

**IDENTIFIKASI ANEMIA MELALUI CITRA KONJUNGtiva MATA  
MENGUNAKAN METODE K-MEANS CLUSTERING DAN  
CONVOLUTIONAL NEURAL NETWORK**

**SKRIPSI**

**PUTRI YANTI NAHAMPUN**

**201402061**



**PROGRAM STUDI S1 TEKNOLOGI INFORMASI  
FAKULTAS ILMU KOMPUTER DAN TEKNOLOGI INFORMASI  
UNIVERSITAS SUMATERA UTARA  
MEDAN  
2024**

**IDENTIFIKASI ANEMIA MELALUI CITRA KONJUNGtiva MATA  
MENGUNAKAN METODE K-MEANS CLUSTERING DAN  
CONVOLUTIONAL NEURAL NETWORK**

**SKRIPSI**

Diajukan untuk melengkapi dan memenuhi syarat memperoleh ijazah  
Sarjana Teknologi Informasi

**PUTRI YANTI NAHAMPUN**

**201402061**



**PROGRAM STUDI S1 TEKNOLOGI INFORMASI  
FAKULTAS ILMU KOMPUTER DAN TEKNOLOGI INFORMASI  
UNIVERSITAS SUMATERA UTARA  
MEDAN  
2024**

**PERSETUJUAN**

Judul : Identifikasi Anemia Melalui Citra Konjungtiva  
Mata Menggunakan Metode K-Means Clustering  
dan Convolutional Neural Network

Kategori : Skripsi

Nama Mahasiswa : Putri Yanti Nahampun

Nomor Induk Mahasiswa : 201402061

Program Studi : Sarjana (S-1) Teknologi Informasi

Fakultas : Ilmu Komputer dan Teknologi Informasi  
Universitas Sumatera Utara

Medan, 14 Juni 2024

Komisi Pembimbing:

Pembimbing 2,



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

Pembimbing 1,



Ivan Jaya S.Si., M. Kom.  
NIP. 198407072015041001

Diketahui / disetujui oleh

Program Studi S-1 Teknologi Informasi

Ketua,



Dedy Arisandi S.T., M. Kom.  
NIP. 197908312009121002

## **PERNYATAAN**

### **IDENTIFIKASI ANEMIA MELALUI CITRA KONJUNGTTIVA MATA MENGUNAKAN METODE K-MEANS CLUSTERING DAN CONVOLUTIONAL NEURAL NETWORK**

## **SKRIPSI**

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

Medan, 14 Juni 2024



Putri Yanti Nahampun

201402061

## UCAPAN TERIMAKASIH

Puji dan syukur disampaikan kepada Tuhan Yang Maha Esa, untuk kasih dan penyertaan-Nya, sehingga diperoleh berkat yang berlimpah serta kekuatan bagi penulis dalam menyelesaikan penyusunan tugas akhir ini dengan baik, sebagai persyaratan memperoleh gelar Sarjana Komputer pada Program Studi S1 Teknologi Informasi, Fakultas Ilmu Komputer dan Teknologi Informasi Universitas Sumatera Utara.

Ungkapan terimakasih disampaikan kepada semua pihak yang berpengaruh selama masa perkuliahan dan penyelesaian tugas akhir. Maka dari itu, penulis menyampaikan rasa terima kasih yang sebesar-besarnya kepada:

1. Keluarga penulis, Bapak Saor Nahampun dan Mamak, Roasi Ritonga, serta Kakak, Lestari Nahampun dan Adik, Ferdinan Nahampun yang telah menjadi sumber motivasi penulis serta menjadi tempat berdiskusi dan berkeluh kesah. Selalu mendoakan dan mendukung penulis tanpa memberatkan penulis dalam melakukan penyelesaian skripsi.
2. Dosen pembimbing 1 penulis, Bapak Ivan Jaya S.Si., M.Kom. dan dosen pembimbing 2 penulis, Ibu Fanindia Purnamasari S.Ti., M.IT., yang telah mengarahkan, memberikan saran dan kritik, serta memotivasi penulis selama pengerjaan tugas akhir.
3. Dosen penguji 1 penulis, Bapak Dedy Arisandi S.T., M. Kom dan dosen penguji 2 penulis, Ibu Rossy Nurhasanah S. Kom., M. Kom yang telah memberikan saran dan masukan yang sangat berharga untuk penelitian ini.
4. Dekan Fasilkom-TI Universitas Sumatera Utara, Ibu Dr. Maya Silvi Lydia, M.Sc.
5. Ketua Program Studi S1 Teknologi Informasi Universitas Sumatera Utara, Bapak Dedy Arisandi S.T., M. Kom.
6. Mentor penulis pada saat mengikuti program studi independen, Kak Nicholas Dominic S.Kom., M.Kom. yang telah memberikan pemahaman tentang *image processing* sehingga penulis memiliki bekal ilmu untuk penyusunan tugas akhir ini.
7. Seluruh Dosen Program Studi S1 Teknologi Informasi yang telah memberikan banyak pengetahuan selama masa perkuliahan.

8. Seluruh staff dan pegawai Fakultas Ilmu Komputer dan Teknologi Informasi Universitas Sumatera Utara, yang telah memberikan bantuan untuk pengurusan administrasi pada masa perkuliahan.
9. Teman-teman *gurls* Monika, Della, Dinda, Irma, Mitha, dan Ruth yang telah menjadi rekan terbaik penulis dari awal masa perkuliahan sampai penyelesaian tugas akhir, saling berjuang dan memberikan doa dan dukungan terbaik.
10. Teman-teman seperbimbingan Fildzah, Dara, Amel, dan Febri yang senantiasa saling mendukung dan berjuang bersama-sama.
11. Beberapa senior baik yaitu Kak Ogestin, Kak Jogiana dan Kak Meina, serta teman sekelas Felix Salim, dan juga teman dekat Maria Lai'a yang telah banyak membantu, berdiskusi dan menjawab pertanyaan penulis terkait berbagai hal.
12. Kelompok kecil *Quattuor Oleena* Kak Frida, Kak Martha dan Ninna sebagai tempat bertumbuh dalam kasih, yang senantiasa memberikan dukungan, motivasi dan doa terbaik untuk penulis.
13. Penulis buku-buku favorit penulis, diantaranya Kyle Idleman, Leila S. Chudori, Tere Liye, Alvi Syahrin, dan Ardhi Mohamad yang melalui tulisannya memberikan motivasi dan pengetahuan yang sangat berharga bagi penulis.
14. Seluruh teman-teman angkatan 2020, senior dan junior, yang telah memberikan banyak pengalaman dan pengetahuan yang berharga untuk penulis.
15. Semua pihak yang secara langsung maupun tidak langsung turut membantu penyelesaian tugas akhir ini dan tidak dapat dituliskan satu per satu.

Medan, 14 Juni 2024

Penulis,



Putri Yanti Nahampun

201402061

## ABSTRAK

Anemia merupakan masalah kesehatan serius yang membutuhkan penanganan dini yang tepat untuk mencegah terjadinya komplikasi penyakit hingga pada kematian. Pemeriksaan anemia umumnya dapat dilakukan secara invasif dan non-invasif. Pemeriksaan secara invasif memerlukan waktu dan biaya yang mahal. Teknis pengambilan sampel darah juga dapat menimbulkan rasa sakit, sehingga beberapa masyarakat enggan untuk melakukan pemeriksaan. Sementara pemeriksaan secara non-invasif hanya memerlukan pengamatan warna pada citra konjungtiva mata, yang dinilai jauh lebih efektif daripada pemeriksaan secara invasif. Namun, pengamatan oleh indera manusia dapat bersifat subjektif sehingga diperlukan sebuah sistem identifikasi anemia. Penelitian ini dilakukan untuk membangun sistem identifikasi anemia melalui citra konjungtiva mata menggunakan metode *K-Means Clustering* dan *Convolutional Neural Network (CNN)*. Penggunaan data pada proses pelatihan berjumlah 400 gambar dan 40 gambar untuk proses pengujian. Hasil penelitian menunjukkan kemampuan sistem dalam mengenali warna pada citra konjungtiva mata penderita anemia dan non-anemia dengan tingkat akurasi sebesar 95%.

