augmentasi</i> VGG19	40
4.4 Distribusi Data <i>Train, Val dan Test</i>	48
4.5 <i>Training</i> dan <i>Validation</i>	49
BAB V KESIMPULAN DAN SARAN.....	58
5.1 Kesimpulan.....	58
5.2 Saran.....	58
BAB IV DAFTAR PUSTAKA.....	60



DAFTAR GAMBAR

Gambar 2. 1 Arsitektur Convolutional Neural Network.....	9
Gambar 2. 2 Arsitektur Fully Connected Layer.....	10
Gambar 2. 3 Arsitektur VGG19 (Marcella dan Delvella 2022).....	11
Gambar 2. 4 Flowchart alur tahap preprocessing	15
Gambar 3. 1 Alur Penelitian.....	17
Gambar 3. 2 Sumber Dataset.....	19
Gambar 3. 3 Mengekstrak Gambar	20
Gambar 3. 4 Flowchart Alur Sistem.....	21
Gambar 3. 5 Menghubungkan Google Collab dengan Google Drive	22
Gambar 3. 6 Import Library	23
Gambar 3. 7 Image Preparation.....	24
Gambar 3. 8 Alur proses training VGG19.....	25
Gambar 3. 9 Hasil dataset training	25
Gambar 3. 10 Convulation Layer.....	26
Gambar 3. 11 Arsitektur CNN menggunakan VGG19	27
Gambar 3. 12 Code compile model.....	27
Gambar 3. 13 Training model cnn.....	27
Gambar 3. 14 Hasil Training epoch 30.....	28
Gambar 3. 15 Code evaluasi model.....	28
Gambar 3. 16 Data Citra Fundus Mata Normal.....	30
Gambar 3. 17 Data Citra Fundus Mata Katarak	30
Gambar 4. 1 Tampilan Halaman Utama	36
Gambar 4. 2 Halaman Upload Gambar	37
Gambar 4. 3 Hasil Upload Gambar	38
Gambar 4. 4 Halaman hasil Prediksi	38
Gambar 4. 5 Model yang diterapkan.....	39
Gambar 4. 6 Distribusi Train, Val dan Test.....	49
Gambar 4. 7 Pembagian dataset	49
Gambar 4. 8 nilai confusion matrix	55
Gambar 4. 9 Hasil klasifikasi CNN (VGG19).....	57

DAFTAR TABEL

Tabel 3. 1 Dataset.....	20
Tabel 3. 2 Analisis Tools Sistem.....	29
Tabel 3. 3 Klasifikasi confusion matrik.....	32
Tabel 3. 4 Black Box Testing	35
Tabel 4. 1 Visualisasi VGG19	41
Tabel 4. 2 Parameter.....	50
Tabel 4. 3 Grafik loss dan accuracy	50
Tabel 4. 4 Hasil Konfigurasi	53
Tabel 4. 5 Hasil pengujian dengan augmentasi data menggunakan optimasi Adam	55
Tabel 4. 6 Hasil Uji Coba.....	56



ABSTRAK

VGG19 atau *Visual Geometry Group 19* adalah arsitektur untuk jaringan saraf tiruan (CNN) 19 lapis. Jaringan ini biasanya digunakan untuk klasifikasi gambar dan tugas penyortiran gambar. Dalam konteks mengidentifikasi gangguan ekspresi wajah pada teks digital, VGG19 dapat digunakan sebagai salah satu komponen sistem pembelajaran adaptif untuk menganalisis isyarat visual yang terkait dengan kondisi tersebut. Implementasi ini menggunakan pelatihan model VGG19 pada dataset normal dan katarak agar dapat digabungkan. Model yang telah dilatih dan disempurnakan kemudian dapat diterapkan untuk menganalisis gambar baru dan mengidentifikasi kata kunci. Model VGG19 banyak digunakan dalam berbagai aplikasi komputer dan memiliki produktivitas tenaga kerja yang tinggi dalam tugas pengeditan gambar. Dengan menggunakan konfigurasi pengoptimal Adam, konfigurasi optimal diwakili oleh konfigurasi 2, yang memiliki akurasi 0,9956 dan kerugian 0,0134. Selain itu, terdapat akurasi validasi sebesar 1,6791 dan loss validasi sebesar 0,8190. Hasil evaluasi data uji akurasi menunjukkan loss sebesar 0,135 dan akurasi sebesar 91,66%.

Kata Kunci : VGG19, Mata Katarak

ABSTRAK

VGG19 or Visual Geometry Group 19 is an architecture for a 19-layer artificial neural network (CNN). These networks are commonly used for image classification and image sorting tasks. In the context of identifying facial expression disorders in digital text, VGG19 can be used as one component of an adaptive learning system to analyze the visual cues associated with these conditions. This implementation uses the training of the VGG19 model on normal and cataract datasets in order to combine them. The trained and refined model can then be applied to analyze new images and identify keywords. The VGG19 model is widely used in various computer applications and has high labor productivity in image editing tasks. Using the Adam optimizer configuration, the optimal configuration is represented by configuration 2, which has an accuracy of 0.9956 and a loss of 0.0134. In addition,

there is a validation accuracy of 1.6791 and a validation loss of 0.8190. The accuracy test data evaluation results show a loss of 0.135 and an accuracy of 91.66%.

Keywords : VGG19, Eye Cataract



BAB I PENDAHULUAN

1.1 LATAR BELAKANG

Indra merupakan alat dalam tubuh manusia yang dapat mengenali atau merasakan sesuatu dari lingkungan sepenuhnya yang kemudian diproses secara otomatis, sehingga manusia dapat memperoleh serta mengolah informasi mengenai lingkungan sepenuhnya. Pada umumnya manusia memiliki lima indra atau disebut panca indra, salah satunya adalah indra penglihatan atau mata. Mata merupakan indra yang berfungsi untuk menerima gambar visual lalu dikirim ke otak untuk diolah. Mata menjadi salah satu indra yang sering digunakan dalam kehidupan sehari-hari, seperti bekerja, bersekolah, dan lain lain. Salah satu penyakit mata adalah katarak (Peryanto, dkk 2019).

Katarak merupakan suatu penurunan fungsi progresif kejernihan pada lensa mata (Bu'ulolo 2021). Penyakit mata katarak merupakan salah satu masalah kesehatan yang umum terjadi di seluruh dunia, terutama pada populasi lanjut usia. Katarak adalah penyakit yang menyebabkan kekeruhan pada lensa mata, mengganggu penglihatan, dan dapat berpotensi menyebabkan kehilangan penglihatan jika tidak diobati dengan tepat. Deteksi dini dan pengobatan yang tepat sangat penting untuk mencegah dampak serius dari penyakit ini (Hananta, dkk 2022).

Salah satu pemeriksaan pada pasien yang mengalami atau terkena penyakit mata seperti penyakit mata katarak yaitu dengan cara mengambil gambar *fundoscopy* menggunakan alat *direct ophthalmoscope*. Analisis mata katarak atau normal pada citra *fundoscopy* masih cukup sulit bagi tenaga kesehatan sehingga memerlukan metode ilmu komputer yang tepat. Pada penelitian ini dilakukan klasifikasi penyakit mata katarak berdasarkan citra gambar *fundoscopy* menggunakan *Convolutional Neural Network* dengan arsitektur *Visual Geometry Group* (VGG19). Dataset yang digunakan berasal dari *cataract dataset* yang memiliki 600 citra gambar *fundoscopy*. Dari

dataset tersebut, terdapat 2 kelas yang diambil dan kemudian dari dataset ini dilakukan klasifikasi, yaitu cataract dan normal. Dataset dibagi masing-masing menjadi data latih dan data uji dengan perbandingan rasio 80:20 dengan *optimizer* yang digunakan yaitu Adam. Berdasarkan hasil pengujian, hasil klasifikasi terbaik didapatkan dengan *augmentasi* data latih terlebih dahulu yang memiliki akurasi sebesar 100%.

1.2 RUMUSAN MASALAH

Salah satu pertanyaan utama dalam tugas akhir ini adalah bagaimana menerapkan metode CNN menggunakan pola citra dengan arsitektur VGG19 untuk mengklasifikasikan penyakit katarak. Rumusan masalah dari tugas akhir ini adalah untuk mengidentifikasi setiap pertanyaan tersebut sebagai berikut:

1. Bagaimana metode VGG19 dapat diimplementasikan menggunakan pola citra untuk mengklasifikasikan penyakit mata katarak?
2. Bagaimana akurasi metode VGG19 menggunakan pola citra dalam mengklasifikasikan penyakit mata katarak?
3. Bagaimana perbandingan metode CNN menggunakan pola citra dengan arsitektur VGG19 lainnya dalam mengklasifikasikan penyakit mata katarak?

1.3 BATASAN MASALAH

Berdasarkan hasil penelusuran, berikut ini adalah batasan masalah dari usulan implementasi CNN untuk mengenali pola pada citra digital untuk mendeteksi katarak:

1. Metode VGG19 yang diusulkan akan berfokus pada penggunaan pola citra untuk mengklasifikasikan penyakit mata katarak.
2. Penelitian ini akan menggunakan dataset citra fundus mata katarak dan mata normal untuk melatih dan menguji model CNN arsitektur VGG19.

3. Metode yang diusulkan akan diimplementasikan sebagai aplikasi pembelajaran mesin yang dapat membantu mencegah kebutaan dan meningkatkan deteksi dini dan pengobatan penyakit mata katarak.

1.4 TUJUAN

Tujuan dari tugas akhir ini adalah menciptakan system deteksi penyakit mata katarak berbasis *web* dengan menerapkan metode CNN menggunakan arsitektur VGG19.

1.5 MANFAAT

Berdasarkan permasalahan dan tujuan penelitian, maka dapat diambil manfaat dari penelitian ini adalah :

1. Dapat membantu dalam mendeteksi dini penyakit mata katarak secara lebih mudah.
2. Memudahkan untuk segera melakukan Tindakan pencegahan lanjutan terhadap penyakit mata katarak.

1.6 SISTEMATIKA PENULISAN

Sistem penulisan yang akan penulis gunakan dalam laporan akhir tugas akhir adalah sebagai berikut:

BAB I : PENDAHULUAN

Dalam bab 1 ini, penulis membahas batasan-batasan penelitian, seperti rumusan masalah, tujuan penelitian, metodologi, dan sistematika penulisan..

BAB II : TINJAUAN PUSTAKA DAN DASAR TEORI

Adalah beberapa temuan penelitian sebelumnya dan kerangka teori yang berguna untuk membantu penulis dalam memahami bagaimana teori berhubungan dengan algoritma *Visual*

Geometry Group(VGG19) menggunakan Virtual untuk penelitian ini

BAB III : METODE PENELITIAN

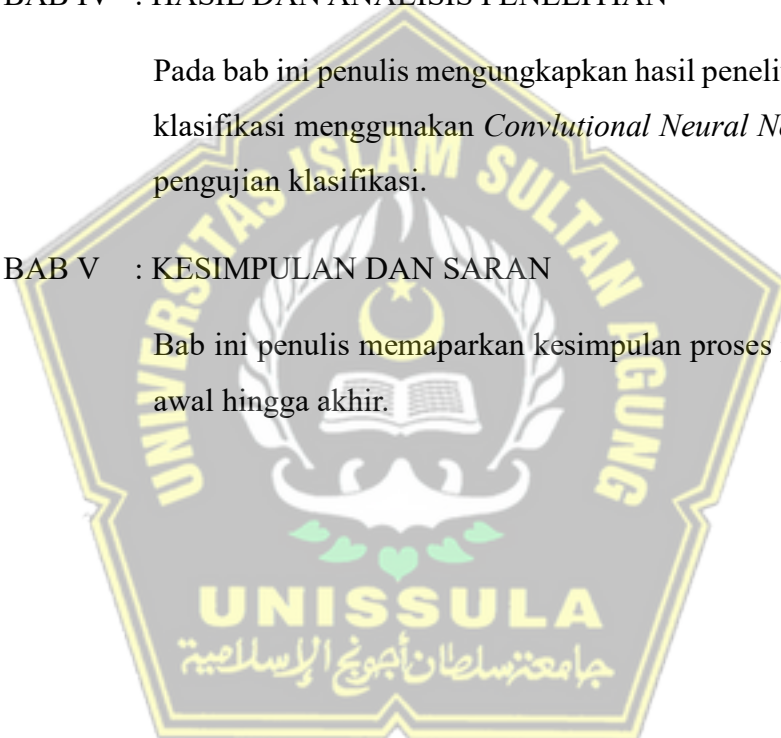
Pada bab ini menjelaskan langkah-langkah yang terlibat dalam proses penelitian, yang dimulai dengan mengumpulkan data dan diakhiri dengan klasifikasi data yang ada.

BAB IV : HASIL DAN ANALISIS PENELITIAN

Pada bab ini penulis mengungkapkan hasil penelitian yaitu hasil klasifikasi menggunakan *Convolutional Neural Network* beserta pengujian klasifikasi.

BAB V : KESIMPULAN DAN SARAN

Bab ini penulis memaparkan kesimpulan proses penelitian dari awal hingga akhir.



BAB II

TINJAUAN PUSTAKA DAN DASAR TEORI

2.1 Tinjauan Pustaka

Dalam sebuah penelitian yang telah dilakukan Marcella dan kawan – kawan mengatakan “katarak merupakan penyebab kerusakan mata tertinggi di Indonesia dengan persentase 0,78%, lalu disusul oleh glaukoma dengan persentase 0,20%, gangguan refraksi dengan persentase 0,14%, gangguan retina (terutama diabetic retinopathy) dengan persentase 0,13%, dan abnormalitas kornea (terutama Xerophthalmia) dengan persentase 0,10%”. Penelitian yang dilakukan oleh Dewi Marcella Menggunakan arsitektur *Convolutional Neural Network* (CNN) dengan arsitektur VGG-19 untuk mengklasifikasikan penyakit mata, termasuk katarak. Penelitian ini melibatkan 601 sampel, yang terdiri dari 430 sampel panjang, 50 sampel validasi, dan 121 sampel uji coba dengan total 4 kelas. Hasil dari penelitian ini menunjukkan bahwa metode CNN dengan arsitektur VGG-19 dapat diterapkan dengan baik dalam mengklasifikasikan penyakit mata, termasuk katarak (Marcella, Delvella 2022).

Dari tinjauan pustaka tersebut, dapat disimpulkan bahwa penelitian menggunakan metode CNN dan arsitektur VGG-19 dapat digunakan untuk mengklasifikasikan penyakit mata, termasuk katarak. Penelitian tersebut dapat membantu dalam membedakan antara mata katarak dan mata normal serta memudahkan klasifikasi jenis katarak berdasarkan kelainan patologis. Dalam penelitian tersebut menghasilkan klasifikasi Akurasi tertinggi didapatkan pada skenario kedua yaitu dengan menggunakan arsitektur VGG-19, clipLimit=1.0, optimizer Adagrad dan 150 iterasi dengan nilai akurasi sebesar 65,29% (Marcella, Devella 2022).

Dalam penelitian yang kedua yang dilakukan Diki Hananta Firdaus dan kawan-kawan, mereka melakukan klasifikasi penyakit katarak pada mata menggunakan metode *convolutional neural network* (CNN) berbasis web.

Hasil klasifikasi yang didapatkan cukup baik dengan akurasi 52,20%, epoch 10 mendapatkan akurasi sebesar 89,15% dan epoch 25 mendapatkan akurasi sebesar 99,74%. Berdasarkan hasil *training* dari beberapa epoch tersebut maka sistem mendapatkan hasil akurasi terbaik yaitu sebesar 99.74% (Hananta Firdaus dkk 2022).

Dalam penelitian yang ketiga yang dilakukan Geza Jeremia Bu'ulolo dan kawan-kawan, mereka melakukan penelitian Implementasi dari analisa *Convolutional Neural Network* mendapatkan hasil terbaik pada pengujian nilai parameter pertama dengan 3 lapisan konvolusi menggunakan *optimizer* Adam dan nilai *dropout* sebanyak 0.25 dan 5 serta 0.26 dan 6 dengan nilai akurasi yang didapatkan yaitu 93.31% pada data *training* dan 95% pada data *test* serta 82.55% pada data *training* dan 90% pada data *test* (Bu'ulolo 2021).

Dalam sebuah penelitian yang dilakukan oleh Abdul Jalil Rozaqi dan kawan-kawan, mereka melakukan klasifikasi penyakit pada daun kentang melalui sebuah citra menggunakan metode *Convolutional Neural Network*. Akurasi yang didapat pada epoch ke-10 dari 922 data latih dan 230 data testing adalah 95% pada *training* dan 94% pada validasi. Dengan demikian, metode yang diusulkan dapat mendeteksi kelainan pada mata tanpa menyebabkan kerusakan pada jenis penyakit mata tertentu. Metode yang diusulkan ini diharapkan dapat memberikan hasil yang dapat diandalkan, sehingga cocok untuk digunakan dalam sistem diagnosis berbantuan komputer dan membantu dalam mendeteksi penyakit mata katarak sejak dini untuk meningkatkan standar hidup masyarakat umum (Rozaqi, Sunyoto, Arielf n.d.).

Berdasarkan kelima hasil penelitian diatas penulis dapat mengambil kesimpulan bahwa arsitektur *Visual Geometry Group* (VGG19) memiliki kemampuan yang baik dalam mengidentifikasi objek dengan tingkat akurasi yang tinggi. Oleh karena itu, dalam penelitian ini nantinya akan membuat sebuah sistem yang dapat mendeteksi atau mengklasifikasi jenis penyakit mata katarak dengan menggunakan arsitektur VGG19.

2.2 Dasar Teori

2.2.1 Katarak

Penyakit mata katarak adalah salah satu masalah kesehatan mata yang sering ditemui di seluruh dunia. Katarak adalah kekeruhan lensa mata yang dapat mengaburkan penglihatan dan pada akhirnya dapat mengganggu kemampuan seseorang untuk melihat dengan jelas. Penanganan penyakit mata katarak memerlukan diagnosis dini yang akurat, sehingga pengobatan yang tepat dapat segera diberikan. Diagnosa katarak biasanya dilakukan oleh dokter mata melalui pemeriksaan mata yang canggih, tetapi dalam beberapa kasus, pemeriksaan ini mungkin tidak praktis atau tersedia, terutama di daerah yang kurang berkembang (Android dan Meltodel).

Banyak orang yang tidak menyadari bahwa mereka menderita katarak karena sulit untuk membedakan antara katarak normal dan katarak di lingkungan sepenulisnya. Pada awalnya, sangat penting untuk mengidentifikasi jenis huruf sebelum masalah muncul. Untuk itu, diperlukan sebuah sistem yang dapat membedakan antara teks yang bersifat tekstual dan teks yang tidak bersifat tekstual.

Oleh karena itu, tujuan dari penelitian ini adalah untuk mengimplementasikan model jaringan saraf tiruan yang kuat, yaitu VGG19, dalam pengenalan pola citra untuk mendeteksi penyakit mata katarak. Model jaringan saraf tiruan dapat digunakan untuk mengotomatisasi proses diagnostik dengan mengidentifikasi ciri-ciri katarak pada gambar mata, seperti kekeruhan lensa dan perubahan warna. Dengan demikian, penelitian ini dapat membantu dalam diagnosis dini katarak dan memungkinkan tindakan pengobatan yang lebih cepat.

Dengan kemajuan teknologi, sidik jari digital telah membuat analisis dan klasifikasi katarak menjadi lebih akurat. Pada tugas akhir ini, penulis membuat aplikasi *machine learning* untuk membandingkan katarak dan normal dengan menggunakan metode *Convolutional Neural Network* (CNN) berbasis web (Risma, Patmasari, dan Magdalena, 2019). Data

diambil dari sebuah situs web bernama Kaggle yang terkenal di bidang data science. Dataset yang digunakan terdiri dari 512 citra digital dengan dua subset : katarak dan normal. Hasil akhir yang diharapkan adalah model yang dibangun dapat membedakan antara mata normal dan katarak. (Hananta Firdaus dkk. 2022).

2.2.2 Penyebab Katarak

Sebagai akibat dari bertambahnya usia penyebab kondisi ini, lensa menjadi lebih kaku dan kurang fleksibel. Hal ini menyebabkan kerusakan protein dan berkurangnya pembuluh darah yang masuk ke retina, sehingga menciptakan lapisan sensitif pada permukaan kulit Anda. Pada gilirannya, hal ini menyebabkan pandangan menjadi kabur dan tidak tajam. Keausan lensa pada awalnya ditandai dengan peringatan kecoklatan ringan, tetapi pada akhirnya akan hilang seiring berjalannya waktu. Secara perlahan-lahan, penderita akan mulai sulit mengenali warna ungu atau biru.

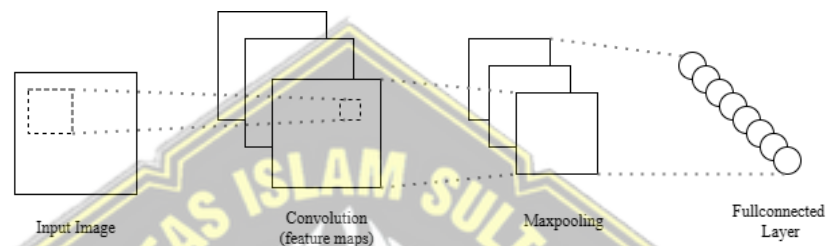
Dengan kemajuan teknologi yang semakin hari semakin canggih dan tidak beraturan, maka diperlukan suatu sistem pakar yang dapat memelihara pakar ahli dengan harapan dapat membantu masyarakat dalam lebih mengetahui penyakit pakar sebelum berkembang. Adapun metode yang digunakan untuk mendiagnosa penyakit katarak adalah metode *Certainty Factor*, yang menggunakan CF untuk menghitung kepastian dari fakta yang diberikan yang belum pasti dan memperoleh tingkat presentasi dengan nilai 95%. Oleh karena itu, penggunaan metode *Certainty Factor* dapat digunakan untuk memahami masing-masing setiap permasalahan (Fahmi 2019).

2.2.3 Convolutional Neural Networks

Convolutional Neural Networks adalah pengembangan dari *Multilayer Perceptron* (MLP) yang dirancang untuk membagi input dua dimensi. CNN diklasifikasikan sebagai *Deep Neural Network* karena kedalaman jaringan yang tinggi dan banyak diaplikasikan pada data citra (Athallah, dkk). CNN

pertama diciptakan dengan nama NeoCognitron oleh peneliti Kunihiko Fukushima dari NHK *Broadcasting Science Research Laboratories* di Kinuta, Setagaya, Tokyo, Jepang (Nurona Cahya dkk).

CNN juga adalah *supervised learning* yaitu model di-*training* terlebih dahulu untuk mengenali pola antara input data dan label output. Arsitektur CNN yang digunakan pada penelitian ini yaitu arsitektur VGG19. *Convolutional Neural Networks* (CNN) sendiri terdapat beberapa layer, diantaranya *Convolution Layer*, *Maxpooling Layer*, *Fully Connected layer*



Gambar 2. 1 *Arsitektur Convolutional Neural Network*

Gambar 2.1 pada arsitektur CNN tersebut, sebuah gambar masukan dideteksi atribut atau fiturnya dengan menggunakan konvolusi tiga filter. Hasil dari konvolusi disebut *feature map* yang kemudian akan dilakukan *max pooling* dan akan menghasilkan tiga buah gambar baru dengan resolusi yang lebih kecil. Terakhir, hasil *max pooling* dimasukkan ke dalam *fully connected layer*.

a. *Convolutional Layer*

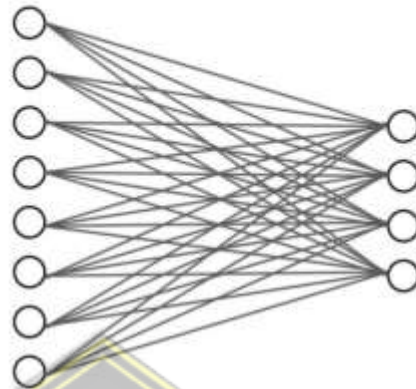
Convolutional Layer merupakan layer pertama kali yang menerima input. Operasi pada layer ini adalah operasi konvolusi (matriks yang berfungsi untuk melakukan filter).

b. *Maxpooling Layer*

Tujuan dari lapisan *maxpooling* adalah untuk mengurangi ukuran matriks sehingga mempercepat proses komputasi karena parameter yang digunakan dalam proses selanjutnya menjadi lebih sedikit dan tidak rentan terhadap *overfitting*.

c. *Fully Connected layer*

Fully Connected



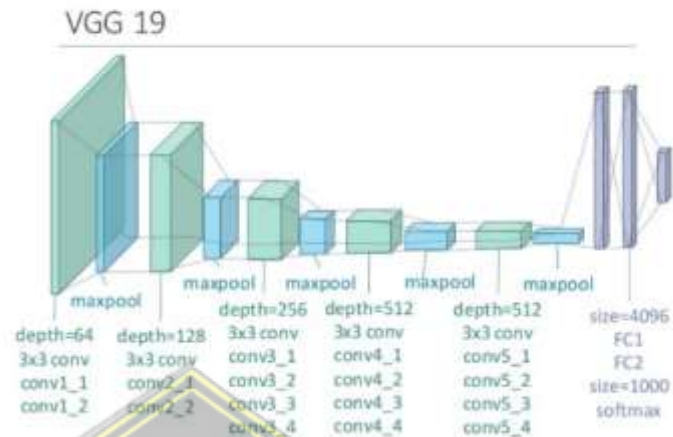
Gambar 2. 2 Arsitektur *Fully Connected Layer*

Layer tersebut merupakan layer yang biasanya dipakai dalam penerapan MLP dan bertujuan untuk melakukan transformasi pada dimensi data agar dapat diklasifikasikan secara linear. Disetiap neuron pada convolution layer perlu ditransformasi menjadi data satu dimensi terlebih dahulu sebelum dapat dimasukkan kedalam sebuah fully connected layer. Dengan itu menyebabkan kehilangan data informasi spasialnya dan menjadi tidak reversibel, fully connected layer hanya bisa diimplementasikan di akhir jaringan. Convolution layer dengan ukuran kernel 1×1 menjalankan fungsi yang sama dengan sebuah fully connected layer namun dengan tetap mempertahankan karakter spasial dari data (Hananta Firdaus dkk. 2022)

2.2.4 VGG19

Visual Geometry Group (VGG19) adalah salah satu arsitektur *Convolutional Neural Network* (CNN) yang digunakan dalam pengolahan citra. Arsitektur ini terdiri dari 19 layer, termasuk 16 layer konvolusi (*convolutional layers*) dan 3 layer *fully connected* (FC layers). Arsitektur VGG19 telah digunakan dalam beberapa penelitian untuk mengklasifikasikan penyakit mata, termasuk katarak.

Define input (feature) and output (label)



Gambar 2. 3 Arsitektur VGG19 (Marcellla dan Delvellla 2022).

Arsitektur VGG19 terdiri dari banyak lapisan konvolusi yang digunakan untuk mengekstrak fitur-fitur dari gambar. Konvolusi adalah operasi matematis yang memungkinkan jaringan saraf tiruan untuk mengidentifikasi pola dan fitur dalam gambar. Jaringan VGG19 menggunakan konvolusi berukuran kecil (biasanya 3x3) dengan jarak 1 piksel untuk mengurangi dimensi gambar secara bertahap.

Setelah setiap lapisan konvolusi, VGG19 mengikuti max-pooling layer untuk mengurangi dimensi gambar. *Max-pooling* adalah operasi yang memilih nilai terbesar dalam area tertentu dari gambar yang disebut kernel. Ini membantu mengurangi jumlah parameter dalam model dan memungkinkan jaringan untuk fokus pada fitur-fitur penting.

Setelah sejumlah lapisan konvolusi dan *max-pooling*, VGG19 memiliki beberapa lapisan *fully connected* (FC) atau lapisan terhubung penuh. Lapisan-lapisan ini digunakan untuk menghubungkan hasil dari lapisan konvolusi ke output kelas yang diinginkan. Biasanya, ada dua atau tiga lapisan FC terakhir yang menghasilkan prediksi kelas akhir.

VGG19 menggunakan kegunaan aktifitas ReLU (*Rectified Linear Unit*) pada setiap lapisan konvolusi dan FC, kecuali pada lapisan *output*. ReLU adalah fungsi aktivasi yang sederhana, yang mengubah semua nilai negatif menjadi nol dan membiarkan nilai positif tetap tidak berubah. Untuk menghindari *overfitting* (ketika model belajar "mengingat" data pelatihan daripada "mengeneralisasi" dari data), VGG19 menggunakan *dropout* pada lapisan-lapisan *fully connected*. *Dropout* secara acak mematikan sebagian unit (*neuron*) dalam lapisan ini selama pelatihan, sehingga memaksa jaringan untuk belajar dengan cara yang lebih generalisasi.

Pada lapisan *output*, VGG19 biasanya menggunakan fungsi aktivasi sigmoid untuk menghasilkan probabilitas prediksi kelas. Fungsi sigmoid adalah fungsi matematika yang menghasilkan output antara 0 dan 1. Fungsi ini umumnya digunakan dalam machine learning untuk klasifikasi biner. Dalam konteks jaringan saraf tiruan, fungsi sigmoid sering digunakan sebagai fungsi aktivasi karena kemampuannya untuk memetakan nilai input ke rentang antara 0 dan 1, yang berguna untuk tugas seperti klasifikasi biner dan estimasi probabilitas.

2.2.5 Python

Salah satu bahasa pemrograman yang populer dan berguna adalah Python. Bahasa ini dapat digunakan untuk berbagai sistem arsitektur dan digunakan di berbagai bidang seperti pengembangan web, ilmu data, dan pembelajaran mesin. *Python* mudah dipelajari karena menggunakan sintaks yang sederhana dengan banyak kata kunci dalam bahasa Inggris, serta sintaks yang sederhana dan mudah dipahami. Di antara keunggulan lainnya adalah kemampuannya untuk berjalan di berbagai sistem operasi, kecepatan penetikannya yang kuat, dan penekanannya pada kemudahan pengguna. *Python* juga merupakan bahasa pemrograman yang banyak digunakan yang menawarkan kesempatan belajar yang baik di beberapa industri. *Python* memfasilitasi pengembangan aplikasi yang kompleks untuk berbagai

kebutuhan, termasuk pengembangan web, analisis data, peningkatan proses bisnis, dan banyak lagi.

2.2.6 TensorFlow

TensorFlow adalah *library open-source* yang dikembangkan oleh Google yang digunakan untuk melatih dan menganalisis berbagai model dalam pembelajaran mesin, pembelajaran mendalam, dan pekerjaan analitis tingkat lanjut untuk ahli statistik dan ilmuwan data. TensorFlow dirancang untuk mempercepat pengembangan aplikasi analitik jangka panjang. *TensorFlow* mampu menangani set data dengan node dan edge dalam bentuk grafis. *Edge* membuat koneksi antara setiap node dalam grafik vektor dan kemudian menentukan apa yang dikenal sebagai tensor. *TensorFlow* sering digunakan dengan jaringan syaraf untuk melakukan berbagai tugas, seperti pengenalan gambar, pengenalan teks, dan pemrosesan bahasa alami (NLP). Meskipun *TensorFlow* dapat digunakan untuk banyak tugas, fokus utamanya adalah pada pembelajaran dan pelatihan mesin. Selain itu, TensorFlow menawarkan bantuan kepada para pengembang yang perlu melakukan debugging dan penelitian prediksi berskala besar. *Library* ini cukup populer dan digunakan untuk berbagai keperluan, seperti pencarian gambar, terjemahan teks ke suara, analisis data, dan lain-lain.

2.2.7 Keras

Keras adalah API jaringan saraf tingkat tinggi yang ditulis dalam bahasa *Python* yang beroperasi di atas *TensorFlow*, CNTK, atau Theano. Pengguna dapat dengan mudah membuat, mempelajari, dan mengevaluasi model jaringan syaraf berkat Keras. Keras menawarkan alat pembuatan model jaringan saraf yang intuitif dan mudah digunakan, sehingga pengguna dapat berkonsentrasi pada model dan desain eksperimen. Keras juga menyediakan beberapa fitur yang berguna, seperti dukungan untuk berbagai jenis lapisan, fungsi aktivasi, dan pengoptimal. Keras dapat digunakan untuk berbagai

tugas, seperti pemrosesan bahasa alami (NLP), pemrosesan gambar, dan pemrosesan teks. Perangkat lunak ini dibuat oleh François Chollet dan memiliki lisensi sumber terbuka.

2.2.8 *Streamlit*

Streamlit adalah *library open-source* yang dapat digunakan untuk mengembangkan pembelajaran mesin dan aplikasi web data besar dengan mudah. Platform ini memungkinkan pengguna untuk dengan cepat dan mudah membuat aplikasi online yang menampilkan data, grafik, tabel, dan visualisasi lainnya. Selain itu, *Streamlit* dilengkapi dengan beberapa fitur yang memungkinkan pengguna untuk melakukan kontrol terhadap aplikasi web seperti input teks, checkbox, choose box, dan lain sebagainya. Dengan menggunakan *Streamlit*, pengguna dapat mengakses berbagai macam tipe data, seperti data dalam file CSV atau Excel, data dari database SQL, dan bahkan data melalui API online. Platform ini juga memiliki kemampuan refresh otomatis, yang memudahkan pengguna untuk melakukan iterasi dan memelihara aplikasi web dengan cepat tanpa harus melakukan prosedur refresh manual. *Streamlit* juga memiliki komunitas yang besar dan aktif.

2.2.9 *Jupyter Notebook*

Jupyter Notebook merupakan sebuah platform aplikasi web yang bersifat *open-source*. Platform ini didasarkan pada pembelajaran mesin (ML), pemodelan statistik, visualisasi data, transformasi data, dan pemodelan numerik. Untuk meningkatkan kualitas data, *Jupyter Notebook* menyediakan lingkungan komputasi yang interaktif. Pengguna dapat melampirkan kode untuk perangkat lunak, output komputer, teks penjelasan, dan konten dokumen dengan *Jupyter Notebook*. *Python*, *R*, *Julia*, dan *Scala* adalah beberapa di antara sekian banyak bahasa pemrograman yang didukung oleh platform ini. *Jupyter Notebook* juga memungkinkan pengguna untuk mengeksekusi kode dan melihat hasil komputer secara

diam-diam di server web. Selain itu, *Jupyter Notebook* memfasilitasi kemampuan pengguna untuk membuat grafik secara tidak terstruktur, sehingga memudahkan para data scientist untuk mendokumentasikan proses analisis mereka dan berbagi hasil kerja mereka dengan rekan kerja. *Jupyter Notebook* juga dapat digunakan untuk berbagai tugas pembelajaran mesin dan kecerdasan buatan, serta untuk memfasilitasi penelitian, kolaborasi, dan eksperimen. Platform ini juga memfasilitasi integrasi dengan berbagai perangkat big data, seperti Apache Spark yang berbasis Python, R, dan Scala.

2.2.10 Flowchart

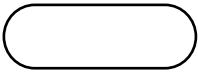


Flowchart merupakan penggambaran secara grafik dari langkah-langkah dan urutan prosedur suatu program. Biasanya mempengaruhi penyelesaian masalah yang khususnya perlu dipelajari dan dievaluasi lebih lanjut (Inovasi Penelitian dkk).