**Kata Kunci:** Identifikasi Anemia, Konjungtiva Mata, CNN, K-Means Clustering.

**IDENTIFICATION OF ANEMIA THROUGH CONJUNCTIVAL EYE IMAGE  
USING K-MEANS CLUSTERING AND CONVOLUTIONAL NEURAL  
NETWORK**

**ABSTRACT**

*Anemia is a serious health problem that, if not promptly and accurately addressed, can lead to death. Anemia examinations can generally be performed invasively and non-invasively. Invasive examinations require time and expensive costs. The technical process of blood sampling can also cause discomfort, leading some people to hesitate in undergoing examinations. Meanwhile, non-invasive examinations only require observation of the paleness of the color on the conjunctival eye, which is considered far more effective than invasive examinations. However, human observation can be subjective, necessitating a system for identifying anemia using the K-Means Clustering and Convolutional Neural Network (CNN) methods. The data used consisted 400 images for the training process and 40 images for the testing process. The research results show the system's ability to recognize conjunctival eye images of individuals with anemia or non-anemia with an accuracy rate of 95%.*

**Keywords:** *Anemia Identification, Conjunctival Eye, CNN, K-Means Clustering.*



**DAFTAR ISI**

PERSETUJUAN	iii
PERNYATAAN	iv
UCAPAN TERIMAKASIH	v
ABSTRAK	vii
ABSTRACT	viii
DAFTAR ISI	ix
DAFTAR TABEL	xi
DAFTAR GAMBAR	xii
BAB 1 PENDAHULUAN	1
1.1 Latar Belakang	1
1.2 Rumusan Masalah	4
1.3 Tujuan Penelitian	4
1.4 Batasan Masalah	4
1.5 Manfaat Penelitian	4
1.6 Metodologi Penelitian	5
1.7 Sistematika Penulisan	5
BAB 2 LANDASAN TEORI	7
2.1 Anemia	7
2.2 Konjungtiva Mata	7
2.3 Citra Digital	8
2.3.1 Citra RGB	8
2.4 Pengolahan Citra Digital	9
2.5 Convolutional Neural Network	9
2.5.1 Convolutional Layer	10
2.5.2 Activation Function	11
2.5.3 Pooling Layer	11
2.5.4 Fully Connected Layer	11
2.6 K-Means Clustering	12
2.6.1 Dice Coefficient (Dice Score)	12
2.7 Confusion Matrix	14
2.7.1 Accuracy	15
2.7.2 Precision	15

2.7.3 Recall	15
2.7.4 F-1 Score	15
2.8 Penelitian Terdahulu	16
2.9 Perbedaan Penelitian	21
<b>BAB 3 ANALISIS DAN PERANCANGAN SISTEM</b>	<b>22</b>
3.1 Data yang digunakan	22
3.2 Analisis Sistem	22
3.2.1 Data acquisition	24
3.2.2 Image pre-processing	24
3.2.3 Image segmentation	30
3.2.4 Image identification dengan Convolutional Neural Network	39
3.2.5 Deployment	45
3.2.6 Testing process	46
3.3 Perancangan Antarmuka Sistem	46
3.3.1 Rancangan tampilan splashscreen	47
3.3.2 Rancangan tampilan halaman utama	48
3.3.3 Rancangan tampilan halaman deteksi	48
3.3.4 Rancangan tampilan halaman hasil deteksi	49
<b>BAB 4 IMPLEMENTASI DAN PENGUJIAN</b>	<b>50</b>
4.1 Implementasi Sistem	50
4.1.1 Perangkat keras dan perangkat lunak	50
4.1.2 Implementasi data	50
4.1.3 Implementasi desain interface	52
4.2 Prosedur Operasional	54
4.3 Pelatihan Sistem	55
4.3.1 Percobaan nilai batch size 30	57
4.3.2 Percobaan nilai batch size 50	58
4.3.3 Percobaan nilai batch size 64	58
4.4 Pengujian Sistem	60
<b>BAB 5 PENUTUP</b>	<b>67</b>
5.1 Kesimpulan	67
5.2 Saran	67
<b>DAFTAR PUSTAKA</b>	<b>xiii</b>
<b>LAMPIRAN</b>	<b>1</b>

**DAFTAR TABEL**

<b>Tabel 2.1</b> Confusion Matrix	14
<b>Tabel 2.2</b> Rangkuman Penelitian Terdahulu	18
<b>Tabel 3.1</b> Pembagian Data	22
<b>Tabel 3.2</b> Perhitungan Nilai Intensitas Citra	31
<b>Tabel 4.1</b> Pembagian Data Setelah Augmentasi	51
<b>Tabel 4.2</b> Perbandingan Hasil Training	59
<b>Tabel 4.3</b> Pengujian Identifikasi Anemia	60
<b>Tabel 4.4</b> Perhitungan Hasil Uji	63
<b>Tabel 4.5</b> Nilai Precission, Recall, dan F-1 Score	64
<b>Tabel 4.6</b> Percobaan Berdasarkan Cahaya Pengambilan	65

## DAFTAR GAMBAR

<b>Gambar 2.1</b> Konjungtiva Anemia (atas) & Non-anemia (bawah)	8
<b>Gambar 2.2</b> Citra RGB	9
<b>Gambar 2.3</b> Arsitektur CNN	10
<b>Gambar 2.4</b> Proses Segmentasi Dengan K-Means Clustering	13
<b>Gambar 3.1</b> Citra Konjungtiva Anemia dan Non-anemia	22
<b>Gambar 3.2</b> Arsitektur Umum	24
<b>Gambar 3.3</b> Citra Sebelum dan Sesudah Cropping	25
<b>Gambar 3.4</b> Citra Hasil Resizing	25
<b>Gambar 3.5</b> Labelling	29
<b>Gambar 3.6</b> Citra Sebelum dan Sesudah Horizontal Flip	29
<b>Gambar 3.7</b> Citra Sebelum dan Sesudah Brightness Range	29
<b>Gambar 3.8</b> Citra Konjungtiva Mata dan Nilai Intensitas Pikselnya	30
<b>Gambar 3.9</b> Perhitungan Variasi Intensitas Pada Piksel Citra	36
<b>Gambar 3.10</b> Perubahan Citra Asli ke Citra Hasil Equalization	32
<b>Gambar 3.11</b> Perubahan Citra Equalization ke Hasil Unsharp Masking Filter	35
<b>Gambar 3.12</b> Grafik Elbow	36
<b>Gambar 3.13</b> Citra Hasil Segmentasi	38
<b>Gambar 3.14</b> Proses Perhitungan Filter Convolutional Layer	40
<b>Gambar 3.15</b> Pergeseran Filter Pada Proses Konvolusi	40
<b>Gambar 3.16</b> Activity Diagram	49
<b>Gambar 3.17</b> Rancangan Halaman Splashscreen	49
<b>Gambar 3.18</b> Rancangan Halaman Utama	49
<b>Gambar 3.19</b> Rancangan Halaman Deteksi	49
<b>Gambar 3.20</b> Rancangan Halaman Hasil Deteksi	49
<b>Gambar 4.1</b> Citra Konjungtiva Anemia	51
<b>Gambar 4.2</b> Citra Konjungtiva Non-anemia	51
<b>Gambar 4.3</b> Tampilan Halaman Splashscreen	52
<b>Gambar 4.4</b> Tampilan Halaman Utama	52
<b>Gambar 4.5</b> Tampilan Halaman Deteksi, Galeri, Kamera	53
<b>Gambar 4.6</b> Tampilan Halaman Hasil Deteksi	53
<b>Gambar 4.7</b> Tampilan Halaman panduan	54

<b>Gambar 4.8</b> Tampilan Halaman Informasi Anemia	54
<b>Gambar 4.9</b> Grafik Akurasi dan Loss Batch Size 30	57
<b>Gambar 4.10</b> Grafik Akurasi dan Loss Batch Size 50	58
<b>Gambar 4.11</b> Grafik Akurasi dan Loss Batch Size 64	59
<b>Gambar 4.12</b> Confusion Matrix	63

# **BAB 1**

## **PENDAHULUAN**

### **1.1 Latar Belakang**

Anemia merupakan penyakit dengan kandungan hemoglobin dalam darah yang lebih rendah dari kandungan normalnya. Penyakit ini ditandai dengan kelelahan, kulit sedikit lebih pucat hingga keadaan hiperdinamik seperti denyut nadi dan detak jantung yang bekerja sedikit lebih cepat (Fitriany et al., 2018). Anemia menjadi masalah kesehatan serius yang mempengaruhi sekitar 40% balita, 30% remaja perempuan dan 37% wanita hamil diseluruh dunia, termasuk di Indonesia (WHO, 2023). Prevalensi anemia di Indonesia meningkat dari 18,40% menjadi 32% antara tahun 2013 dan 2018 dengan sekitar 14,7 juta penduduk Indonesia, menderita anemia (Riskesdas, 2018).