Flowchart dapat digunakan untuk menyajikan kegiatan manual, kegiatan pemrosesan ataupun keduanya. Flowchart merupakan rangkaian symbol-symbol yang digunakan untuk mengkontruksi. Symbol yang digunakan sebagai berikut :



Gambar 2. 4 Flowchart alur tahap *preprocessing*

Pada gambar 2. 4 merupakan flowchart alur *preprocessing* dimana tahapannya adalah :

No	Simbol	Keterangan
1		Terminator Symbol - Yaitu symbol untuk permulaan (start) atau akhir (stop) dari suatu kegiatan
2		<i>Processing Symbol</i> - Yaitu symbol yang menunjukkan pengolahan yang dilakukan oleh komputer
3		<i>Symbol Input - Output</i> - Yaitu symbol yang menyatakan proses input dan output tanpa bergantung dengan jenis peralatannya



BAB III METODE PENELITIAN

3.1 Metode Penelitian

Dalam tahapan penelitian, algoritma atau metode yang digunakan dalam klasifikasi penyakit katarak pada mata katarak menggunakan *Convolutional Neural Network* (CNN). Metode penelitian yang digunakan dalam penulisan ilmiah ini adalah sebagai berikut :



Gambar 3. 1 Alur Penelitian

3.2 Identifikasi Masalah

Identifikasi masalah merupakan langkah awal yang penting dalam proses penelitian karena membantu peneliti untuk memahami masalah yang ingin dipecahkan. Dalam konteks penelitian tentang penerapan *Visual Geometry Group* (VGG19) untuk mengidentifikasi pola citra dalam deteksi penyakit katarak, ada beberapa masalah utama yang perlu diidentifikasi antara lain.

- a. Krisis Kesehatan Masyarakat: Katarak merupakan salah satu krisis kesehatan yang paling signifikan di seluruh dunia, terutama di negara-negara berkembang. Identifikasi katarak secara dini sangat penting untuk mengurangi dampaknya terhadap kualitas hidup penderitanya. Oleh karena itu, diperlukan pengembangan metode deteksi yang efektif dan cepat.
- b. Deteksi Kemanusiaan: Meskipun para ahli medis dapat mendeteksi gumpalan darah melalui penusukan kulit, metode ini sering kali tidak akurat dan membutuhkan waktu yang lama. Variabilitas juga ada dalam kemampuan deteksi antara satu ahli dengan ahli lainnya.
- c. Ketersediaan data dan anotasi: Untuk melatih model pembelajaran mendalam seperti VGG19, diperlukan sejumlah besar data yang beranotasi dengan baik. Salah satu masalah potensial adalah ketersediaan dataset yang bersih dengan pelabelan yang akurat dan representatif.
- d. Paket Citra Visual Lengkap Mata: Mata adalah organ yang kompleks dengan struktur yang kasar. Membandingkan gambar mata biasa dan katarak dapat membingungkan, terutama karena ada perbedaan individual dalam anatomi mata dan variasi pencahayaan pada gambar.
- e. Teknologi: Menggunakan model pembelajaran mendalam seperti VGG19 dalam pengenalan suara membutuhkan pengetahuan tentang teknologi yang relevan dan bagaimana menerapkannya dengan tepat dalam konteks medis. Selain itu, skalabilitas dan integrasi dengan sistem medis yang ada juga merupakan faktor penting.
- f. Validasi dan Evaluasi: Penting untuk memvalidasi dan menilai kinerja model yang dicapai secara komprehensif. Hal ini termasuk melakukan pemeriksaan klinis untuk memastikan bahwa model yang dikembangkan dapat diterapkan dalam praktik medis sehari-hari.

Setelah mengidentifikasi permasalahan tersebut, penelitian mengenai penerapan VGG19 untuk menganalisis pola citra dalam pendeteksian

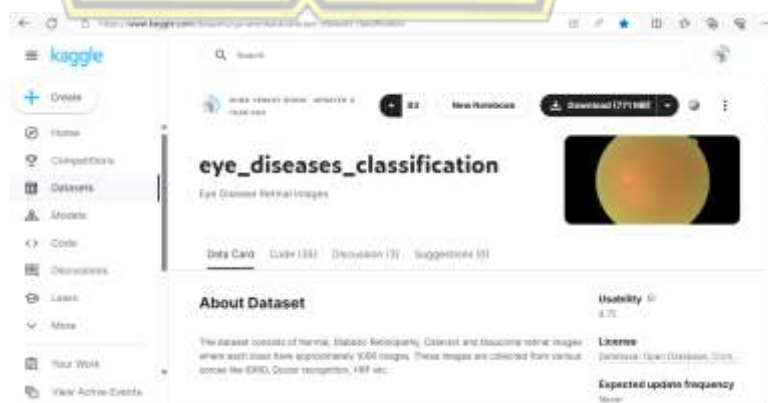
penyakit katarak dapat diarahkan untuk mengatasi permasalahan tersebut dengan meningkatkan efisiensi pendeteksian dini dan pelaksanaan penyakit katarak.

3.3 Studi Literatur

Penulis akan meninjau beberapa e-book, artikel, jurnal, hasil penelitian terdahulu seperti tesis dan skripsi, serta berbagai situs website untuk bisa mempelajari teori dan berlatih mengenai *Convolutional neural networks*, *machine learning*, *deep learning*, dan *image classification* dari berbagai sumber yang ada seperti halnya artikel, jurnal, serta berbagai situs *website* yang ada di Internet.

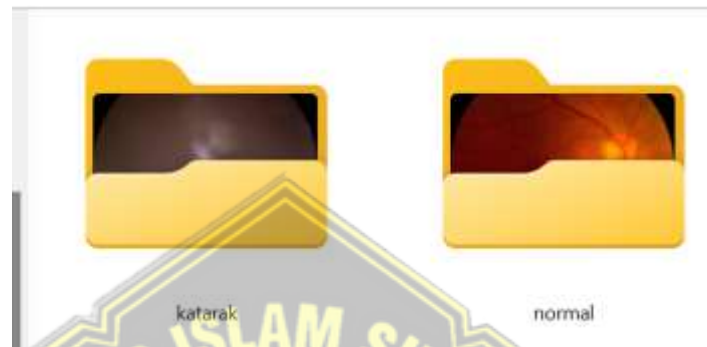
3.4 Pengumpulan Data

Pada tahap ini, penulis mengumpulkan data yang diperlukan terkait penelitian yang akan dilakukan dengan menggunakan dataset yang berasal dari penelitian sebelumnya. Dataset yang dimaksud berasal dari Kaggle (*eye_diseases_classification* ([kaggle.com](https://www.kaggle.com/abdulrahman123/eye-diseases-classification))) dan memiliki total 600 entri. Dataset yang digunakan terdiri dari citra fundus mata yang dibagi menjadi dua kategori yaitu normal dan katarak. Setiap kelas memiliki jumlah data yang berbeda, terdiri dari 300 citra untuk kelas normal dan 300 citra untuk kelas katarak.



Gambar 3. 2 Sumber Dataset

Pada gambar 3.2 merupakan Dataset yang bersumber dari eye diseases classification (kaggle.com). Dataset tersebut digunakan akan dibagi menjadi dua tahap pembelajaran, yaitu pembelajaran fitur dan klasifikasi, dengan total 600 citra yang digunakan dalam dataset.



Gambar 3. 3 Mengekstrak Gambar

Pada gambar 3.3 merupakan pembagian label penyakit mata yaitu terbagi menjadi dua yaitu Normal dan Katarak yang nantinya akan diproses ke tahap selanjutnya.

Tabel 3. 1 Dataset

TABEL JUMLAH GAMBAR <i>DATASET</i>			
	<i>Training</i>	<i>Test</i>	<i>Val</i>
Katarak	240	30	30
Normal	240	30	30
Total	480	60	60

Pada tabel 3.1 merupakan pembagian Dataset untuk proses klasifikasi, yaitu dibagi untuk *Training*, *Test*, *Val*.

3.5 Perancangan Sistem

Penulis pada tahap ini akan menentukan alur kerja system berupa flowchart yang akan memberikan gambaran dari alur system nantinya. Dimana metode ini nantinya akan melakukan klasifikasi citra mata baru, dan

ada beberapa tahapan yang perlu dilakukan dalam penelitian ini, seperti yang terlihat pada gambar 3.4 :



Gambar 3. 4 *Flowchart* Alur Sistem

Pada gambar 3.4 merupakan flowchart alur *preprocessing* dimana tahapannya adalah :

1. *Ekstrak Dataset* : Proses ekstraksi dataset dari file ZIP merupakan langkah penting dalam pengelolaan data, terutama dalam konteks pengembangan model klasifikasi seperti pada penelitian tentang klasifikasi mata katarak dan normal menggunakan histogram.
2. User memasukkan citra dari galeri : Proses user memasukkan citra ke galeri, sistem hanya bisa menggunakan format jpg, jpeg dan png.
3. Proses klasifikasi citra : Proses klasifikasi citra merupakan teknik untuk mengkategorikan piksel, objek, atau potongan gambar ke dalam kelas tertentu. Awalnya, dominasi metode klasifikasi citra penginderaan jauh adalah pendekatan berbasis piksel, di mana analisis dilakukan pada tingkat piksel individual. Namun, dengan perkembangan teknologi, muncul paradigma baru dalam bentuk metode klasifikasi berbasis objek dan berbasis pemandangan. Metode berbasis objek mempertimbangkan konteks spasial antara

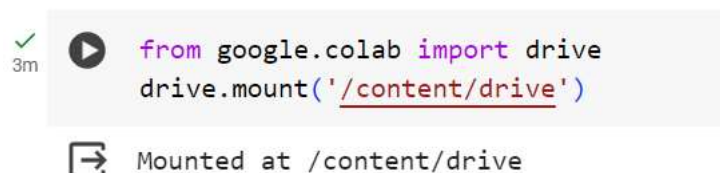
piksel, sementara metode berbasis pemandangan memberikan pemahaman holistik tentang citra dan konteksnya (Ambarwari, Husni, dan Mahayana 2023).

Alur proses sistem ini dimulai ketika user membuka website sistem. Setelah user berhasil masuk ke halaman utama, langkah selanjutnya adalah memasukkan citra mata katarak yang akan dideteksi. Setelah citra dimasukkan, user diharuskan menekan tombol 'Upload' untuk memulai proses klasifikasi oleh sistem. Selanjutnya, sistem akan melakukan klasifikasi berdasarkan citra mata katarak yang telah dimasukkan dan menampilkan hasil akurasi mata katarak. Setelah semua proses tersebut selesai, alur proses sistem penelitian ini telah berhasil diselesaikan.

3.5.1. Dataset

Pada penelitian ini, Google Collab digunakan sebagai platform untuk membuat aplikasi klasifikasi kerusakan teks berbasis website. Di sisi lain, untuk mengekspor data, gunakan aplikasi Google Drive. Dataset telah diunggah ke Google Drive. Saat mengembangkan aplikasi, Google Collab dan Google Drive harus terhubung. Kode untuk menghubungkan Google Drive dan Google Collab ditunjukkan pada Gambar 3.3.

Untuk menghubungkan Google Drive dan Google Collab, seseorang dapat mengikuti petunjuk yang diberikan di notebook atau menggunakan cuplikan kode berikut :



```

✓ 3m ▶ from google.colab import drive
      drive.mount('/content/drive')

↳ Mounted at /content/drive

```

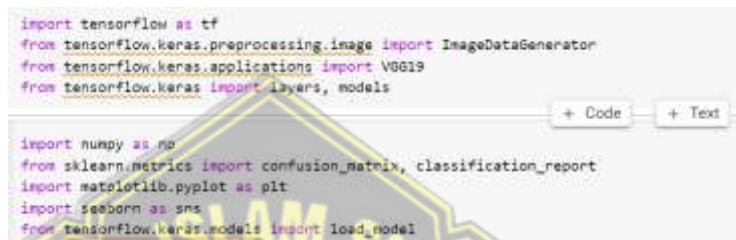
Gambar 3. 5 Menghubungkan Google Collab dengan Google Drive

Kode ini akan meminta pengguna untuk mengotorisasi akses Google Drive dan menghasilkan kode otentikasi yang perlu dimasukkan ke dalam

buku catatan. Setelah autentikasi selesai, Google Drive dapat diakses dari Google Collab menggunakan jalur /content/drive/My Drive/

3.5.2. *Import Library*

Pada langkah ini, penulis akan menggunakan library Python yang diperlukan TensorFlow, Keras, OpenCV, dan lainnya. Untuk membuat sistem klasifikasi penyakit katarak di bidang ini. Gambar 3.4 adalah kode yang digunakan untuk mengatur library yang digunakan.



```
import tensorflow as tf
from tensorflow.keras.preprocessing.image import ImageDataGenerator
from tensorflow.keras.applications import VGG19
from tensorflow.keras import layers, models

import numpy as np
from sklearn.metrics import confusion_matrix, classification_report
import matplotlib.pyplot as plt
import seaborn as sns
from tensorflow.keras.models import load_model
```

Gambar 3. 6 *Import Library*

Kode tersebut berisi semua modul dan fungsi yang diperlukan untuk membuat, mempelajari, mengevaluasi, dan membuat model klasifikasi gambar menggunakan arsitektur VGG19.

3.5.3. *Image Preparation*

Image Preparation adalah proses mempersiapkan gambar atau bagan sebelum pekerjaan yang lebih luas dilakukan. Proses ini mencakup akuisisi citra atau pengambilan gambar, *preprocessing* atau pengolahan awal seperti scaling dan konversi warna, serta peningkatan kualitas gambar atau *image enhancement*. Operasi lain seperti restorasi atau pemulihan citra, kompresi citra, representasi dan pemodelan citra, serta representasi juga dapat disertakan dalam persiapan citra. Langkah pertama dalam pemrosesan gambar, atau persiapan gambar, disebut pemrosesan gambar.

Dataset yang sudah terhubung dengan Google Drive akan diekstrak dan diubah menjadi sebuah variabel untuk mempercepat proses klasifikasi CNN. Proses persiapan ini adalah ketika data akan dianalisis dengan meningkatkan *Train*, *Test*, dan *Valid* menjadi variabel. Setelah itu, *target_size* dari dataset grafik yang akan digunakan akan dipertimbangkan. Setelah itu, atur *batch_size* dan *class_mode*. *Batch_size* yang digunakan adalah 32, dan *class*

mode menggunakan binary karena peneliti hanya menggunakan tiga kelas yang digunakan adalah binary.

```
# menyiapkan data training dengan pengaturan yang sudah ditentukan
train_data = train_datagen.flow_from_directory(train_data_dir,
                                              target_size=img_size,
                                              batch_size=32,
                                              class_mode='binary',
                                              seed=42)

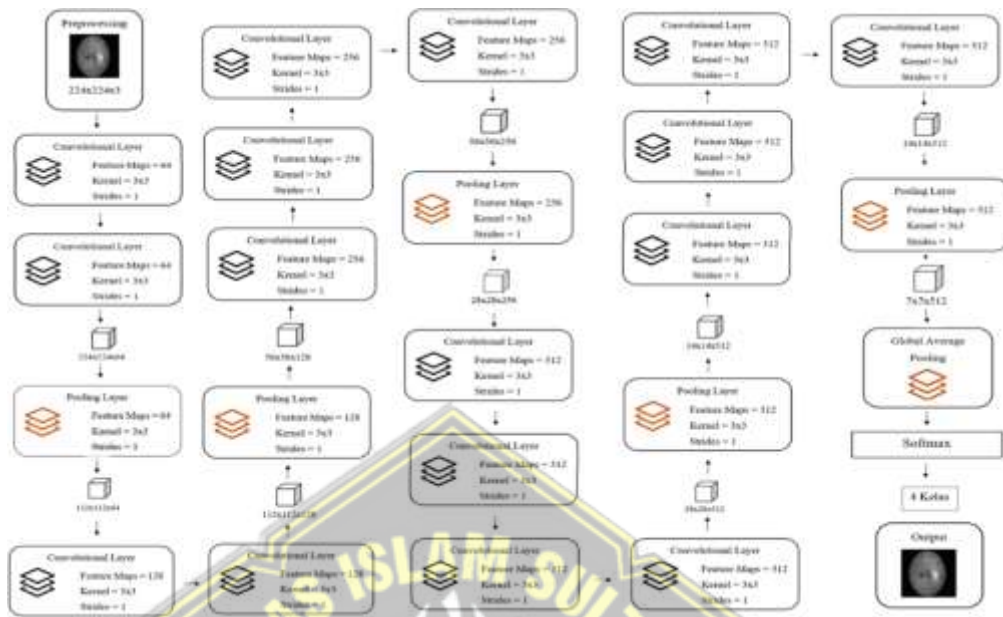
# menyiapkan data validasi
val_data = val_datagen.flow_from_directory(val_data_dir,
                                          target_size=img_size,
                                          batch_size=32,
                                          class_mode='binary',
                                          seed=42)

# menyiapkan data uji
test_data = test_datagen.flow_from_directory(test_data_dir,
                                             target_size=img_size,
                                             batch_size=32,
                                             class_mode='binary',
                                             shuffle=False)
```

Gambar 3. 7 Image Preparation

3.5.4. Training Dataset

Ini adalah prosedur di mana CNN menjalani pelatihan untuk mendapatkan tingkat akurasi yang tinggi. Ini adalah langkah pertama dalam proses CNN, ketika proses ini digunakan untuk membuat model yang nantinya akan digunakan dalam proses pengujian. Terdapat dua proses dalam langkah ini yaitu klasifikasi menggunakan feedforward dan pembelajaran menggunakan backpropagation. Proses *feedforward* dimulai ketika dataset dibersihkan. Proses kerja *feedforward* melibatkan vektor yang menggunakan proses *max pooling* dan *convolucency* untuk mengurangi radius dan meningkatkan jumlah neuron. Dengan demikian, ada banyak jaringan yang memungkinkan data yang berbeda untuk dipelajari. Hasil dari proses *feedforward* adalah bobot yang akan digunakan untuk menilai proses *neural network* itu sendiri. Alur dataset pelatihan dapat ditunjukkan pada Gambar 3.8



Gambar 3. 8 Alur proses *training* VGG19

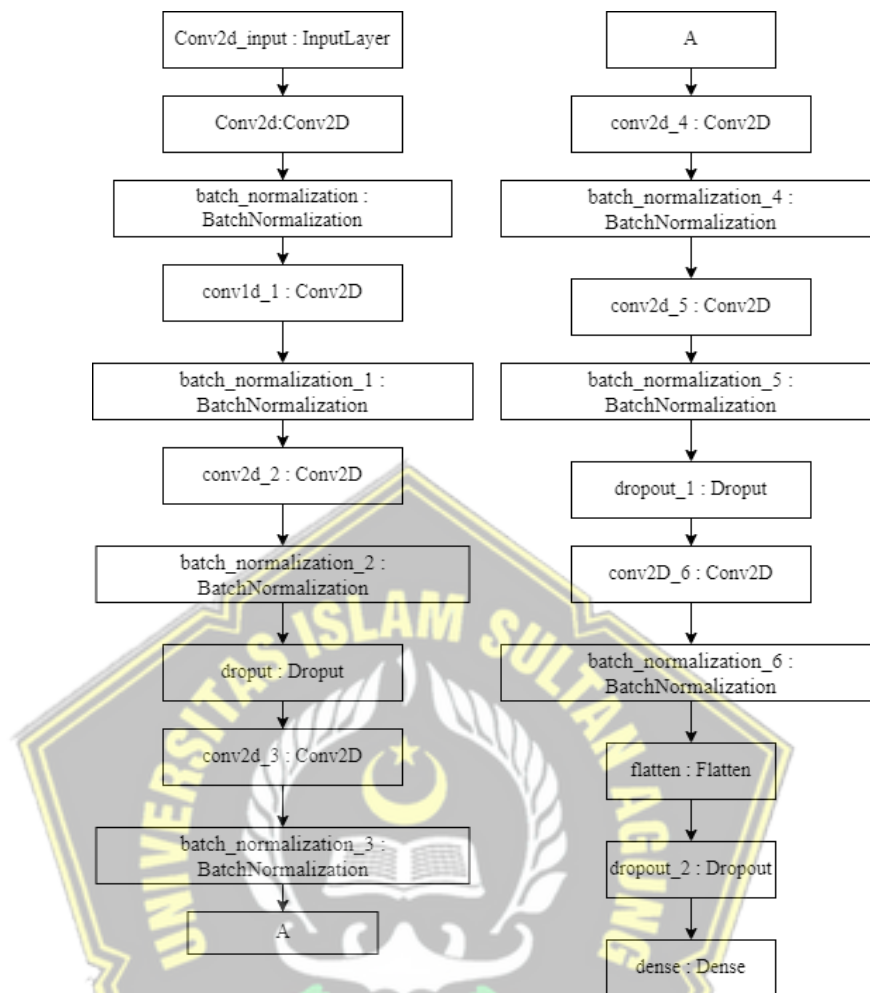
Langkah ini akan menunjukkan dataset yang digunakan dan apakah kode yang digunakan sebelumnya sudah pernah memberikan hasil. Pada proses ini, citra data *training* akan dianalisa dan dibuatkan label normal dan katarak agar sistem dapat mengidentifikasi label masing-masing citra.

➡ Jumlah sampel dalam dataset training: 480
 Jumlah batch dalam dataset training: 15

Gambar 3. 9 Hasil dataset *training*

3.5.5. Membuat Arsitektur VGG19

Dengan menggunakan pendekatan ini, penulis akan membuat arsitektur berdasarkan model VGG19 untuk mengklasifikasikan gambar menggunakan keras dan sekuensial. Saat membuat model, penulis harus menggunakan *Conv2D*, *MaxPooling2D*, *Flatten*, dan *Dense Layer*. Kedua, fungsi aktif yang digunakan dalam kode ini adalah Sigmoid dan ReLu untuk mengklasifikasikan tiga kelas.



Gambar 3. 10 Convulation Layer

Convulation Layer melakukan operasi konvolusi pada output dari lapisan sebelumnya. Lapisan tersebut adalah proses utama yang menerjemahkan CNN. Bobot pada lapisan tersebut mengkhhususkan kernel konvolusi yang diterapkan sehingga dapat dilonggarkan berdasarkan masukan pada CNN. Tujuan melakukan analisis konvolusi pada data input adalah untuk merepresentasikan fitur-fitur data secara visual. Konvolusi memiliki kemampuan untuk menghasilkan transformasi linier dari data masukan sesuai dengan informasi spasial pada data tersebut. Robot layer-by-layer spesifikasi kernel konvolusi yang dipakai, juga dapat dilatih berdasarkan input pada CNN.

```
[19] from tensorflow.keras.applications import VGG19
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Flatten, Dense

# Load pre-trained VGG19 model
vgg = VGG19(weights='imagenet', include_top=False, input_shape=(224, 224, 3))

# Mengatur semua layer VGG19 menjadi tidak dapat dilatih
for layer in vgg.layers:
    layer.trainable = False

# Membuat model Sequential
model = Sequential()
model.add(vgg)
model.add(Flatten())
model.add(Dense(1, activation="sigmoid"))

model.summary()
```

Gambar 3. 11 Arsitektur CNN menggunakan VGG19

3.5.6. Compile Model

Pada bagian ini, penulis akan membuat model perbandingan CNN menggunakan pustaka Keras untuk meningkatkan kecepatan dan akurasi pembelajaran. Alat pembelajaran terakhir yang digunakan adalah Adam, meskipun akurasi adalah kunci dari ketepatan matematisnya. Proses ini melibatkan pembangunan model dengan menyesuaikan berbagai parameter seperti optimizer, loss function, dan metrix evaluasi.

```
# Compile the model
custom_model.compile(optimizer='adam', loss='binary_crossentropy', metrics=['accuracy'])
```

Gambar 3. 12 Code compile model

3.5.7. Training Model CNN

Training dilakukan untuk memberikan pengetahuan tentang arsitektur *Convolutional Neural Network* (CNN) yang baru-baru ini menggunakan arsitektur VGG19.

```
# Melatih model use early stopping
history_1 = custom_model.fit(
    train_data,
    validation_data= val_data,
    epochs=5,
)
```

Gambar 3. 13 Training model cnn

Pada gambar 3.13 adalah kode yang digunakan untuk melakukan data *training*, kemudian epoch yang digunakan dalam penelitian ini adalah epoch 5, 15 dan 30. Hasil *training* terbaik berhasil didapatkan pada epoch 30 dengan

akurasi yang diperoleh adalah sebesar 100%. Pada gambar 4.14 merupakan hasil dari *training* model dengan epoch 30.

```
Epoch 1/5
15/15 [#####] - 272s 18s/step - loss: 0.2157 - accuracy: 0.0167 - val_loss: 0.1038
Epoch 2/5
15/15 [#####] - 271s 18s/step - loss: 0.0125 - accuracy: 1.0000 - val_loss: 0.0406
Epoch 3/5
15/15 [#####] - 268s 18s/step - loss: 0.0022 - accuracy: 1.0000 - val_loss: 0.0259
Epoch 4/5
15/15 [#####] - 263s 18s/step - loss: 0.0021 - accuracy: 1.0000 - val_loss: 0.0155
Epoch 5/5
15/15 [#####] - 260s 18s/step - loss: 0.0021 - accuracy: 1.0000 - val_loss: 0.0123
```

Gambar 3. 14 Hasil *Training* epoch 30

Training model CNN melibatkan serangkaian langkah yang komprehensif untuk memastikan model dapat mempelajari fitur-fitur yang relevan dari data gambar dan menghasilkan prediksi yang akurat.

3.5.8. Evaluasi

Evaluasi model menggunakan kode `custom_model.evaluate` (`test_dataset`) untuk mengetahui bagaimana prediksi model yang telah dibuat sebelumnya dengan menggunakan data baru. Gambar 3.15. Hasil analisis model VGG19 kedua dalam mendapatkan data baru. Akurasi yang dapat diperoleh dari data baru ini hampir mencapai 100%. Gambar 3.15 merupakan kode evaluasi model..

```
loss, accuracy = custom_model.evaluate(test_data)

print(f'Loss: {loss:.4f}')
print(f'Accuracy: {accuracy*100:.2f}%')
```

Gambar 3. 15 Code evaluasi model

3.5.9. Deployment

Deployment ini digunakan menggunakan *Streamlit*, yang memberikan antarmuka pengguna yang transparan yang memudahkan pengguna untuk memanipulasi gambar dan melihat hasilnya sebagai normal atau katarak. *Streamlit* adalah *framework open-source* untuk membuat aplikasi data interaktif menggunakan bahasa pemrograman Python. *Streamlit* memungkinkan pengguna untuk membuat aplikasi data interaktif dengan cepat dan mudah tanpa harus berurusan dengan teknologi web yang rumit. Selain itu, *Streamlit* menyediakan fitur untuk memvisualisasikan

persimpangan lapisan aktif dalam model *Convolutional Neural Network* (CNN), seperti model VGG19 yang sebelumnya telah dilatih pada dataset ImageNet.

Dalam proses *deployment* ini, langkah pertama adalah membuat pustaka yang diperlukan untuk sistem klasifikasi flask. Library yang dibutuhkan adalah *Flask*, *OS*, *TensorFlow*, *Keras*, dan *NumPy*. Setelah semua library ini dibuat, model CNN yang telah dibuat dan dievaluasi sebelumnya diperlukan agar aplikasi dapat memprediksi grafik secara akurat. Setelah model VGG-19 disesuaikan dan disempurnakan untuk kategorisasi berdasarkan kategorisasi, dapat menilai kinerjanya dengan menggunakan seperangkat validasi yang sesuai. Jika kinerjanya tidak sesuai, penulis dapat menyesuaikan *hiperparameter* agar model dapat bekerja lebih baik saat menentukan akurasi klasifikasi data. Setelah itu, router dipindahkan ke halaman muka situs web.

3.6 Implementasi

Pada langkah implementasi sistem, berbagai alat dan teknik yang akan digunakan untuk meningkatkan sistem akan dianalisis. Data mengenai instrumen dan alat yang akan digunakan dapat dilihat pada tabel 3.2.

Tabel 3. 2 Analisis *Tools* Sistem

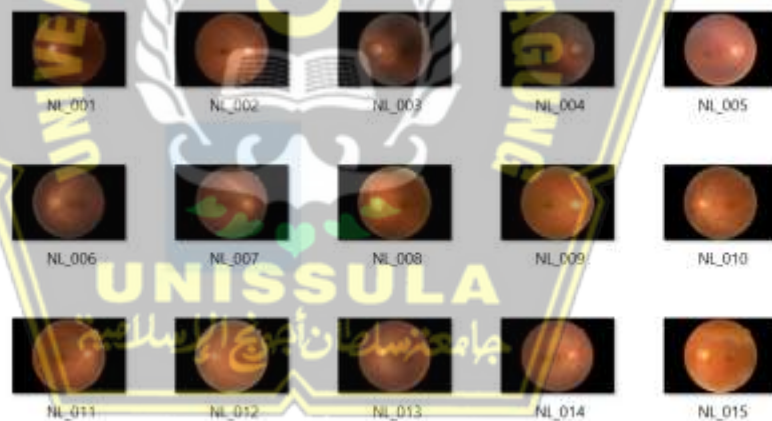
<i>Tools</i>	<i>Version</i>
<i>Python</i>	3.11.4
<i>TensorFlow</i>	2.13.0
<i>Keras</i>	2.13.1
<i>Streamlit</i>	1.25.0
<i>Jupyter Notebook</i>	7.0.0
<i>Visual Studio Code</i>	1.80.1

3.7 Pengujian

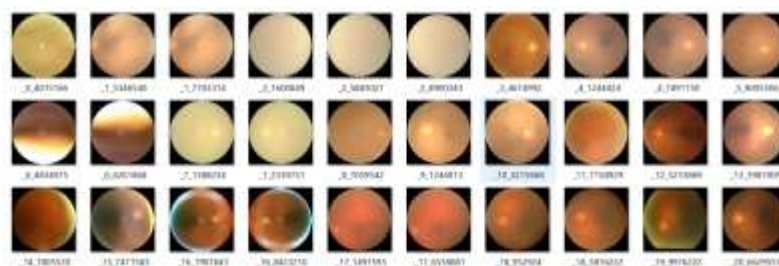
Pada tahap ini, data yang digunakan dalam penelitian ini adalah grafik fundus pada katarak dan grafik fundus pada normal yang diunggah ke Kaggle. Setelah proses pengumpulan data, partisipan manusia menyelesaikan folder *training* dan *test*. Hal ini dilakukan untuk memudahkan proses identifikasi sistem yang akan digunakan untuk fase prediksi dan pelatihan.

a) Data Collection

Tahap pertama yaitu pengumpulan data berupa citra mata normal dan katarak yang terdiri dari 300 normal dan 300 katarak. Dalam penelitian ini, platform kolaborasi Google digunakan untuk membuat aplikasi berbasis situs web yang mengklasifikasikan penyakit Katarak dan Normal. Sebaliknya, untuk pengarsipan data, menggunakan Google Drive. Saat membuat aplikasi, Google Drive dan Collab harus terhubung.



Gambar 3. 16 Data Citra Fundus Mata Normal



Gambar 3. 17 Data Citra Fundus Mata Katarak

Data citra fundus matra katarak ditunjukkan pada Gambar 3.16 sebagai contoh dan formatnya, sedangkan data citra fundus mata normal ditunjukkan pada Gambar 3.17.

b) Pemodelan CNN

Tahapan pemodelan *Convolutional Neural Network* (CNN) meliputi beberapa langkah penting. Pertama, CNN melakukan ekstraksi fitur dari data visual, seperti gambar dan video, dengan menggunakan lapisan konvolusional yang melintasi dimensi spasial dengan menerapkan satu set filter yang dapat dipelajari ke data masukan. Proses ini memungkinkan penggunaan jaringan untuk memvisualisasikan fitur lokal, seperti teks, gambar, dan sudut, dari gambar masukan. Selanjutnya, CNN melakukan proses pooling untuk melakukan operasi *downsampling* spasial pada input, metode yang biasa digunakan adalah dengan menggunakan max pooling atau mengekstrak nilai tertinggi dari wilayah tersebut. Setelah itu, hasil *pooling* dilakukan proses *flattening* untuk mengubah matriks hasil *pooling* menjadi vector. Kemudian, CNN melakukan proses *full connection layer*, yaitu menghubungkan setiap neuron di lapisan sebelumnya ke setiap neuron di lapisan berikutnya, hingga menghasilkan output akhir. Tahapan terakhir adalah proses evaluasi model untuk mengevaluasi kinerja model yang telah dibangun.

c) *Training*

Training dilakukan untuk memberikan pengetahuan tentang arsitektur *visual geometry group* (VGG19). Dimana nantinya akan dilihat nilai akurasi dari 19 layer untuk mendapatkan konfigurasi terbaik. Setelah arsitektur dibangun, saya akan memulai proses *training*. Selain itu, pengujian juga melibatkan perubahan beberapa parameter untuk menghasilkan model yang digunakan untuk mendapatkan tingkat akurasi terbaik dalam mengklasifikasi penyakit mata katarak. Tahapan pengujian juga melibatkan proses transfer learning dan fine tuning terhadap model, serta evaluasi model.

d) *Testing*

Proses *testing* adalah proses klasifikasi data menggunakan bias dan bobot berdasarkan hasil pelatihan. Dengan demikian, hasil akhir dari proses ini adalah rata-rata dari klasifikasi yang dilakukan, data yang diklasifikasikan dengan benar, jumlah kelas yang diklasifikasikan dengan benar, dan struktur jaringan yang dibentuk oleh proses *feedforward*. Saat ini, label sepenuhnya terkait dengan lapisan keluaran. Untuk menilai akurasi, peneliti akan menggunakan matriks kebingungan klasifikasi yang memberikan informasi tentang *Accuracy*, *Precision*, *Recall*, dan *F1-Score* untuk memastikan bahwa model berkinerja baik. Di bawah ini adalah penjelasan dari setiap fungsi yang akan digunakan dalam proses perhitungan:

a) *Confusion Matrix*

Matriks Membingungkan (*Confusion Matrix*) adalah tabel yang digunakan untuk mengilustrasikan proses kerja model klasifikasi. *Confusion matrix* membandingkan label saat ini dari data dengan label yang diprediksi oleh model. *Confusion matrix* memiliki baris dan kolom yang menunjukkan kelas aktual dan kelas yang diprediksi. Setiap baris dalam matriks merepresentasikan jumlah atau persentase contoh yang digabungkan ke dalam kombinasi tertentu berdasarkan prediksi. *Confusion matrix* untuk masalah klasifikasi binary adalah sebagai berikut:

Tabel 3. 3 Klasifikasi confusion matrik

	<i>Actual Positive</i>	<i>Actual Negative</i>
<i>Predicted Positive</i>	True Positive (TP)	False Positive (FP)
<i>Predicted Negative</i>	False Negative (FN)	True Negative (TN)

Confusion matrix adalah alat yang berguna dalam studi kasus arsitektur VGG19 untuk mengevaluasi kinerja model dalam mengklasifikasikan gambar ke tingkat yang sesuai. Dengan menggunakan confusion matrix, kami dapat mengukur berbagai metrik evaluasi, seperti accuracy, presisi, recall dan F1-Score untuk setiap tingkat kelas. Hal ini dapat membantu kami memahami jenis

kesalahan yang dilakukan model dan meningkatkan kinerja model tersebut.

b) *Accuraccy*

Akurasi dalam *confusion matrix* berkorelasi dengan nilai kebenaran dari setiap prediksi yang benar. Untuk menghitung akurasi dengan menggunakan rumus ini :

$$\text{Accuraccy} = \frac{TP+TN}{\text{Total}} \quad (1)$$

Keterangan :

- TP adalah True Positive (prediksi positif yang benar),
- TN adalah True Negative (prediksi negatif yang benar), dan
- Total adalah total jumlah prediksi.