Dalam beberapa kasus, anemia dapat bersifat ringan, namun dapat menjadi masalah kesehatan serius jika tidak memperoleh penanganan yang tepat. Terdapat 41 kasus penyakit anemia yang diperkirakan terjadi setiap harinya di Indonesia dengan 20 orang diantaranya khususnya perempuan, meninggal dunia. Masalah ini disebabkan oleh kurangnya kesadaran dan pengetahuan masyarakat terhadap gejala, risiko, dan dampak dari anemia (Yuliatin, 2018).

Pemeriksaan anemia umumnya dilakukan secara invasif yaitu melalui pengujian kadar hemoglobin pada sampel darah. Namun pemeriksaan ini memerlukan biaya dan waktu yang lama. Teknis pengambilan sampel darah juga dapat menyebabkan rasa sakit sehingga sebagian masyarakat enggan untuk melakukan pemeriksaan secara invasif (Amalia et al., 2023).

Pemeriksaan anemia dapat dilakukan secara non-invasif yaitu melalui pengamatan terhadap warna pada konjungtiva mata. Pemeriksaan ini merupakan pemeriksaan alternatif yang dapat digunakan oleh masyarakat untuk mengenali gejala anemia. Warna pada konjungtiva mata dapat menjadi parameter untuk diagnosis beberapa penyakit. Beberapa kondisi kesehatan seperti hipoksia dan hiperbilirubinemia, dapat mempengaruhi warna pada konjungtiva mata menjadi biru dan kuning pucat. Warna pada konjungtiva mata merupakan ciri fisik paling efektif untuk pemeriksaan anemia dibandingkan dengan ciri fisik lainnya, seperti kuku dan telapak tangan (Sheth et al., 1997). Namun, pengamatan manusia dapat bersifat subjektif, sehingga diperlukan

sebuah sistem untuk identifikasi anemia yang dapat mencegah kesalahan dugaan terhadap warna pada konjungtiva mata penderita anemia.

Beberapa penelitian terbaru menunjukkan bahwa warna pada konjungtiva mata dapat menjadi indikator awal penyakit anemia. Penelitian yang dilakukan oleh Asare et al. (2023) menggunakan 424 gambar konjungtiva mata anemia dan 286 gambar konjungtiva non-anemia. Dataset tersebut digunakan untuk pelatihan model CNN (*Convolutional Neural Network*) dengan metode pemrosesan citra *Gaussian Blur* dan penambahan metode klasifikasi *Logistic Regression*, sehingga menghasilkan akurasi sebesar 92.5%. Dengan menggunakan dataset yang sama, penelitian lainnya memanfaatkan arsitektur model AlexNet untuk mengetahui potensi model *pre-trained transfer learning* dalam mengidentifikasi anemia melalui gambar konjungtiva mata. Hasil akurasi yang diperoleh adalah 85%. (Amalia et al., 2023). Penelitian berikutnya menggunakan metode HOG (*Histogram of Oriented Gradients*) dengan *Decision Tree Classification* dan *Logistic Regression* untuk melakukan deteksi pada 100 gambar konjungtiva mata. Akurasi yang diperoleh adalah 92.5% (Salsabila et al., 2022). Penggabungan metode lainnya juga dilakukan pada penelitian Asiyah et al. (2022), dengan menggunakan metode PCA (*Principal Component Analysis*) dan K-NN (*K-Nearest Neighbor*) untuk mengidentifikasi anemia melalui 100 citra konjungtiva mata. Hasil akurasi yang diperoleh adalah 87.5%. Penelitian lainnya (Noor et al., 2019) membandingkan performa algoritma *Decision Tree*, SVM, dan K-NN dengan 81 citra konjungtiva mata untuk data pelatihan dan 23 citra konjungtiva mata untuk data pengujian. Hasil terbaik diperoleh oleh *Decision Tree* yaitu sebesar 82,61%.

Penelitian-penelitian sebelumnya yang menggunakan metode pemrosesan gambar seperti *Gaussian Blur* dan PCA, menghasilkan performa model yang cukup baik. Hal ini mengindikasikan bahwa teknik pemrosesan gambar merupakan proses penting yang dapat mempengaruhi kinerja algoritma yang digunakan. *K-Means Clustering* merupakan algoritma segmentasi gambar yang banyak diimplementasikan pada tugas-tugas segmentasi, khususnya segmentasi warna. Penelitian oleh Mehmood et al. (2022) menggunakan algoritma *K-Means Clustering* untuk segmentasi warna pada 500 citra kulit vitiligo. Penelitian membuktikan keberhasilan metode ini dalam mengelompokkan warna kulit vitiligo dan non-vitiligo dalam satu gambar dengan benar. Penelitian lainnya menggabungkan *K-Means Clustering* dengan algoritma SVM (*Support Vector Machine*) untuk deteksi citra api. *K-Means Clustering* berperan untuk melakukan

segmentasi pada gambar, dengan mengambil warna citra api yang dibutuhkan, sebelum digunakan oleh algoritma SVM. Penelitian ini menghasilkan akurasi sebesar 90% (Febrianti, 2018). Pengujian algoritma *K-Means Clustering* pada penelitian-penelitian tersebut menunjukkan performa yang baik, khususnya untuk tugas identifikasi citra yang membutuhkan segmentasi warna seperti citra konjungtiva mata.

Metode CNN merupakan algoritma pengenalan gambar dengan arsitektur yang dapat dirancang sesuai kebutuhan sehingga banyak diimplementasikan, khususnya pada tugas yang membutuhkan arsitektur model yang lebih spesifik. Arsitektur CNN dapat mempelajari fitur-fitur penting pada gambar, sehingga cocok digunakan pada dataset kecil atau data dengan karakteristik yang lebih sederhana. Citra hasil segmentasi merupakan salah satu data visual yang memiliki karakteristik yang sederhana sehingga cocok digunakan pada metode CNN. Banik et al. (2020) memakai data BCCD (*Blood Cell Count and Detection*) dengan jumlah data yang sedikit yaitu 365 citra sel darah putih untuk mengelompokkan jenis sel darah putih melalui warna *nucleus* (inti sel) nya. Penelitian memanfaatkan metode *K-Means Clustering* untuk proses segmentasi warna *nucleus* dari sel darah putih. Proses segmentasi akan menghasilkan citra objek yang dibutuhkan saja, yaitu *nuclues*, sehingga karakteristik yang akan diekstrak lebih sederhana. Citra hasil segmentasi akan digunakan sebagai *input* pada arsitektur CNN yang telah dirancang sesuai dengan tugas klasifikasi. Metode CNN berhasil melakukan klasifikasi dengan akurasi yang dicapai sebesar 97%. Pengujian penggunaan CNN dan *K-Means Clustering* pada penelitian tersebut menunjukkan performa yang baik. Penggunaan *K-Means Clustering* yang menghasilkan segmentasi warna yang baik, serta arsitektur CNN yang berhasil mengklasifikasi citra dengan data latih hasil segmentasi, menunjukkan potensi kedua metode yang dapat diimplementasikan, salah satunya pada tugas identifikasi anemia melalui warna pada konjungtiva mata manusia.