Dalam konteks arsitektur VGG19, akurasi *confusion matrix* akan memberikan informasi tentang beberapa model VGG19 yang baik dalam hal mengklasifikasikan gambar ke dalam kelas dengan benar. Salah satu metrik evaluasi yang penting untuk menilai kinerja model klasifikasi adalah akurasi. Dengan menggunakan akurasi, penulis dapat menentukan beberapa model yang baik untuk prediksi kelas-kelas yang didasarkan pada data yang disediakan.

c) *Precision*

Precision dalam *confusion matrix* mengacu pada persentase dari prediksi positif yang benar. Untuk menghitung *Precision* , penulis dapat menggunakan rumus.

$$\text{Precision} = \frac{TP}{TP+FP} \quad (2)$$

Precision confusion matrix akan memberikan gambaran tentang seberapa baik model VGG19 dalam memprediksi kelas positif dengan benar. *Precision* merupakan salah satu metrik evaluasi yang penting untuk mengevaluasi performa model klasifikasi. Dengan menggunakan *Precision* , penulis dapat mengetahui seberapa baik model dalam memprediksi kelas positif dari data yang diberikan.

d) *Recall*

Recall dalam *confusion matrix* mengacu pada persentase dari instance positif yang benar diprediksi oleh model. Untuk menghitung *recall*, penulis dapat menggunakan rumus:

$$\text{Recall} = \frac{TP}{TP+FN} \quad (3)$$

e) *F1-Score*

F1-Score dalam *confusion matrix* adalah sebuah metrik evaluasi yang menggabungkan *Precision* dan *recall*. *F1-Score* memberikan gambaran tentang seberapa baik model dalam memprediksi kelas positif dengan benar dan seberapa baik model dalam menghindari prediksi yang salah. *F1-Score* dapat dihitung menggunakan rumus

$$\text{Recall} = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \times 100\% \quad (4)$$

3.8 Evaluasi Sistem

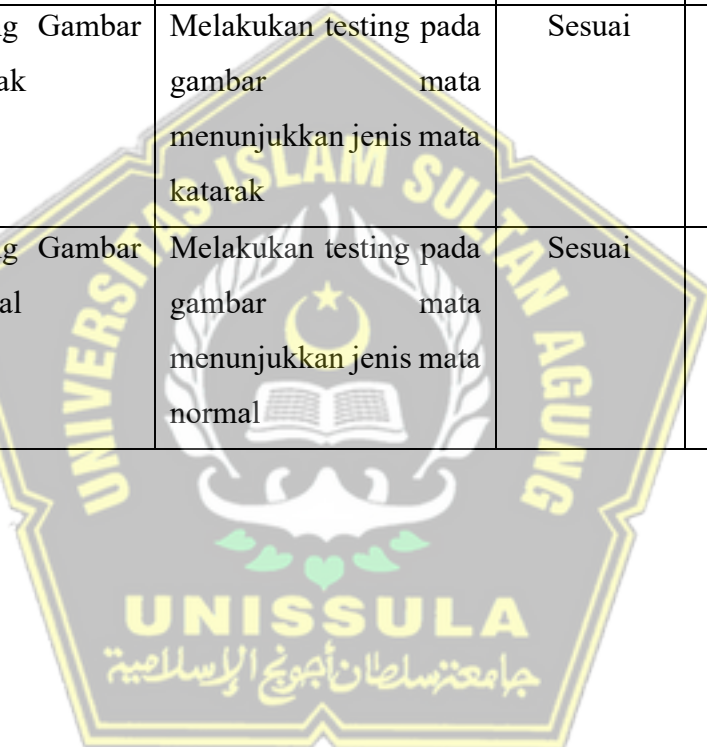
Pada tahap ini, dilakukan evaluasi sistem menggunakan metode black box testing. Metode *black box testing* adalah teknik pengujian perangkat lunak yang bertujuan untuk menguji fungsionalitas sistem tanpa perlu mengetahui rincian struktur atau cara kerja internalnya. Beberapa jenis *black box testing* yang umum digunakan termasuk *function testing*, *non-function testing*, dan *regression testing*.

Dalam penelitian ini, dilakukan *function testing* untuk menguji setiap fungsi yang ada pada sistem. Tujuan dari pengujian *function testing* adalah untuk memastikan bahwa setiap fungsi berjalan dengan benar dan sesuai dengan spesifikasi yang telah ditetapkan.

Hasil pengujian *function testing* pada sistem ini kemudian ditampilkan dalam tabel 3.4. Tabel tersebut berisi hasil dari pengujian berbagai fungsi pada sistem dan menunjukkan apakah fungsi-fungsi tersebut berhasil berjalan dengan benar atau terdapat kesalahan yang perlu diperbaiki.

Tabel 3. 4 *Black Box* Testing

Skenario Pengujian	Kasus Pengujian	Hasil Pengujian	Kesimpulan
Membuka <i>website</i>	Menampilkan Halaman Utama	Sesuai	Normal
Upload gambar dari <i>storage</i>	Melakukan upload gambar mata dari <i>storage</i>	Sesuai	Normal
Testing Gambar Katarak	Melakukan testing pada gambar mata menunjukkan jenis mata katarak	Sesuai	Normal
Testing Gambar Normal	Melakukan testing pada gambar mata menunjukkan jenis mata normal	Sesuai	Normal



BAB VI

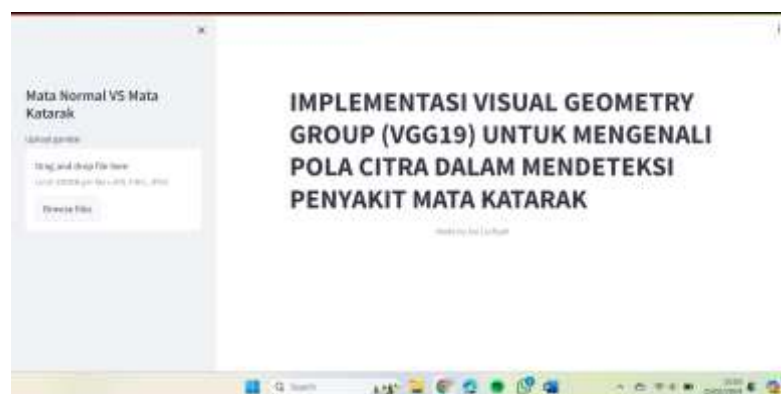
HASIL DAN ANALISIS PENELITIAN

4.1 *Deployment*

Pada *deployment* ini penulis menggunakan *streamlit*, *streamlit* digunakan selama penerapan karena memudahkan pengembang dan ilmuwan data untuk membuat aplikasi web interaktif dengan jumlah bisnis yang sedikit. Selain itu, *streamlit* memungkinkan pengguna untuk menambahkan elemen interaktif seperti tombol, slider, dan input teks untuk memungkinkan interaksi pengguna dengan aplikasi. Selain itu, Streamlit dapat diintegrasikan dengan banyak pustaka dan kerangka kerja ilmu data populer, termasuk Pandas, Matplotlib, Plotly, dan Scikit-learn. Selain itu, Streamlit menawarkan metode yang mudah digunakan untuk menyebarkan aplikasi ke berbagai platform cloud. Dengan menggunakan Streamlit, para peneliti dapat membuat aplikasi yang lebih mudah digunakan dan menarik, memfasilitasi eksplorasi data, dan menyajikan hasil analisis mereka dengan cara yang lebih mudah dimengerti dan lebih intuitif.

4.4.1 Halaman Utama

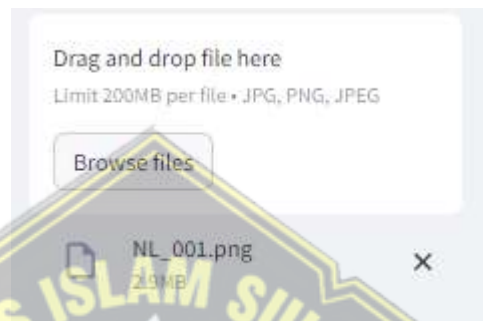
Halaman utama situs web adalah ringkasan yang dapat memberikan informasi komprehensif tentang konten dan fitur yang tersedia. Selain itu, bilah navigasi utama dapat menarik perhatian pengguna dan memfasilitasi navigasi ke area lain.



Gambar 4. 1 Tampilan Halaman Utama

Pada gambar merupakan tampilan utama dari website. Pada gambar di atas, terdapat sebuah tombol atau bagian sebelah kiri yang digunakan untuk menyorot gambar yang akan diklasifikasikan.

Setelah pengguna memilih gambar mata dengan menggunakan tombol "*browse files*", teks akan secara otomatis berubah menjadi seperti gambar 4.2 di bawah ini.

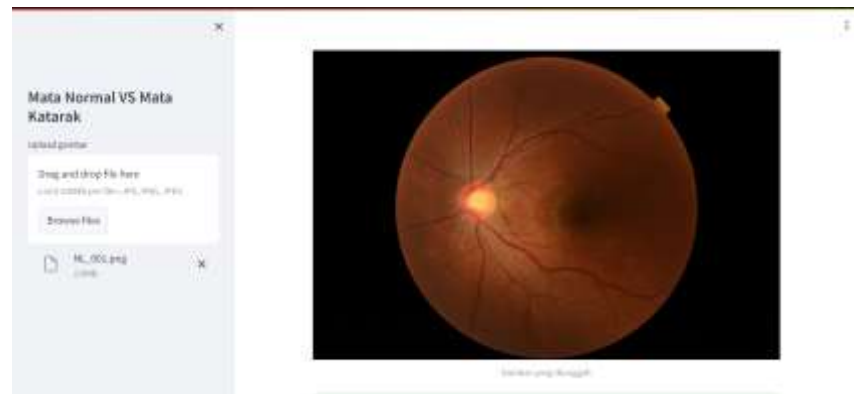


Gambar 4. 2 Halaman Upload Gambar

Padagambar 4.2 bagian bawah "*Browse Files*" untuk upload gambar. Terdapat file yang telah diupload oleh user, jadi user bisa mengetahui file yang telah diupload nya.

4.4.2 Halaman Klasifikasi

Ketika pengguna memilih tombol "Upload" setelah berhasil menavigasi ke gambar, proses klasifikasi dimulai. Selanjutnya, sistem akan menjalankan proses klasifikasi untuk mengidentifikasi jenis rambut wajah yang ada pada gambar yang ditampilkan. Setelah proses klasifikasi selesai, tabel akan menampilkan hasil klasifikasi yang mencakup informasi mengenai jenis wajah yang dapat diidentifikasi beserta ambang batas probabilitasnya. Informasi ini dapat ditunjukkan pada Gambar 4.3



Gambar 4. 3 Hasil Upload Gambar



Gambar 4. 4 Halaman hasil Prediksi

Tampilan antarmuka sistem mengacu pada representasi visual dari sebuah aplikasi atau sistem yang memungkinkan pengguna untuk berinteraksi dengan sistem yang telah dikembangkan dengan mudah. Sistem yang telah dikembangkan dengan mudah. Tampilan awal atau halaman awal dari sistem yang baru dirancang diwakili oleh Gambar 4.3 dan Gambar 4.4 merupakan tampilan halaman hasil pada sistem. Tampilan halaman awal menunjukkan hasil prediksi gambar citra mata yang dimasukkan pada sistem. Dari hasil prediksi tersebut terdapat keterangan klasifikasi mata normal atau katarak

4.2 Hasil Model Layer

Penelitian ini menggunakan CNN VGG19 sebagai arsitektur untuk mengklasifikasikan kesalahan kategorisasi. Arsitektur dalam hal ini telah terbukti mampu menganalisis objek pada layar secara akurat dengan ketepatan yang tinggi karena memiliki banyak lapisan kompleksitas yang dapat digunakan untuk mengekstrak detail yang rumit dari gambar. Selain itu, arsitektur VGG19 dapat diterapkan pada jenis klasifikasi gambar lainnya, termasuk kategorisasi berdasarkan kata kunci. Selama pelatihan, arsitektur VGG19 dapat disesuaikan dengan dataset citra katarak yang digunakan untuk meningkatkan akurasi klasifikasi. Model VGG19 yang digunakan dalam penelitian ini telah disesuaikan dan disempurnakan untuk mengklasifikasikan teks dengan memanfaatkan dataset citra teks yang besar dan menghasilkan label klasifikasi yang akurat. Selain itu, pembelajaran transfer digunakan untuk memaksimalkan waktu belajar siswa dan meminimalkan percepatan pembelajaran. Setelah model VGG19 disesuaikan, kinerjanya dinilai dengan menggunakan serangkaian validasi yang diperluas. Jika performa kerja tidak sesuai standar, tuning *hyperparameter* dilakukan agar model dapat bekerja lebih baik saat menentukan klasifikasi data.

Arsitektur VGG19 yang diterapkan pada penelitian ini sebagai berikut

:

```
Model: "sequential_3"
```

Layer (type)	Output Shape	Param #
vgg19 (Functional)	(None, 7, 7, 512)	20024384
flatten_2 (Flatten)	(None, 25088)	0
dense_5 (Dense)	(None, 1)	25089

```

=====
Total params: 20049473 (76.48 MB)
Trainable params: 25089 (98.00 KB)
Non-trainable params: 20024384 (76.39 MB)
=====

```

Gambar 4. 5 Model yang diterapkan

Pada gambar 4.5 merupakan lapisan layer VGG19 yang diterapkan pada model CNN.

4.3 Hasil *Visualisasi* dan *Augmentasi* VGG19

Visualisasi lapisan aktif dalam model *Convolutional Neural Network* (CNN) dapat membantu penulis memahami bagaimana model CNN mengekstrak fitur dari gambar yang diberikan dan bagaimana memanipulasi fitur-fitur tersebut. Dalam model CNN, setiap lapisan memiliki beberapa filter yang dapat digunakan untuk menyoroti fitur tertentu dari gambar. Visualisasi lapisan aktif dapat membantu penulis memahami fitur yang dihadirkan oleh setiap filter pada setiap lapisan.

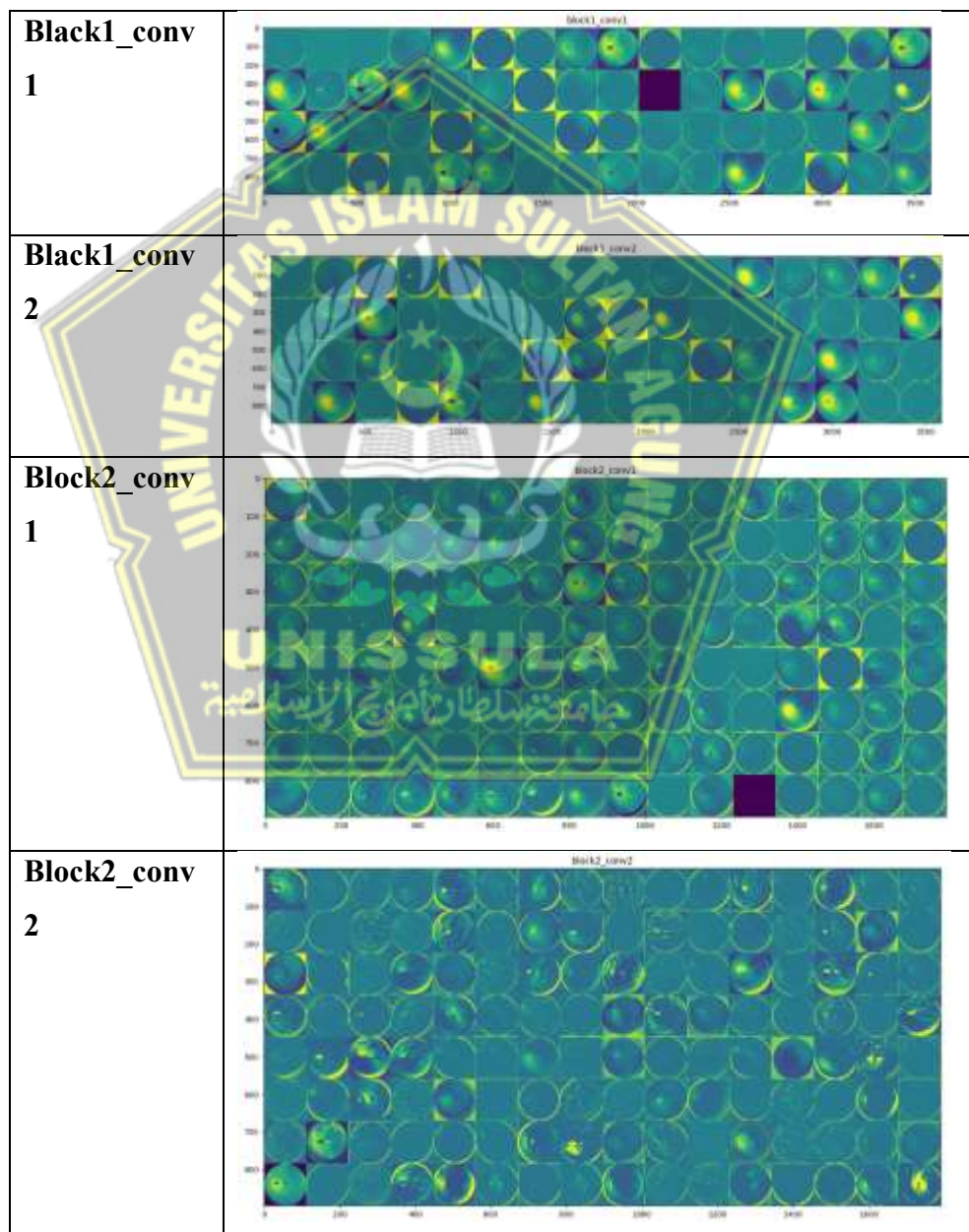
Untuk memvisualisasikan lapisan aktif pada model CNN, yaitu model VGG19 yang sebelumnya telah dilatih pada dataset ImageNet, penulis dapat menggunakan teknik yang disebut Grad-CAM (*Gradient-weighted Class Activation Mapping*). Teknik ini memungkinkan penulis untuk memvisualisasikan bagian gambar yang paling signifikan mempengaruhi hasil klasifikasi model CNN.

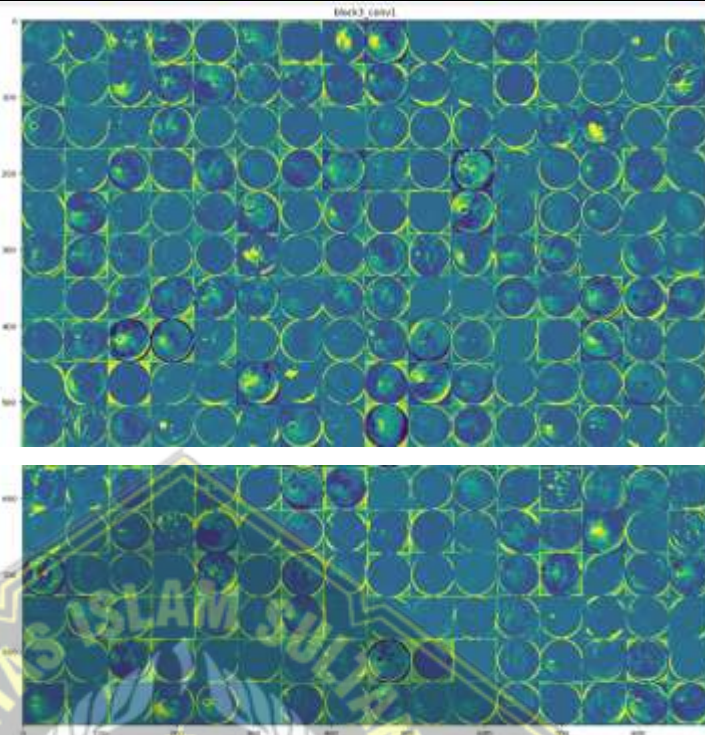
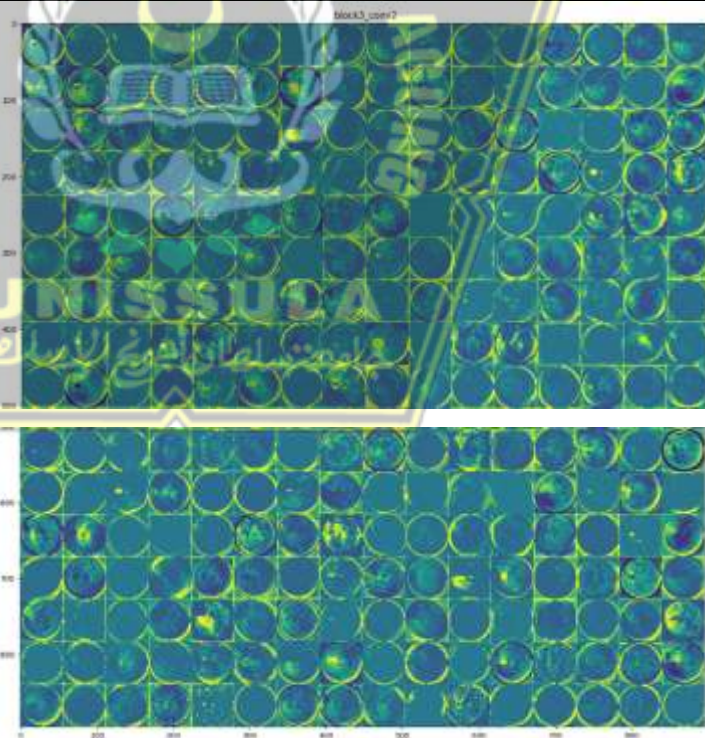
Untuk melihat aktivitas lapisan pada model VGG19 menggunakan Grad-CAM.

1. Model VGG19 yang telah dilatih sebelumnya dari dataset ImageNet.
2. Memasukkan gambar yang diinginkan untuk memvisualisasikan lapisan yang aktif.
3. Melakukan pra-pemrosesan pada gambar, seperti mengubah ukurannya menjadi 224 x 224 piksel dan menerapkan normalisasi.
4. Melakukan prediksi menggunakan model VGG19 pada gambar.
5. Mendapatkan *output* dari lapisan yang harus divisualisasikan ketika lapisan tersebut diaktifkan.
6. Gradien dari keluaran lapisan tersebut terhadap keluaran kelas yang diprediksi.

7. Totalitas bobot rata-rata dari setiap filter dalam lapisan tersebut ditentukan oleh gradien yang diukur sebelumnya, dan
8. Memanfaatkan nilai bobot rata-rata dengan output lapisan yang disebutkan di atas untuk mendapatkan peta panas yang menunjukkan bagian gambar yang paling signifikan mempengaruhi hasil klasifikasi model CNN.

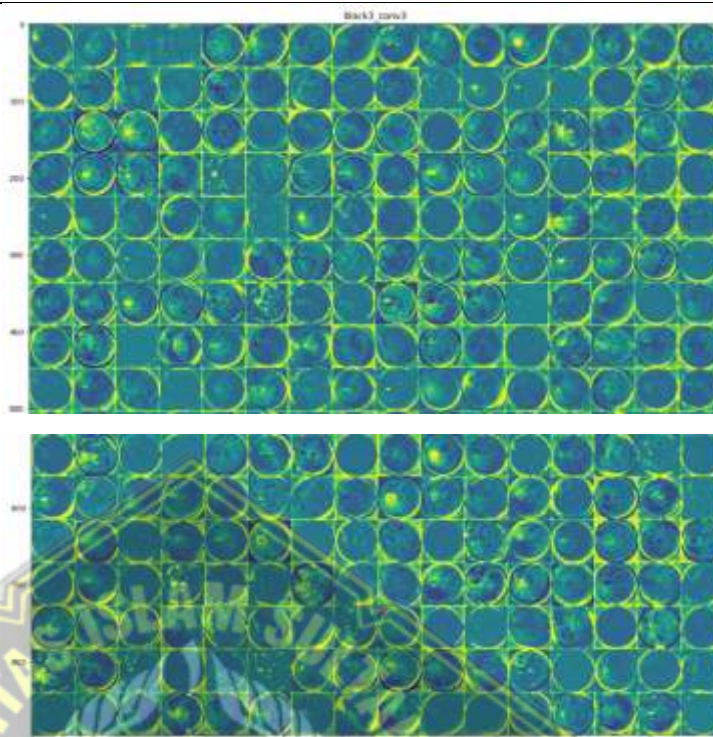
Tabel 4. 1 Visualisasi VGG19



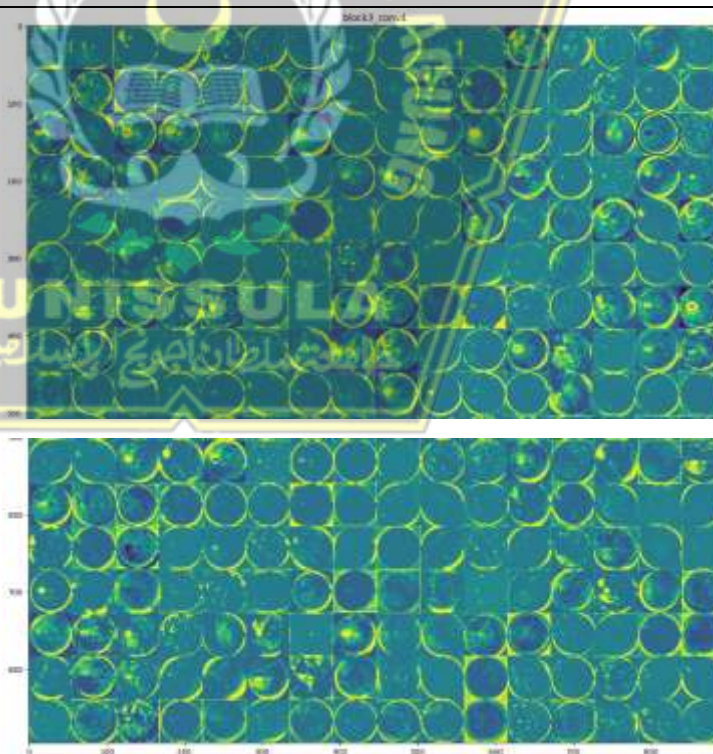
Block3_conv**1****Block3_conv****2**

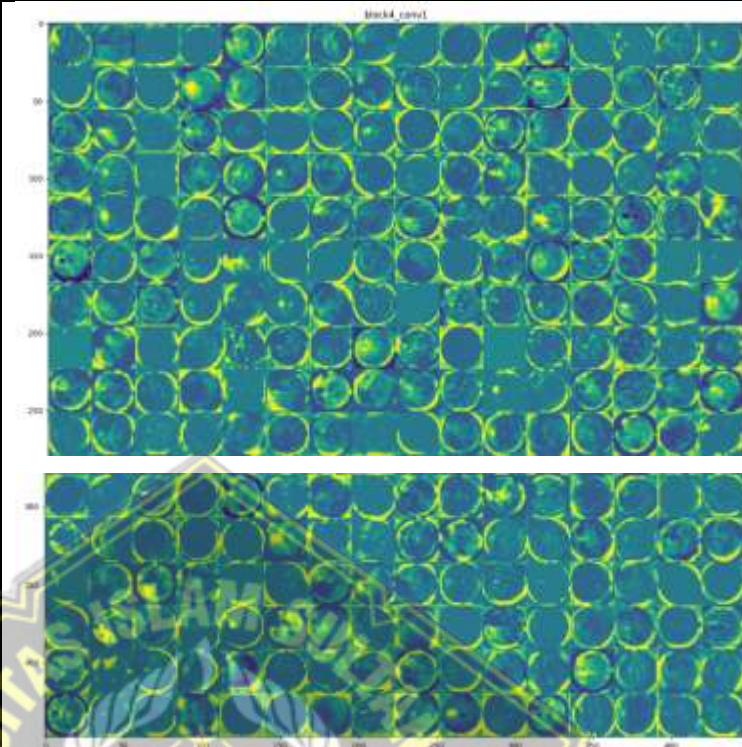
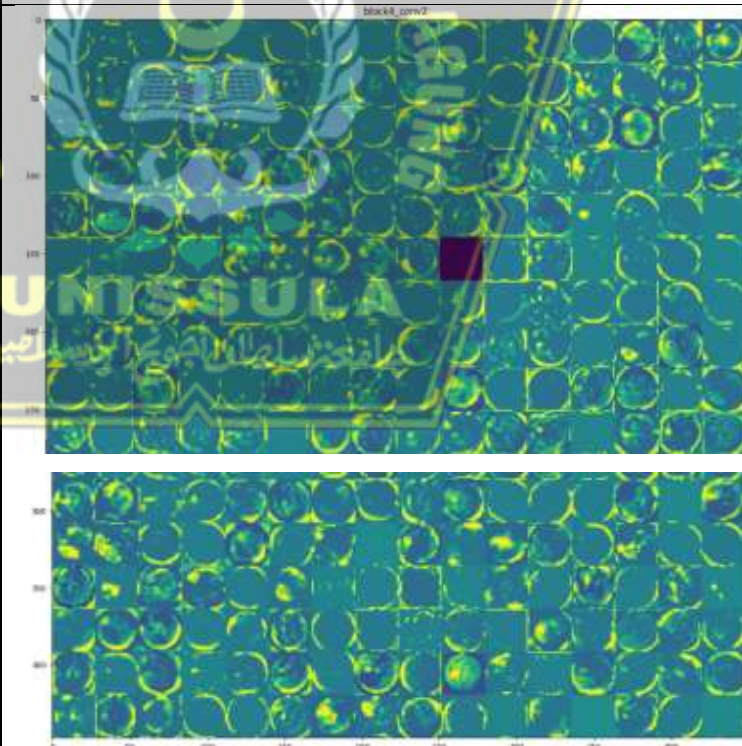
Block3_conv