Pengembangan sistem untuk identifikasi anemia melalui citra konjungtiva mata dapat menjadi solusi alternatif untuk pemeriksaan anemia secara non-invasif. Sistem ini dapat memudahkan masyarakat untuk melakukan pemeriksaan dini sehingga dapat mengambil tindakan pencegahan dan pengobatan yang tepat. Berdasarkan latar belakang masalah tersebut, maka penulis mengangkat penelitian dengan judul **“IDENTIFIKASI ANEMIA MELALUI CITRA KONJUNGTIVA MATA MENGGUNAKAN METODE K-MEANS CLUSTERING DAN CONVOLUTIONAL NEURAL NETWORK”**.



## 1.2 Rumusan Masalah

Pemeriksaan dini terhadap anemia merupakan tindakan penting untuk mencegah anemia yang lebih parah. Namun pemeriksaan yang dilakukan pada umumnya adalah pemeriksaan secara invasif yang membutuhkan biaya dan waktu yang lama. Pemeriksaan lainnya dapat dilakukan secara non-invasif yaitu melalui pengamatan warna pada citra konjungtiva mata. Namun pengamatan manusia dapat bersifat subjektif sehingga kesalahan dugaan dapat terjadi.

## 1.3 Tujuan Penelitian

Untuk melakukan identifikasi anemia pada citra konjungtiva mata menggunakan metode *K-Means Clustering* dan *Convolutional Neural Network* sehingga dapat diketahui apakah seseorang menderita anemia atau tidak serta memperoleh rekomendasi penanganan dan pencegahan awal yang tepat.

## 1.4 Batasan Masalah

Beberapa batasan masalah pada penelitian ini, yaitu:

1. Penyakit yang diidentifikasi adalah anemia.
2. Data yang digunakan adalah citra konjungtiva mata.
3. Identifikasi dilakukan berdasarkan warna pada citra konjungtiva mata.
4. Pengambilan citra dengan menggunakan kamera *smartphone* 16 MP.
5. Output yang dihasilkan dari penelitian ini adalah pengimplementasian model dalam sistem berbasis *Android* yang dapat mengidentifikasi gambar konjungtiva mata yang digunakan oleh pengguna kedalam kategori anemia atau non-anemia beserta solusi penanganannya.

## 1.5 Manfaat Penelitian

Beberapa manfaat penelitian ini, yaitu:

1. Memudahkan masyarakat untuk melakukan pemeriksaan awal anemia secara non-invasif sehingga dapat dilakukan penanganan dini beserta pencegahan dan perawatan yang dibutuhkan.
2. Mengetahui kinerja metode *K-Means Clustering* dan *Convolutional Neural Network* untuk identifikasi anemia melalui citra konjungtiva mata. Hasil implementasi kedua metode dapat digunakan sebagai pembelajaran untuk penelitian-penelitian berikutnya.

## 1.6 Metodologi Penelitian

Beberapa tahap yang dilakukan pada penelitian ini, yaitu:

### 1. Studi Literatur

Tahap pertama adalah proses analisis jurnal, buku, artikel, dan sumber referensi lainnya tentang penyakit anemia, hubungan antara konjungtiva mata dengan anemia, konsep *image processing*, metode CNN, dan implementasi *K-Means Clustering* pada segmentasi warna citra digital.

### 2. Analisis Permasalahan

Tahap ini merupakan proses pemahaman masing-masing topik yang telah ditemukan sebelumnya pada tahap studi literatur, yaitu dengan memahami potensi penggunaan CNN dan algoritma *K-Means Clustering* dalam mengidentifikasi penyakit anemia melalui citra konjungtiva mata manusia.

### 3. Perancangan Sistem

Pada bagian ini dilakukan perancangan pada arsitektur umum dan desain antarmuka sistem, serta penentuan data latih dan data uji.

### 4. Implementasi

Hasil perancangan sistem akan diimplementasikan dan dianalisis untuk menghasilkan sistem yang sesuai dengan tujuan penelitian.

### 5. Pengujian Sistem

Hasil dari tahap implementasi akan digunakan pada proses pengujian sistem untuk mengetahui tingkat akurasi dari metode CNN dan *K-Means Clustering* dalam mengidentifikasi anemia melalui citra konjungtiva mata manusia.

### 6. Penyusunan Laporan

Bagian ini merupakan tahap penyusunan laporan yang memuat keseluruhan hasil akhir penelitian.

## 1.7 Sistematika Penulisan

Sistematika penulisan pada penelitian ini terdiri atas lima bagian, yaitu:

### Bab 1: Pendahuluan

Bab pertama mencakup latar belakang, rumusan masalah, tujuan penelitian, batasan masalah, manfaat penelitian, metodologi penelitian, dan sistematika penulisan pada penelitian.

### Bab 2: Landasan Teori

Bab kedua mencakup sekumpulan teori yang berhubungan dengan penelitian serta berkaitan dengan masalah dalam penelitian. Beberapa teori diantaranya adalah teori tentang anemia, konjungtiva mata, pengolahan citra digital, metode segmentasi, algoritma CNN dan *K-Means Clustering*.

### **Bab 3: Analisis dan Perancangan**

Bab ketiga menjelaskan tentang analisis data, arsitektur umum dan penjelasannya, serta rancangan sistem yang akan dibangun.

### **Bab 4: Implementasi dan Pengujian Sistem**

Bab keempat mencakup implementasi dari perancangan sistem serta hasil pengujian dan analisis sistem yang telah dibangun.

### **Bab 5: Kesimpulan dan Saran**

Bab kelima memuat kesimpulan dan saran dari penelitian yang telah dilakukan untuk penelitian berikutnya.

## **BAB 2**

### **LANDASAN TEORI**

#### **2.1 Anemia**

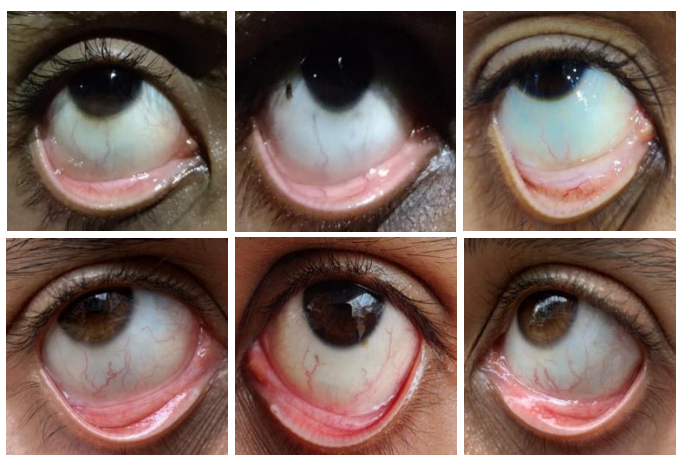
Anemia merupakan kondisi medis yang ditandai dengan turunnya kandungan hemoglobin dalam darah menjadi lebih rendah dari kadar normalnya. Hemoglobin (Hb) merupakan protein sel darah merah yang digunakan untuk mendistribusikan oksigen ke seluruh tubuh. Rendahnya kadar Hb dapat mengakibatkan tubuh tidak memperoleh cukup oksigen. Hal ini akan menyebabkan munculnya gejala pada penderita anemia, seperti kelelahan, kulit sedikit lebih pucat hingga keadaan hiperdinamik seperti denyut nadi dan detak jantung yang bekerja sedikit lebih cepat. Kadar Hb normal pada laki-laki dewasa berkisar 14-18 g/dL sedangkan untuk wanita dewasa berkisar 12-16 g/dL. Seseorang memiliki kadar Hb rendah apabila memiliki Hb kurang dari 13,5 g/dL pada laki-laki dewasa serta kurang dari 12 g/dL pada wanita dewasa (Muhayati et al., 2019).

Anemia dapat diidentifikasi melalui dua jenis pemeriksaan, yaitu pemeriksaan secara invasif dan non-invasif. Pemeriksaan secara invasif umumnya dilakukan dengan pengambilan sampel darah untuk mengetahui kadar Hb pada darah melalui pengujian laboratorium sehingga diagnosis penyakit dapat dilakukan. Pemeriksaan ini membutuhkan biaya dan waktu yang lama. Teknis pengambilan sampel darah juga dapat menyebabkan rasa sakit sehingga sebagian masyarakat enggan untuk melakukan pemeriksaan secara invasif. Jenis pemeriksaan lainnya dapat dilakukan secara non-invasif, yaitu melalui pengamatan terhadap warna pada konjungtiva mata. Pemeriksaan ini merupakan alternatif yang lebih praktis untuk membantu masyarakat mengenali gejala anemia sehingga dapat melakukan pencegahan terhadap anemia (Amalia et al., 2023).

#### **2.2 Konjungtiva Mata**

Konjungtiva mata merupakan membran tipis dan transparan yang melindungi permukaan bagian dalam kelopak mata dan mengandung banyak pembuluh darah sehingga memberikan warna kemerahan pada konjungtiva mata. Membran tipis dan transparan ini akan menghasilkan lapisan lendir yang berfungsi untuk mencegah iritasi pada bagian dalam kelopak mata (Handoyo, 2006). Konjungtiva mata bersifat sensitif

sehingga mudah untuk dipengaruhi oleh beberapa penyakit, salah satunya anemia. Penurunan kadar Hb pada anemia, akan mempengaruhi kondisi fisik penderitanya, termasuk kondisi warna pada area konjungtiva mata yang mengandung banyak pembuluh darah sehingga gejala anemia dapat dilihat dari warna pada konjungtiva mata, dari warna merah memudar hingga pada pucat putih kelabu (Chai et al., 2021). Kekurangan *eritrosit* (sel darah merah) menyebabkan darah tidak dapat didistribusikan dengan merata sehingga mengakibatkan perubahan warna pada konjungtiva mata. Contoh konjungtiva mata penderita anemia dan konjungtiva mata sehat dicantumkan pada Gambar 2.1.



**Gambar 2.1** Konjungtiva Anemia (atas) & Non-anemia (bawah)

*Sumber: (Dimauro et al. (2023))*

## 2.3 Citra Digital

Citra digital merupakan gambaran yang diciptakan oleh intensitas cahaya pada permukaan dua dimensi. Citra terbentuk dari sekumpulan piksel berkoordinat  $(x, y)$  sebagai penunjuk lokasi piksel dan amplitudo  $f(x, y)$  yang menampilkan tingkatan warna pada piksel. Citra digital terdiri atas citra RGB, citra *grayscale*, dan citra biner.

### 2.3.1 Citra RGB

Citra RGB terdiri atas piksel-piksel dengan warna merah (*red*), hijau (*green*), dan biru (*blue*) yang berukuran  $(m \times n) \times 3$ . Setiap piksel yang membentuk citra memperlihatkan kombinasi dan tingkat kecerahan warnanya. Citra ini mempunyai kedalaman warna senilai 24-bit dengan masing-masing elemen RGB senilai 8-bit. Hal ini berarti terdapat 256 intensitas cahaya untuk setiap komponen dan menghasilkan sekitar 16 juta

kombinasi warna yang berbeda. Citra RGB disebut juga sebagai *true color* karena dapat mencakup spektrum warna yang sangat luas.



**Gambar 2.2** Citra RGB

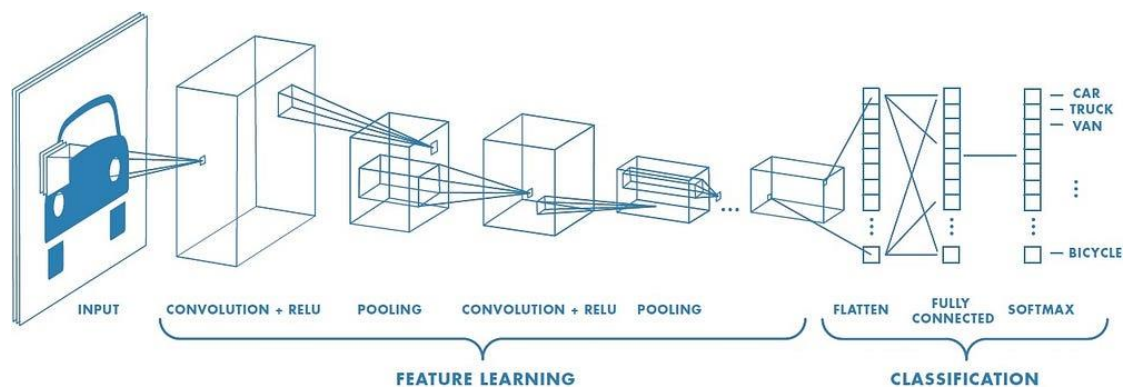
## 2.4 Pengolahan Citra Digital

Pengolahan citra digital dilakukan dengan meningkatkan kualitas citra melalui analisis citra, dalam berbagai aspek. Aspek geometrik berupa proses rotasi, translasi, skala, dan transformasi geometrik lainnya dilakukan untuk mengubah posisi dan bentuk objek dalam citra. Aspek pencahayaan meningkatkan kontras dan transformasi citra untuk memperbaiki serta mempertajam informasi yang diperoleh dari objek dalam citra. Aspek lainnya melibatkan pengecilan atau reduksi data pada citra yang bertujuan untuk mempermudah proses penyimpanan, transfer dan waktu proses data citra. Secara keseluruhan, pengolahan citra digital memiliki beragam tujuan, mulai dari peningkatan resolusi gambar, perbaikan informasi objek, pengenalan objek dalam citra, hingga kemudahan dalam pengelolaan dan pengolahan data citra secara efisien (Hermawati, 2013).

## 2.5 Convolutional Neural Network

CNN (*Convolutional Neural Network*) merupakan arsitektur jaringan saraf tiruan yang dapat melakukan analisis citra digital. CNN diperkenalkan pertama kali oleh Yann LeCun pada 1998 dan menjadi fondasi utama bagi perkembangan *deep learning*. CNN memiliki jaringan saraf *multilayer perceptron* dimana setiap *neuron* terkoneksi dengan semua *neuron* pada lapisan selanjutnya. Arsitektur ini memiliki kemampuan untuk mengenali pola pada data secara hierarkis dengan penggabungan piksel kecil dan sederhana menjadi pola yang lebih kompleks sehingga efektif untuk mengatasi keterkaitan dan kompleksitas piksel dalam citra (Rasywir et al., 2020).

CNN memiliki beberapa komponen utama yaitu lapisan masuk (*input layer*), lapisan konvolusi (*convolution layer*), fungsi aktivasi (*activation function*), lapisan penggabungan (*pooling layer*), dan lapisan terhubung (*fully connected layer*). Lapisan masuk merupakan lapisan penerima input dengan matriks piksel. Citra pada lapisan input akan digunakan pada lapisan berikutnya, yaitu lapisan konvolusi. Lapisan konvolusi menggunakan operasi matematis dengan filter atau kernel, untuk menghasilkan peta fitur yang menggambarkan ekstraksi fitur pada gambar seperti tepi, warna, atau pola tertentu. Proses konvolusi akan dibantu oleh fungsi aktivasi untuk membantu model CNN dalam mempelajari pola-pola yang lebih kompleks. Hasil dari proses konvolusi akan digunakan pada lapisan penggabungan untuk mengurangi kompleksitas dan ukuran data sambil mempertahankan informasi penting. Hasil ekstraksi fitur kemudian akan masuk ke lapisan terhubung untuk menggabungkan fitur-fitur yang telah diekstrak dan menghasilkan output sesuai dengan tugas identifikasi atau klasifikasi yang diinginkan (Azmi et al., 2023).