3

**Block3_conv**

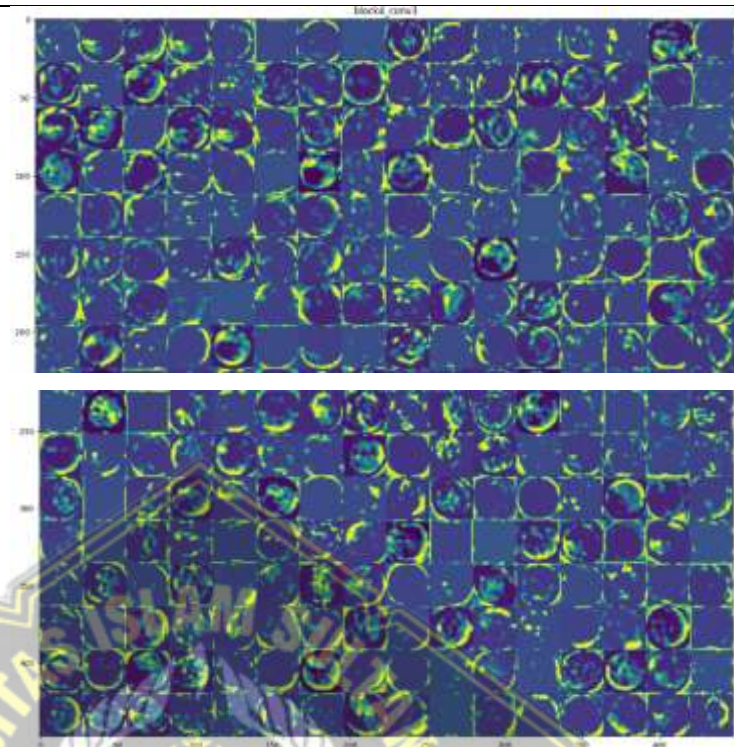
4



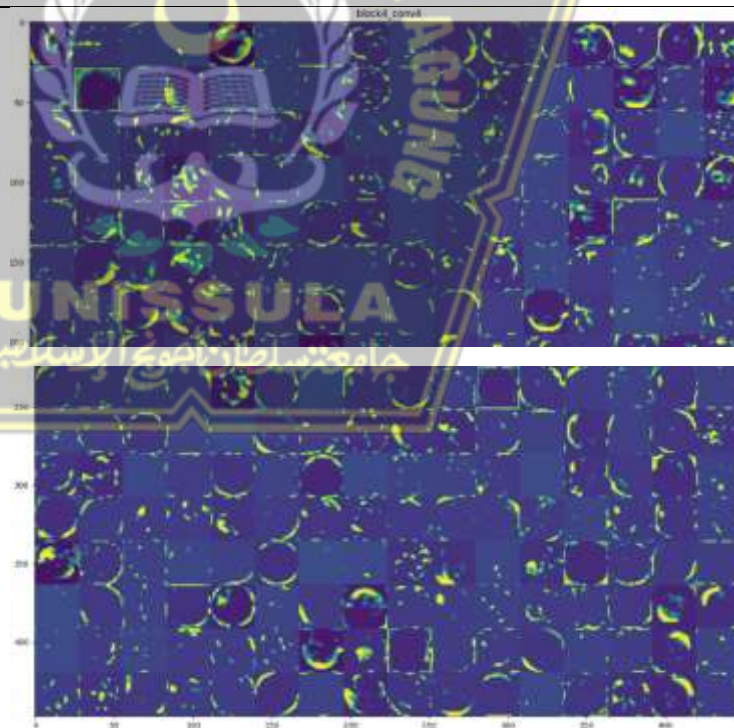
Block4_conv**1****Block4_conv****2**

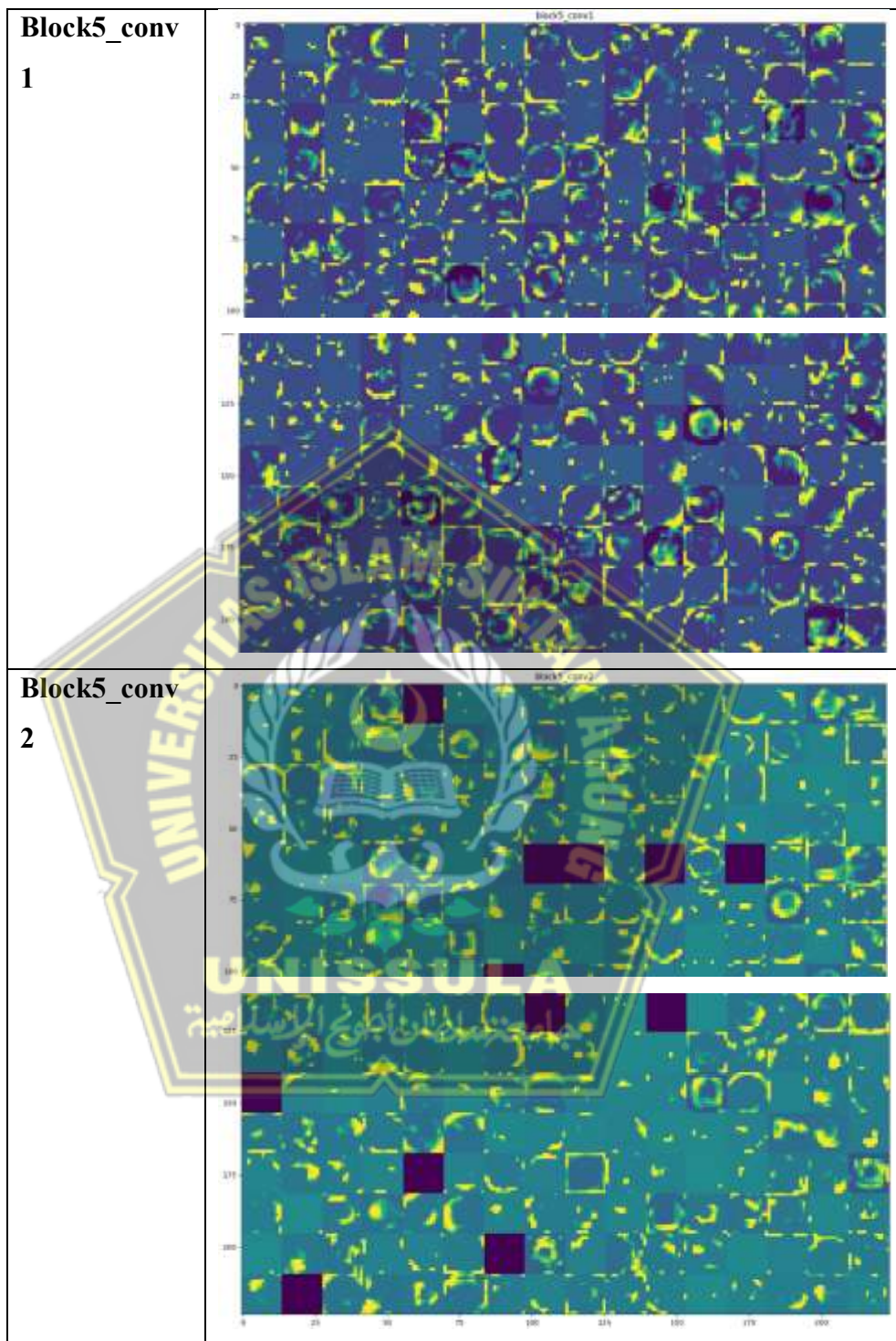
Block4_conv

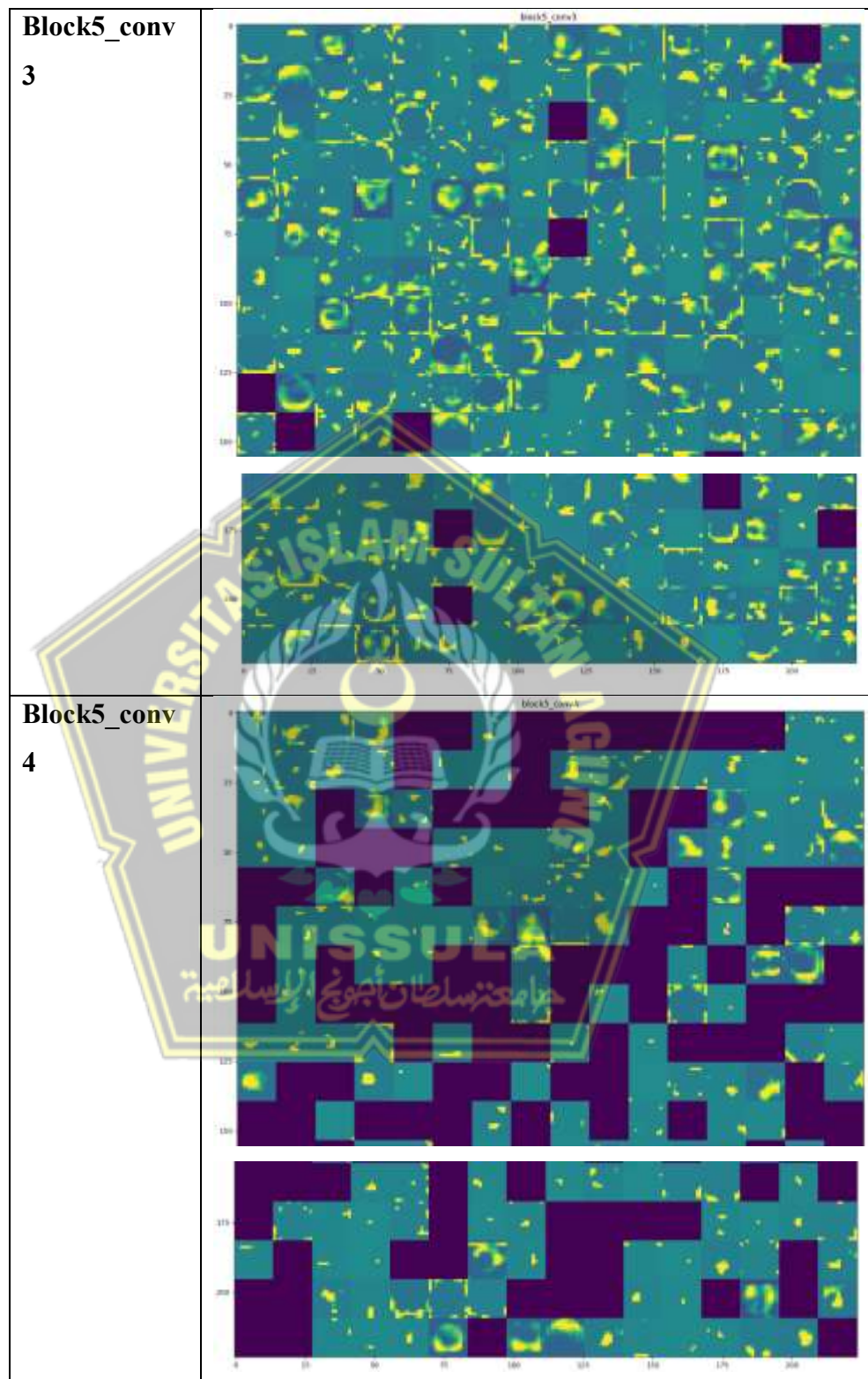
3

**Block4_conv**

4







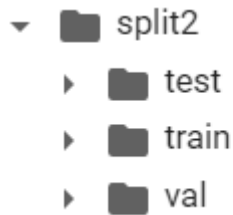
Dengan memvisualisasikan lapisan aktif pada model VGG19, penulis dapat memahami bagaimana model CNN mengekstrak gambar dan mengekstrak fitur tertentu dari gambar. Hal ini dapat membantu penulis

memahami kesimpulan yang diambil oleh model CNN dan meningkatkan cara model menginterpretasikan hasil klasifikasi.

4.4 Distribusi Data *Train, Val dan Test*

Distribusi data *train, test, dan valid* pada klasifikasi grafik batang katarak menggunakan VGG19 sangat penting dilakukan untuk memastikan bahwa model yang dikembangkan dapat berfungsi dengan baik dengan data yang belum diteliti. Pengumpulan data ini dilakukan dengan membagi dataset ke dalam tiga kategori: dataset valid, uji, dan latih. Umumnya, 70% dari parameter digunakan untuk *train* set, 15% untuk test set, dan 15% untuk validation set. Selain itu, penting untuk memastikan bahwa setiap data set memiliki distribusi kelas yang konsisten dan melakukan augmentasi data pada *train* set untuk mencegah *overfitting*. Distribusi data yang dirancang dengan baik akan memungkinkan model untuk belajar dari data berkualitas tinggi dan membandingkannya dengan data berkualitas tinggi untuk menghasilkan prediksi yang akurat dari data yang belum pernah ditinjau sebelumnya.

Pada tahap pra-pemrosesan, ukuran gambar disesuaikan menjadi 224x224 piksel, dengan gambar berbentuk seragam yang membuatnya lebih mudah dilihat pada komputer. Selain itu, bentuk citra yang bergerigi dapat lebih memudahkan proses pengenalan. Untuk mengakses data *train, test, dan valid* untuk klasifikasi file gambar katarak menggunakan VGG19, Anda harus terlebih dahulu menyiapkan dataset dengan mengatur file gambar sesuai dengan kriteria *Train, Test, dan Val*.



Gambar 4. 6 Distribusi *Train*, *Val* dan *Test*

4.5 *Training* dan *Validation*

Selama proses pelatihan dan validasi, dataset yang dibutuhkan sebelumnya (augmentasi) digunakan. Dataset ini terdiri dari 480 sampel untuk data pelatihan dan 240 sampel untuk setiap kelompok. Sebaliknya, untuk validasi, ada 60 sampel yang telah diverifikasi dengan masing-masing 30 sampel perkelas nya. Metode untuk mengimplementasikan distribusi data yang disebutkan adalah

```

Found 480 images belonging to 2 classes.
Found 60 images belonging to 2 classes.
Found 60 images belonging to 2 classes.
  
```

Gambar 4. 7 Pembagian dataset

Dalam proses klasifikasi citra mata menggunakan arsitektur neural berupa *Convolutional Neural Networks* (CNN) data *Train*, *Test* dan *Val* memiliki peran penting dalam proses pelatihan, *fine-tuning*, dan penilaian model CNN. *Training Dataset* digunakan untuk melatih model CNN. Data ini akan membantu model untuk mempelajari pola dan karakteristik yang berhubungan dengan katarak. Setiap kali model dipelajari, data pembelajaran akan digunakan untuk menyempurnakan berbagai parameter CNN. *Validation Dataset* digunakan untuk mengevaluasi model dan memastikan bahwa model tersebut dapat bekerja dengan baik dengan data yang belum pernah diperiksa sebelumnya. Data ini membantu menentukan seberapa besar model dapat digeneralisasi dan apakah model telah dipelajari dengan baik melalui observasi. *Testing Dataset* digunakan untuk menilai kinerja model setelah pelatihan. Data ini tidak digunakan dalam pengajaran dan membantu

menentukan model mana yang sesuai yang dapat digunakan untuk mengklasifikasikan data baru. Hasil evaluasi data uji akan menjadi tolok ukur untuk membandingkan produktivitas model dengan model lain atau model yang telah dioptimalkan sebelumnya.

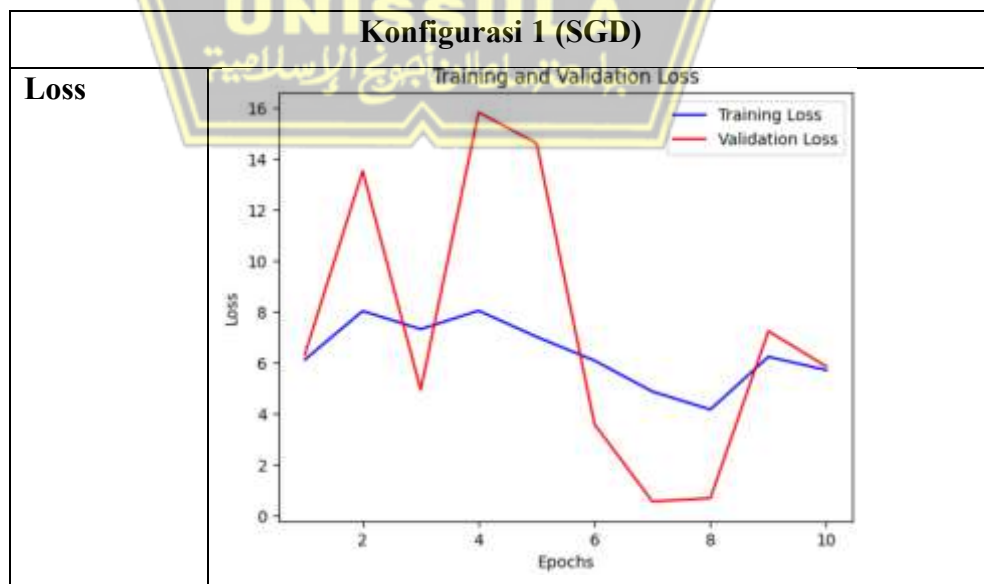
Setelah mendapatkan citra dengan ukuran dan distribusi data yang konsisten, selanjutnya ke langkah klasifikasi. Pada tahap ini, peneliti menggunakan parameter yang sama seperti yang ditunjukkan pada Tabel 4.2.

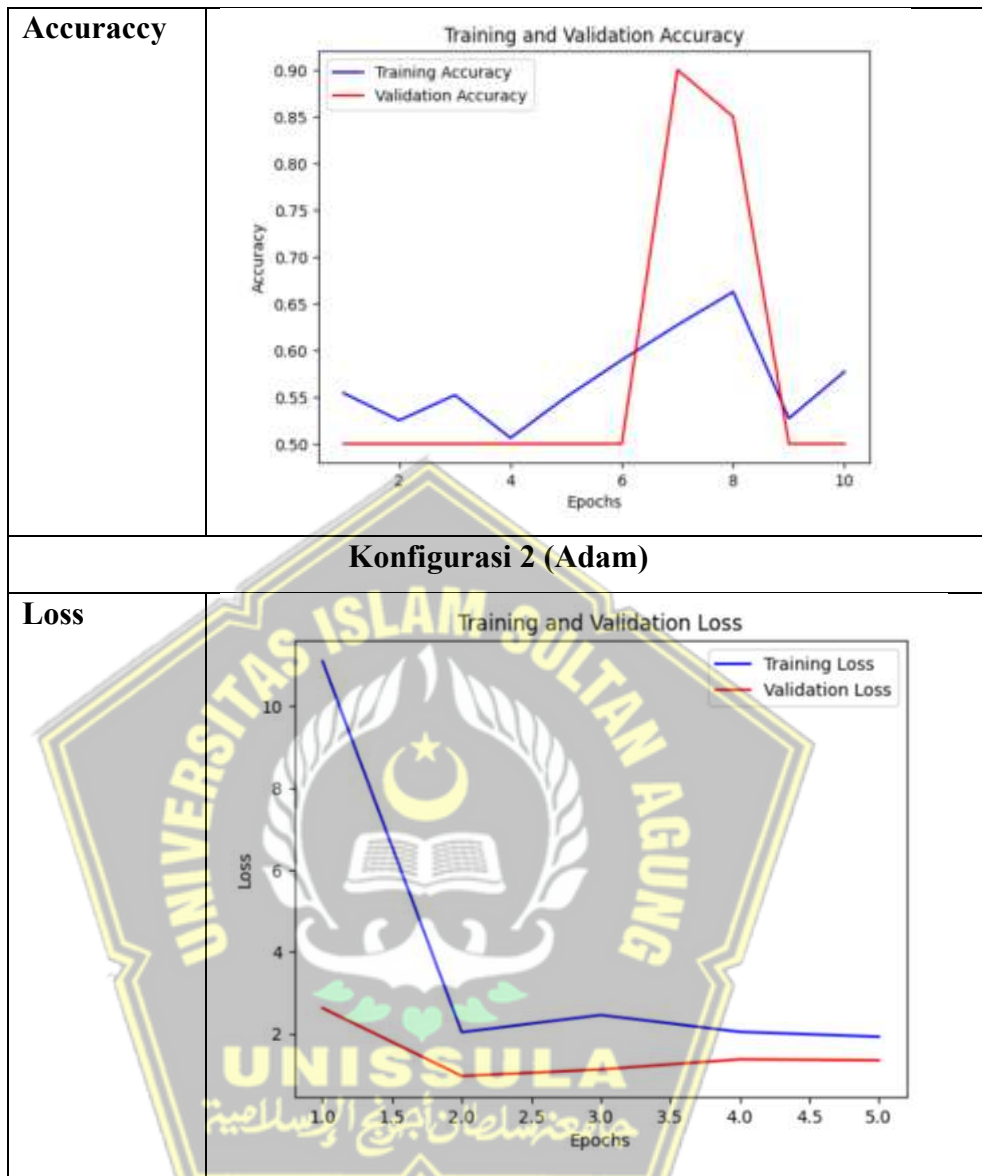
Tabel 4. 2 Parameter

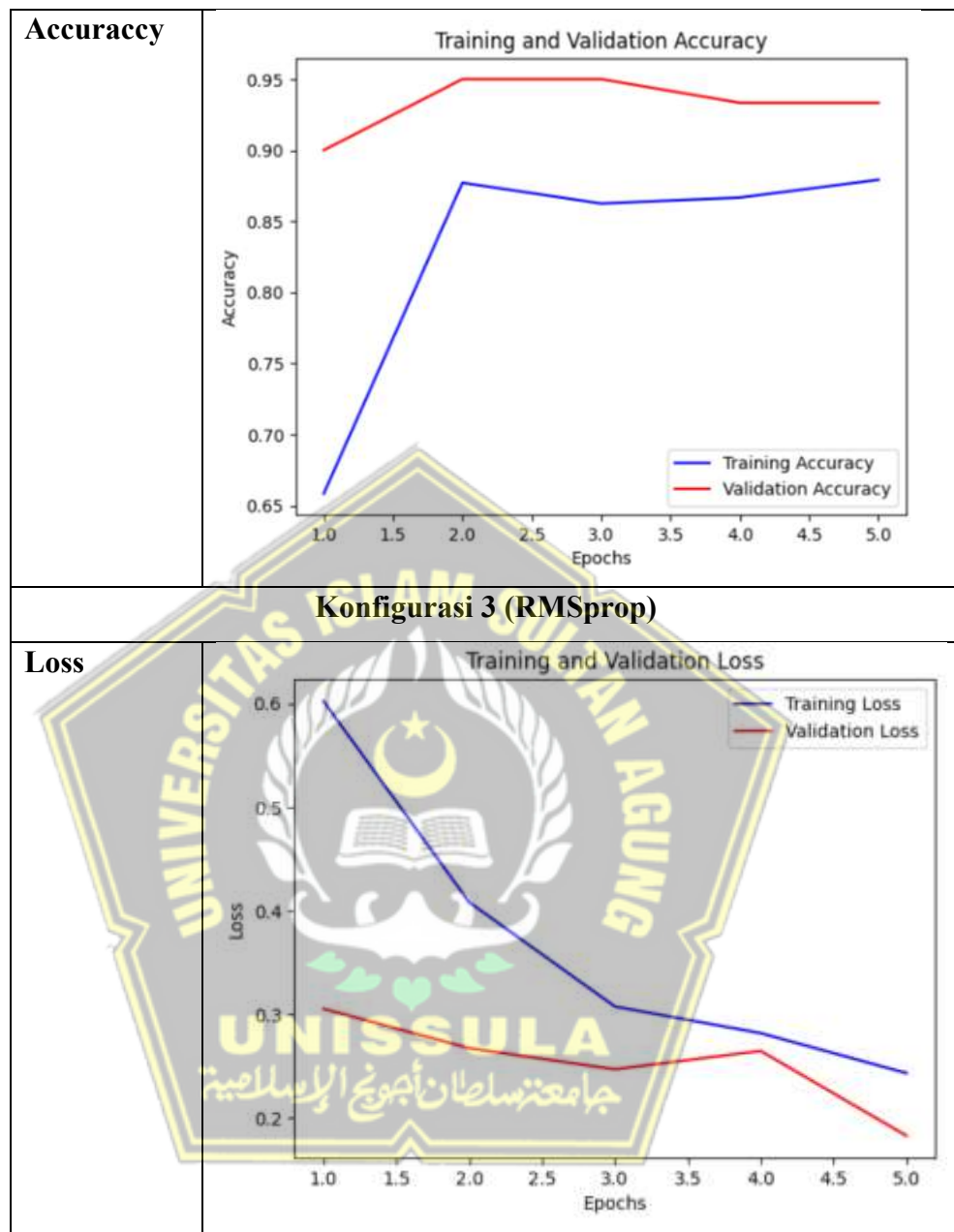
Nama Konfigurasi	Konfigurasi 1	Konfigurasi 2	Konfigurasi 3
<i>batch_size</i>	32		
<i>epoch</i>	30		
<i>learning-rate</i>	0.1	0.001	0.001
<i>optimizer</i>	SGD	ADAM	RMSprop
<i>loss</i>	binary_crossentropy		

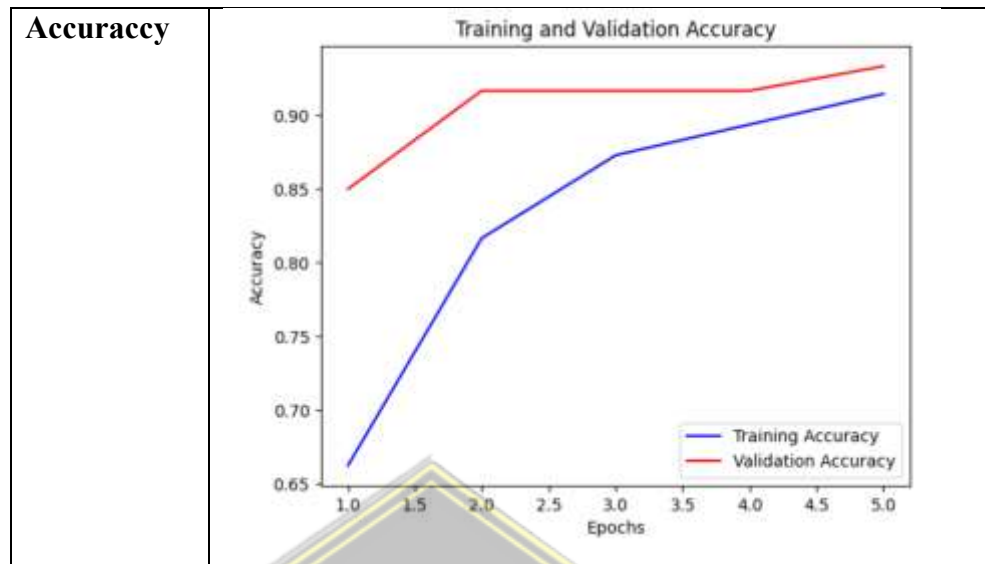
Berdasarkan parameter yang digunakan dan model yang diterapkan, hasilnya seperti yang ditunjukkan pada tabel 4.3 berikut :

Tabel 4. 3 Grafik *loss* dan *accuracy*









Berdasarkan tabel, dapat dilihat bahwa akurasi klasifikasi penyakit mata katarak menggunakan *Convolutional Neural Networks* (CNN) model terbaik adalah Adam. Konfigurasi 2 menjadi yang terbaik karena memiliki *accuracy* 0.1836, *Loss* 0.9271, *validation accuracy* 0.2253, dan *validation loss* 0.9000. Grafik loss menurun, dan grafik akurasi menunjukkan kecenderungan naik. Selain itu, terlihat dari grafik pelatihan dan validasi pada Konfigurasi 2 (Adam) bahwa model yang dimaksud memiliki performa yang baik dan stabil ketika menjalankan tugasnya.