**Gambar 2.3** Arsitektur CNN

Sumber: (Azmi et al., 2023)

### 2.5.1 Convolutional Layer

Lapisan konvolusi (*convolutional layer*) merupakan lapisan yang berfungsi untuk melakukan ekstraksi fitur-fitur penting pada citra. Proses ekstraksi fitur akan melibatkan perhitungan antara nilai filter dengan nilai piksel pada citra sebagai input. Filter akan bergerak melintasi piksel pada citra dan akan melakukan pergerakan sebanyak satu stride. Selama proses pergerakan, akan dilakukan perkalian antara nilai piksel dan filter sehingga dihasilkan nilai-nilai yang membentuk *feature map*, untuk dijadikan input pada *pooling layer* dan lapisan konvolusi berikutnya. secara matematis, proses konvolusi dapat dinyatakan pada Persamaan 2.1.

$$(f \times g)(t) = \sum_T f(T) \cdot g(t - T) \quad (2.1)$$

Keterangan:

- $f$  adalah input data (gambar)
- $g$  adalah kernel/filter
- $t$  adalah posisi output dimana hasil konvolusi dihitung
- $T$  adalah posisi relatif dalam input yang digunakan pada perhitungan konvolusi
- $f(T)$  adalah nilai input pada posisi  $T$
- $g(t-T)$  adalah nilai dari kernel pada posisi  $(t-T)$

### 2.5.2 Activation Function

Fungsi aktivasi (*activation function*) merupakan fungsi yang digunakan untuk menambahkan non-linearitas ke model yang memungkinkan jaringan untuk belajar dan mewakili hubungan yang kompleks dalam data. Contoh fungsi aktivasi yang umum digunakan adalah ReLu (*Rectified Linear Unit*) yang dapat dilihat pada Persamaan 2.2.

$$\text{ReLu}(x) = \max(0, x) \quad (2.2)$$

### 2.5.3 Pooling Layer

Lapisan ini berfungsi untuk mengurangi dimensi spasial dari *feature map* dengan mengurangi jumlah parameter dan komputasi dalam jaringan sehingga model lebih tahan terhadap pergeseran dan distorsi kecil dalam gambar. Jenis *pooling layer* yang umum digunakan adalah *MaxPooling* dengan memilih nilai maksimum dari area *pooling* dan *AveragePooling* dengan menghitung nilai rata-rata pada area *pooling*. Penggunaan *MaxPooling* dapat dilihat pada Persamaan 2.3.

$$\text{MaxPooling} = \max(\text{region}) \quad (2.3)$$

### 2.5.4 Fully Connected Layer

*Fully Connected Layer* berfungsi untuk menghubungkan setiap neuron di lapisan sebelumnya dengan neuron pada lapisan berikutnya. Lapisan ini akan menggabungkan fitur-fitur hasil ekstraksi untuk membuat keputusan akhir dalam proses klasifikasi.



## 2.6 K-Means Clustering

*K-Means Clustering* adalah algoritma segmentasi citra untuk mengelompokkan piksel pada citra berdasarkan kemiripan intensitas pikselnya. Metode ini akan mengidentifikasi area pada gambar yang memiliki karakteristik warna yang sama dengan cara mengelompokkan piksel-piksel citra kedalam kelompok yang memiliki nilai rata-rata warna yang mirip.

Segmentasi citra dengan *K-Means Clustering* diawali dengan penentuan jumlah kluster atau segmen yang diinginkan. Selanjutnya, algoritma secara acak akan menentukan posisi awal pusat kluster. Proses pengelompokkan pada *K-Means Clustering* dilakukan dengan meminimalkan jarak antara setiap piksel dengan pusat kluster terdekat, dimana pusat kluster akan diperbaharui sesuai dengan nilai rata-rata piksel yang termasuk kedalam kluster tersebut. Pusat kluster merupakan representasi dari kelompok piksel yang terdapat dalam kluster, dengan persamaan matematis pada Persamaan 2.4.

$$E_k = \sum_{i=1}^n \sum_{k=1}^K ||x^i - m^k||^2 \quad (2.4)$$

Keterangan:

- $E_k$  adalah jumlah kuadrat jarak untuk kluster ke-k
- $x^i$  sebagai data ke-i
- $m^k$  sebagai pusat kluster ke-k
- $K$  sebagai jumlah kluster
- $n$  sebagai jumlah total data

Persamaan untuk menentukan pengelompokan pada *K-Means Clustering* dapat dilihat pada Persamaan 2.5.

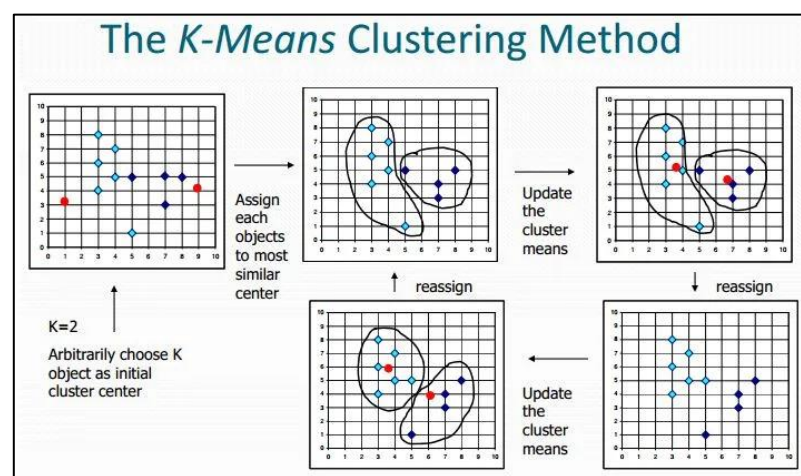
$$y^i = \operatorname{argmin}_k ||x^i - m^k||^2 \quad (2.5)$$

Keterangan:

- $y^i$  adalah indeks kluster yang dipilih untuk data ke-i
- $k$  adalah indeks kluster

- $\|x^i - m^k\|^2$  adalah jarak kuadrat antara data ke-i ( $x_i$ ) dan pusat kluster ke-k ( $m_k$ )
- `argmin` operator yang mengembalikan indeks dengan nilai minimum

Persamaan 2.5 digunakan untuk mengelompokkan setiap titik sesuai dengan kluster yang memiliki jarak paling dekat dengan titik tersebut. Titik yang sudah tergabung dengan kluster tertentu akan mendapat indeks kluster baru. Kelompok kluster yang terbentuk akan memperbaharui pusat kluster, berdasarkan rata-rata posisi titik dalam kluster tersebut. Proses akan berulang hingga konvergensi pada pengelompokan data tercapai.



**Gambar 2.4** Proses Segmentasi Dengan K-Means Clustering  
(Sumber: : Gonzales & Woods, 2008)

### 2.6.1 Dice coefficient (Dice Score)

*Dice coefficient* merupakan metrik penilaian yang digunakan dalam evaluasi performa segmentasi citra dengan mengukur kesamaan antara dua set data, yang dalam konteks segmentasi citra adalah hasil segmentasi prediksi dan *ground truth*. Secara matematis, metrik dapat dinyatakan pada Persamaan 2.6

$$Dice = \frac{2 \cdot |A \cap B|}{|A| + |B|} \quad (2.6)$$

Keterangan:

- A adalah himpunan piksel yang diprediksi sebagai bagian dari objek oleh model segmentasi
- B adalah himpunan piksel yang sebenarnya merupakan bagian dari objek (*ground truth*)

- $|A \cap B|$  adalah jumlah piksel yang diprediksi benar oleh model.

Metrik penilaian ini akan menghasilkan nilai berkisar 0 dan 1, dengan 0 menunjukkan bahwa tidak ada tumpang tindih antara prediksi dan *ground truth*, dan nilai 1 menunjukkan kesamaan sempurna antara prediksi dan *ground truth*. Proses perhitungan *Dice Coefficient* dari hasil segmentasi dilakukan dengan langkah-langkah berikut.

- Dapatkan binary mask hasil segmentasi
- Dapatkan binary mask dari *ground truth*
- Hitung nilai *dice coefficient* antara hasil segmentasi dan *ground truth*.

## 2.7 Confusion Matrix

*Confusion matrix* merupakan metode evaluasi yang membandingkan hasil prediksi model dengan nilai yang sebenarnya. *Confusion matrix* digunakan untuk mengukur performa model klasifikasi dan memperoleh informasi tentang jenis kesalahan yang terjadi, seperti *false positive* dan *false negative*. Penggambaran *confusion matrix* terdapat pada Tabel 2.1.

**Tabel 2.1** *Confusion Matrix*

		Prediksi	
		Positif	Negatif
Aktual	Positif	TP	FN
	Negatif	FP	TN

*Confusion matrix* pada Tabel 2.1 menghasilkan empat jenis nilai yang menggambarkan hasil performa model dalam melakukan tugas klasifikasi atau identifikasi. Nilai-nilai tersebut adalah *True Positive*, *True Negative*, *False Positif*, dan *False Negative*.

- *True Positive* (TP), yaitu nilai yang diperoleh ketika model dengan benar mengidentifikasi kelas positif
- *True Negative* (TN), yaitu nilai yang diperoleh ketika model dengan benar mengidentifikasi kelas negatif
- *False Positive* (FP), yaitu nilai yang diperoleh ketika model melakukan kesalahan dengan mengidentifikasi kelas negatif sebagai kelas positif

- *False Negative* (FN), yaitu nilai yang diperoleh ketika model melakukan kesalahan dengan mengidentifikasi kelas positif sebagai kelas negatif.

Keempat nilai yang sudah diperoleh dapat digunakan untuk menghitung nilai *performance matrix* lainnya pada *confussion matrix*. Adapun perhitungannya dapat dilihat pada penjelasan berikut.

### 2.7.1 Accuracy

*Accuracy* merupakan proses perhitungan nilai yang diperoleh dengan perbandingan nilai prediksi benar (TP dan TN) dengan total data keseluruhan, yang akan menghasilkan nilai akurasi model dalam tugas klasifikasi/identifikasi. Perhitungan dirumuskan pada Persamaan 2.7.

$$Akurasi = \frac{\text{Jumlah data dengan prediksi benar (TP + TN)}}{\text{Jumlah keseluruhan data}} \quad (2.7)$$

### 2.7.2 Precision

Perhitungan nilai *precision* diperoleh dari perbandingan hasil prediksi model yang sesuai data objek yang diinginkan (TP), dengan jumlah hasil prediksi model yang sesuai maupun tidak (TP dan FP). Perhitungan dirumuskan pada Persamaan 2.8.

$$Precision = \frac{TP}{TP + FP} \quad (2.8)$$

### 2.7.3 Recall

Perhitungan *recall* melakukan perbandingan hasil prediksi model yang sesuai data objek yang diinginkan (TP), dengan jumlah hasil prediksi model, baik yang terbaca dengan benar maupun tidak terbaca oleh model. Perhitungan dirumuskan melalui Persamaan 2.9.

$$Recall = \frac{TP}{TP + FN} \quad (2.9)$$

### 2.7.4 F-1 Score

*F-1 Score* adalah perhitungan nilai rata-rata harmonik dari *precision* dan *recall*, untuk menggambarkan keakuratan model dalam melakukan klasifikasi. Perhitungan dapat dirumuskan pada Persamaan 2.10.

$$F1 - Score = 2 \times \frac{\text{Recall} \times \text{Precision}}{\text{Recall} + \text{Precision}} \quad (2.10)$$

## 2.8 Penelitian Terdahulu

Potensi citra konjungtiva mata sebagai parameter untuk mendeteksi anemia telah diteliti menggunakan berbagai metode. Penelitian oleh Noor et al. (2019) membandingkan tiga algoritma klasifikasi yaitu SVM (*Support Vector Machine*), *Decision Tree*, dan KNN (*K-Nearest Neighbor*) untuk mengidentifikasi anemia menggunakan data latih sejumlah 81 citra dan data uji sejumlah 23 citra. Proses ekstraksi fitur-fitur penting pada gambar diperoleh dengan menggunakan modul pada MATLAB, dengan akurasi tertinggi, diperoleh oleh model *Decision Tree*, yaitu sebesar 82,61%. Penelitian lainnya oleh Asiyah et al. (2022) memanfaatkan algoritma PCA (*Principal Component Analysis*) untuk mengurangi dimensi pada citra dengan tetap mempertahankan fitur-fitur penting didalamnya, sehingga kompleksitas data dan *noise* pada data dapat berkurang, serta adanya efisiensi komputasi. Hasil pemrosesan PCA kemudian digunakan pada algoritma KNN untuk mengidentifikasi citra kedalam anemia atau non-anemia. Penelitian ini berhasil mengidentifikasi citra uji dengan akurasi sebesar 87,5%.

Penggunaan Metode HOG (*Histogram of Oriented Gradients*) juga digunakan untuk membantu metode *Decision Tree Classification* dalam mengidentifikasi citra konjungtiva mata. HOG berperan untuk mengekstraksi fitur-fitur penting pada gambar sebelum digunakan pada algoritma klasifikasi. *Logistic Regression* juga membantu algoritma klasifikasi untuk menyeimbangkan data citra yang digunakan. Penelitian berhasil mencapai akurasi 92.5%, dengan pemakaian *optimizer* SGD yang mempengaruhi nilai akurasi model (Salsabila et al., 2022). Pengujian model *transfer learning* dengan *pre-trained* model AlexNet juga dilakukan pada penelitian oleh Amalia et al. (2023). Penelitian memakai data citra konjungtiva mata yang mencakup nilai yang tidak seimbang untuk setiap kelasnya, sehingga digunakan teknik augmentasi untuk menyeimbangkan dan menambah variasi data pada kelas dengan jumlah yang lebih sedikit dari kelas lainnya. Hasil dari uji coba model ini memperoleh akurasi 85% pada identifikasi citra uji. Penelitian lainnya juga menggunakan dataset yang sama, namun dengan perlakuan yang berbeda. Penelitian ini melakukan *pre-processing* gambar terlebih dahulu dengan menggunakan *Gaussian Blur* untuk

mengurangi *noise* pada citra dengan mengaburkan detail-detail kecil dan menghaluskan citra. Hasil dari proses tersebut kemudian digunakan untuk proses ekstraksi fitur-fitur penting pada citra dengan CNN. Hasil ekstraksi fitur oleh CNN akan digunakan oleh metode *Logistic Regression* dalam menentukan kelas citra. Penelitian berhasil mencapai akurasi sebesar 92.5% (Asare et al., 2023).

Penelitian-penelitian terdahulu yang memanfaatkan metode pemrosesan gambar seperti *Gaussian Blur* dan PCA menghasilkan performa yang cukup baik. Penggunaan dataset yang sama pada penelitian yang menggunakan pemrosesan gambar diawal, yaitu penggunaan *Gaussian Blur*, menghasilkan akurasi yang lebih baik dari pada penelitian yang hanya menggunakan model *pre-trained* AlexNet. Selain penggunaan dataset dengan perlakuan yang berbeda, pengolahan gambar yang baik, seperti pengurangan *noise* pada gambar, dapat memberikan pengaruh besar terhadap performa model yang digunakan. Salah satu teknik pengolahan gambar yang dapat digunakan adalah teknik segmentasi. Penelitian ini menggunakan algoritma *K-Means Clustering* untuk segmentasi citra konjungtiva mata.

Penggunaan *K-Means Clustering* sering digunakan pada beberapa tugas segmentasi dan dikombinasikan untuk tugas klasifikasi gambar. Penelitian oleh Febrianti et al. (2018) menggabungkan algoritma *K-Means Clustering* dengan SVM untuk melakukan identifikasi api pada citra hutan dan bangunan. Peneliti memanfaatkan *K-Means Clustering* untuk melakukan segmentasi pada sejumlah citra latih untuk memperoleh objek yang diinginkan. Proses ini berfokus pada segmentasi warna api. Hasil segmentasi kemudian digunakan untuk proses identifikasi oleh SVM dan berhasil dengan akurasi 90%. Penggunaan *K-Means Clustering* pada proses segmentasi memberikan pengaruh yang baik pada performa SVM. Penelitian lainnya menggunakan *K-Means Clustering* untuk segmentasi warna citra kanker kulit. Hasil segmentasi membantu algoritma GLCM (*Gray Level Co-Occurrence Matrix*) untuk ekstraksi fitur hasil segmentasi sehingga dapat digunakan oleh MSVM (*Multiclass Support Vector Machine*) untuk klasifikasi citra kanker kulit. Penelitian berhasil dengan akurasi 96.25% (Monika et al., 2020). Algoritma *K-Means Clustering* juga digunakan pada penelitian oleh Mehmood et al. (2022). *K-Means Clustering* melakukan proses segmentasi pada 500 citra kulit vitiligo dan dihasilkan hasil segmentasi gambar yang baik.