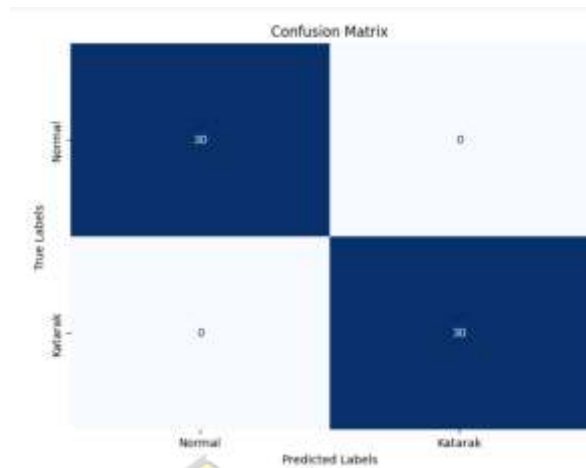
Tabel 4. 4 Hasil Konfigurasi

	<i>epoch</i>	<i>loss</i>	<i>acc</i>	<i>Vall- loss</i>	<i>Val- acc</i>	<i>Learning rate</i>
Konfigurasi 1 (SGD)	12/30	0.4496	0.8771	0.2821	0.9333	0.1
Konfigurasi 2 (Adam)	11/30	0.1836	0.9271	0.2253	0.9000	0.001
Konfigurasi 3 (RMSprop)	7/30	0.2897	0.9146	0.4262	0.8833	0.001

4.2.1. Testing

Peforma klasifikasi dapat dinilai dengan menggunakan *klasifikasi matriks* dan *confusion matriks*. *Matriks* kinerja model klasifikasi memiliki kemampuan untuk meningkatkan *F1-score*, *recall*, *akurasi*, dan *presisi*. Sedangkan *Confusion Matriks* adalah tabel yang menggambarkan cara kerja model klasifikasi menggunakan data uji real-time. Empat bagian dari *confusion matrix* adalah *true positives*, *true negatives*, *false negatives*, dan *true positives*. *True positive* mengacu pada jumlah data yang secara akurat diprediksi sebagai positif; *false positive*, untuk jumlah data yang tidak pasti menjadi positif; *true negative*, untuk jumlah data yang secara akurat diprediksi sebagai negatif dan *false negative*, untuk jumlah data yang secara akurat diprediksi sebagai negatif. *Confusin Matrik* berguna untuk mengukur akurasi, tekanan, memori, dan skor F1.

Proses permutasi ini dilakukan terhadap empat konfigurasi yang berbeda dalam arsitektur VGG19. Tujuan dari latihan ini adalah untuk mengidentifikasi konfigurasi optimal dengan karakteristik performa terbaik saat mengklasifikasikan jenis-jenis wajah berdasarkan *accuracy*, *presision*, *recall*, dan *F1-Score* dari masing-masing konfigurasi yang telah disebutkan sebelumnya. Di bawah ini adalah penjelasan mengenai proses perhitungan yang menggunakan konfigurasi. Pada konfigurasi 2, pengoptimal ADAM digunakan. Pengujian dilakukan dengan menggunakan *test* data yang telah divalidasi oleh pakar dan dapat dievaluasi dengan menggunakan *confusion matrix*. *Accuracy*, *Precision*, *Recall* dan *F1_score* dari konfigurasi 2 (ADAM) ditampilkan pada gambar 4.8



Gambar 4. 8 nilai confusion matrix

Berdasarkan nilai matrix di atas, dapat digunakan untuk menghitung akurasi, presisi, recall, dan f1-score secara otomatis, dengan hasil yang dapat dilihat pada tabel 4.5.

Tabel 4. 5 Hasil pengujian dengan augmentasi data menggunakan optimasi Adam

Class	Accuracy	Precision	Recall	F1_score
Katarak	1.00	1.00	1.00	1.00
Normal	1.00	1.00	1.00	1.00


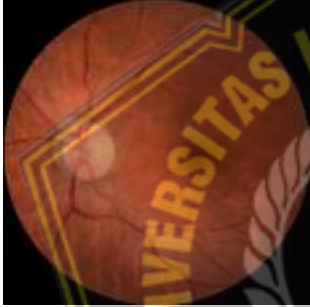

Berdasarkan hasil percobaan yang dilakukan pada setiap skenario, dapat dilihat bahwa model akurasi menunjukkan perbedaan yang sangat kecil antara hasil klasifikasi yang diperoleh dengan menggunakan augmentasi dan yang diperoleh tanpa metode ini ketika menggunakan pengoptimal Adam. Hasil dari penelitian ini menunjukkan bahwa kelompok yang menggunakan augmentasi dan yang tidak memiliki tingkat pertambahan yang hampir sama, seperti yang dapat dilihat pada tabel 4.5.

4.2.2. Hasil Testing

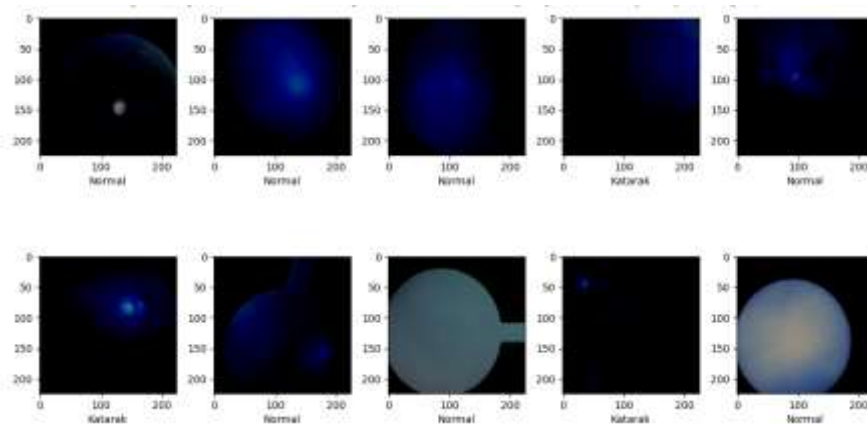
Pada bagian ini akan ditampilkan hasil percobaan menggunakan model yang dievaluasi menggunakan data uji. Pada penelitian ini, penulis memilih konfigurasi nomor dua karena pada proses *training* memiliki akurasi sebesar 0.9271, loss sebesar 0.1836, akurasi validitas sebesar 0.9000, dan loss validitas sebesar 0.2233. Selain itu, penelitian berbasis pengujian

menghasilkan hasil akurasi sebesar 1.00, hasil prediksi sebesar 1.00, hasil recall sebesar 1.00, dan hasil f1-score sebesar 1.00. Hasil pengujian dengan konfigurasi model 2 dapat ditunjukkan pada tabel 4.6.

Tabel 4. 6 Hasil Uji Coba

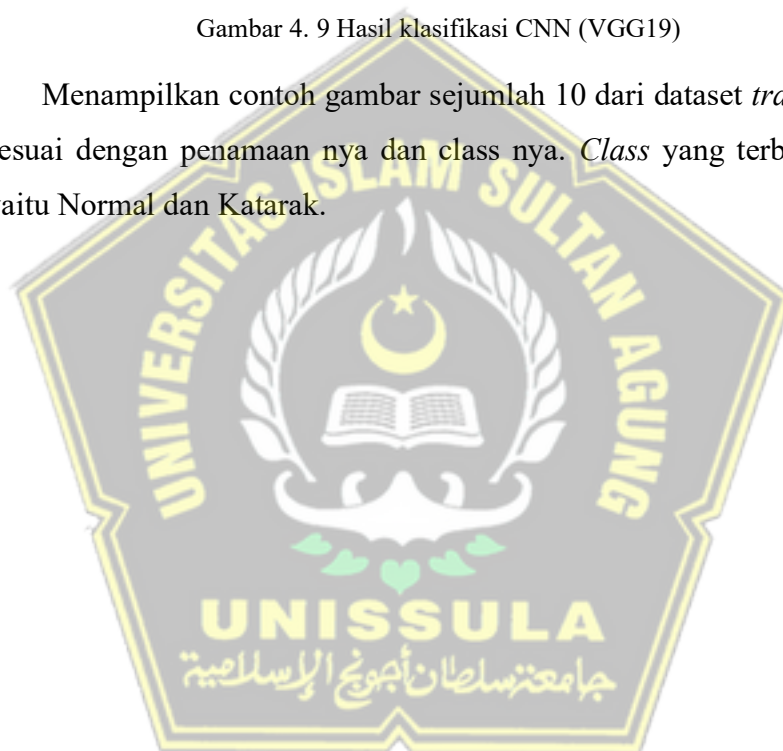
Gambar	Hasil prediksi	Keterangan
	<pre>Saving cataract_001.png to cataract_001.png 1/1 [=====] - 1x 858ms/step cataract_001.png Normal</pre>	Tidak sesuai
	<pre>* E_left.jpg=>img.jpg: 70% bytes, last modified: 2/25/2024 - 100% done Saving E_left.jpg to E_left.jpg 1/1 [=====] - 1x 671ms/step E_left.jpg Normal</pre>	Sesuai
	<pre>_3_4614982.jpg * _3_4614982.jpg=>img.jpg: 5079 bytes, last modified: 2/21/2024 - 100% done Saving _3_4614982.jpg to _3_4614982 (5) .jpg 1/1 [=====] - 1x 663ms/step Katarak</pre>	Sesuai

Seperti yang dapat dilihat dari tabel 4.6, terdapat sedikit kesalahan dalam klasifikasi mengenali penyakit katarak menggunakan VGG19. Kesalahan yang disebutkan di atas ada di kelas katarak, yang secara resmi diakui sebagai kelas normal.



Gambar 4. 9 Hasil klasifikasi CNN (VGG19)

Menampilkan contoh gambar sejumlah 10 dari dataset *train* yang sudah sesuai dengan penamaan nya dan class nya. *Class* yang terbagi menjadi 2 yaitu Normal dan Katarak.



BAB V

KESIMPULAN DAN SARAN

5.1 Kesimpulan

Dari penelitian yang telah dilakukan, dapat disimpulkan bahwa arsitektur VGG19 digunakan untuk mengkategorikan jenis katarak melalui situs web yang menampilkan performa kerja yang baik. Berdasarkan hal ini, temuan penelitian ini adalah sebagai berikut :

1. Dengan menggunakan konfigurasi pengoptimal Adam, konfigurasi optimal diwakili oleh konfigurasi 2, yang memiliki akurasi 0,9956 dan kerugian 0,0134. Selain itu, terdapat akurasi validasi sebesar 1,6791 dan loss validasi sebesar 0,8190. Hasil evaluasi data uji akurasi menunjukkan loss sebesar 0,135 dan akurasi sebesar 91,66%.
2. Model arsitektur VGG19 menyajikan proses kerja yang hampir universal dalam mengklasifikasikan jenis penyakit katarak. Hasil penelitian dengan menggunakan data *test* menunjukkan bahwa accuraccy sebesar 1.00, *Precision* sebesar 1.00, *recall* sebesar 1.00, dan *f1-score* sebesar 1.00. Terdapat sedikit kesalahan dalam klasifikasi mengenali penyakit katarak menggunakan VGG19. Kasalahan yang disebutkan di atas ada pada kelas katarak, yang ternyata dikenali sebagai kelas normal dan terjadinya *overfitting*.

5.2 Saran

Berdasarkan masukan di atas, beberapa metode penelitian untuk meningkatkan penelitian yang telah berhasil diselesaikan adalah sebagai berikut. Mengoptimalkan gradien arsitektur Convolutional Neural Network sehingga sistem dapat terus belajar dengan lebih efektif. Sistem yang baru dirancang ini dapat digunakan baik untuk pelatihan reguler maupun katarak. Ini berarti bahwa sangat mungkin untuk melakukan studi identifikasi menggunakan kumpulan data yang terkait dengan metode arsitektur

Convolutional Neural Network dan arsitektur lain yang berpotensi berguna, seperti pengoptimalisasi dan teknik parameterisasi, untuk menghasilkan hasil yang lebih baik. Diharapkan juga penelitian selanjutnya mencoba menggunakan arsitektur CNN yang lain diantaranya arsitektur Alexnet dan arsitektur Inception V3. Pada analisis berikut, Komputer dengan spesifikasi RAM yang tinggi dapat digunakan untuk memberikan hasil komputer yang lebih cepat.



BAB IV

DAFTAR PUSTAKA

- Ambarwari, Agus, Mauludi Husni, dan Dimitri Mahayana. 2023. “Perkembangan Metode Klasifikasi Citra Penginderaan Jauh dalam Perspektif Revolusi Ilmiah Thomas Kuhn.” *Jurnal Filsafat Indonesia* 6.
- Android, Berbasis, dan Menggunakan Metode. t.t. “SISTEM DETEKSI PENYAKIT MATA PTERIGIUM.”
- Bu’ulolo, Jeremia, Geza. 2021. “jm_informatika,+34208-73917-1-ED+375-382.” *Identification of Cataract eye Disease Using Convolutional Neural Network* 16: 375–82.
- Fahmi, Hasanul. 2019. “Sistem Pakar Mendiagnosa Penyakit Mata Katarak Dengan Metode Certainty Factor Berbasis Web.” *MATICS* 11 (1): 27. <https://doi.org/10.18860/mat.v11i1.7673>.
- Hananta Firdaus, Diki, Bahtiar Imran, Lalu Darmawan Bakti, dan Emi Suryadi. 2022. “KLASIFIKASI PENYAKIT KATARAK PADA MATA MENGGUNAKAN METODE CONVOLUTIONAL NEURAL NETWORK (CNN) BERBASIS WEB.” *Jurnal Kecerdasan Buatan dan Teknologi Informasi (JKBTI)*. Vol. 1.
- Inovasi Penelitian, Jurnal, Oleh Ilham Budiman, Sopyan Saori, Ramdan Nurul Anwar, Muhamamd Yuga Pangestu, Jurusan Administrasi Bisnis, Fakultas Ilmu Administrasi dan Humaniora, dan Universitas Muhammadiyah Sukabumi. t.t. “ANALISIS PENGENDALIAN MUTU DI BIDANG INDUSTRI MAKANAN (Studi Kasus: UMKM Mochi Kaswari Lampung Kota Sukabumi).”
- Marcella, Dewi, dan Siska Devella. 2022. “Klasifikasi Penyakit Mata Menggunakan Convolutional Neural Network Dengan Arsitektur VGG-19” 3 (1): 60–70.
- Nurona Cahya, Fani, Nila Hardi, Dwiza Riana, Sri Hadianti, Nusa Mandiri Jakarta Cipinang Melayu, Kec Makasar, Kota Jakarta Timur, dan Daerah Khusus Ibukota Jakarta. t.t. “SISTEMASI: Jurnal Sistem Informasi Klasifikasi Penyakit Mata Menggunakan Convolutional Neural Network (CNN).” <http://sistemasi.ftik.unisi.ac.id>.
- Peryanto, Ari, Anton Yudhana, dan Dan Rusydi Umar. 2019. “Rancang Bangun Klasifikasi Citra Dengan Teknologi Deep Learning Berbasis Metode Convolutional Neural Network.” *Jurnal*. Vol. 8.

<https://www.mathworks.com/discovery/convolutional-neural-network.html>.

Risma, Herdian Anantya, Raditiana Patmasari, dan Ir Rita Magdalena. t.t. "PERFORMANCE ANALYSIS OF CATARACT DETECTION SYSTEM USING DCT (DISCRETE COSINE TRANSFORM) AND ARTIFICIAL NEURAL NETWORK BACKPROPAGATION."

Rozaqi, Abdul Jalil, Andi Sunyoto, dan Rudyanto Arief. t.t. "Deteksi Penyakit pada Daun Kentang Menggunakan Pengolahan Citra dengan Metode Convolutional Neural Network Detection of Potato Leaves Disease Using Image Processing with Convolutional Neural Network Methods."



LAMPIRAN

A Turn It In