Penggabungan *K-Means Clustering* dengan algoritma klasifikasi citra dapat menghasilkan performa yang baik, salah satunya adalah algoritma CNN. Algoritma CNN memiliki arsitektur jaringan saraf yang digunakan untuk pengenalan citra dan dapat dimodifikasi sesuai dengan kebutuhan tugas atau data yang digunakan. Arsitektur CNN dapat mempelajari fitur-fitur relevan pada gambar, sehingga cocok digunakan pada dataset kecil atau data dengan karakteristik yang lebih sederhana. Citra hasil segmentasi merupakan salah satu data citra yang memiliki karakteristik yang sederhana sehingga cocok digunakan pada metode CNN. Penelitian oleh Banik et al. (2020) memakai data BCCD (*Blood Cell Count and Detection*) dengan jumlah data yang sedikit yaitu 365 citra sel darah putih untuk diklasifikasikan kedalam jenis sel darah putih berdasarkan warna *nucleus* (inti sel) nya. Penelitian menggunakan metode *K-Means Clustering* untuk proses segmentasi warna *nucleus* dari sel darah putih sehingga menghasilkan citra objek yang dibutuhkan saja, yaitu *nuclues*. Hasil segmentasi akan digunakan oleh model arsitektur CNN yang dirancang sesuai tugas klasifikasi. Metode CNN berhasil melakukan klasifikasi dengan akurasi yang dicapai sebesar 97%.

**Tabel 2.2** Rangkuman Penelitian Terdahulu

No	Penulis	Judul	Metode	Keterangan
1	Amanda Febrianti	Penerapan Metode K-Means Clustering dan Support Vector Machine (SVM) dalam Identifikasi Api pada Citra Warna Digital	K-Means Clustering dan SVM	Penelitian melakukan segmentasi dengan <i>K-Means Clustering</i> . Hasil segmentasi citra api digunakan pada metode SVM sebagai <i>input</i> pada proses identifikasi citra api. Penelitian ini berhasil dengan nilai akurasi sebesar 90%.

2	Nahiyan Bin Noor, Md. Saeid Anwar, Mrinmoy Dey	Comparative Study Between Decision Tree, SVM and KNN to Predict Anaemic Condition	Decision Tree Classification	Penelitian ini melakukan perbandingan kinerja tiga metode klasifikasi yaitu: <i>Decision Tree</i> , SVM, dan KNN. <i>Decision Tree</i> berhasil memprediksi anemia melalui citra konjungtiva mata dengan akurasi tertinggi yaitu 82,61%.
3	M. Krishna Monika, N. Arun Vignesh, Ch. Usha Kumari, E. Laxmi Lydia	Skin Cancer Detection and Classification Using Machine Learning	K-Means Clustering dan GLCM + MSVM	Segmentasi citra kanker kulit dilakukan dengan <i>K-Means Clustering</i> . Hasil segmentasi citra kanker dijadikan <i>input</i> pada GLCM dan MSVM untuk klasifikasi citra. Hasil akurasi senilai 96.25%.
4	Engr. Nimra Mehmood, Prof. Dr. Engr. Saad Jawaid Khan, Prof. Dr. Engr. Munaf Rashid	K-Means Clustering Based Color Segmentation on Vitiligo Skin Lesion	K-Means Clustering	Penelitian ini bertujuan untuk melakukan tugas segmentasi gambar pada citra kulit vitiligo. Hasil penelitian menunjukkan <i>k-means clustering</i> dapat bekerja dengan baik pada tugas segmentasi citra kulit vitiligo.



5	Siti Asiyah, Iwan Iwut Tritoasmoro, Sofia Sa'idah	Anemia Detection Through Conjunctiva on Eyes Using PCA Method and K- Nearest Neighbor	PCA dan K-NN	Penelitian memakai PCA untuk reduksi dimensi pada citra dan KNN untuk proses identifikasi citra. Hasil yang diperoleh mencapai 87.5%.
6	Ghine Azzahra Salsabila, Sofia Sa'idah, Rustam	Anemia Detection Through Conjunctiva on Eyes Using Histogram of Oriented Gradients and Decision Tree Classification	HOG dan Decision Tree Classification + Logistic Regression	Penelitian menggunakan HOG untuk ekstraksi fitur-fitur penting pada citra. Hasil proses HOG digunakan pada <i>Decision Tree Classification</i> dan juga <i>Logistic Regression</i> untuk identifikasi anemia melalui citra konjungtiva mata. Penelitian berhasil mengidentifikasi citra uji dengan akurasi 92.5%.
7	Partha Pratim Banik, Rappy Saha, Ki-Doo Kim	An Automatic Nucleus Segmentation and CNN Model Based Classification Method of White Blood Cell	K-Means Clustering dan CNN	<i>K-Means Clustering</i> pada segmentasi citra nukleus sel darah putih akan mengelompokkan citra sesuai dengan jenisnya melalui warna nukleus. Hasilnya akan digunakan sebagai <i>input</i> pada model CNN sehingga diperoleh nilai akurasi sebesar 97%.

8	Elva Amalia, Mustari Lamada, Andi Baso Kaswar, Dyah Darma Andayani	Penerapan Algoritma Transfer Learning Pada Klasifikasi Penyakit Anemia Berdasarkan Citra Palpebral Konjungtiva	AlexNet	Penelitian menggunakan arsitektur model <i>pre- trained transfere learning</i> AlexNet dan berhasil mengidentifikasi data uji dengan akurasi 85%.
	Peter Appiahene, Enoch Justice Arthur, Stephen Korankye	Detection of Anemia Using Conjunctiva Images.		Penelitian menggunakan dataset citra konjungtiva mata yang diproses menggunakan <i>Gaussian</i> <i>Blur</i> dan diekstrak dengan <i>CNN</i> untuk dapat digunakan pada <i>Logistic</i> <i>Regression</i> . Penelitian dengan akurasi 92.5%.

## 2.9 Perbedaan Penelitian

Penelitian ini memiliki perbedaan yang mencakup pada objek penelitian, metode yang digunakan serta *output* yang dihasilkan. Perbedaan penelitian ini dengan penelitian yang dilakukan oleh Amanda, Krishna, Mehmood dan Partha ditemukan pada perbedaan objek yang digunakan. Ditemukan juga perbedaan metode dengan penelitian oleh Noor, Iwan, Ghine, Elva dan Peter, dimana pada penelitian ini digunakan metode *K-Means Clustering* dan *CNN*. Proses pengolahan dataset juga berbeda-beda, dimana penelitian ini memanfaatkan beberapa teknik pemrosesan citra seperti *image equalization* yang memanfaatkan fungsi *histogram equalization*, dan *image unsharp masking filter* yang memanfaatkan fungsi *gaussian blur*. Hasil penelitian juga memiliki *output* yang berbeda dengan penelitian lainnya, dimana penelitian ini akan menghasilkan sistem dalam bentuk *android* dengan hasil prediksi yang disertai dengan informasi cara penanganan dan pencegahan dasar penyakit.

## BAB 3

### ANALISIS DAN PERANCANGAN SISTEM

#### 3.1 Data yang digunakan

Penelitian menggunakan data citra konjungtiva mata pada penelitian oleh Dimauro et al. (2023) dan pengambilan citra secara langsung. Data latih dan validasi yang digunakan berjumlah 400 gambar, dengan masing-masing kelas untuk anemia dan non-anemia sebanyak 200 gambar. Data latih berjumlah 320 gambar dan data validasi berjumlah 80 gambar. Contoh data dapat dilihat pada Gambar 3.1.



**Gambar 3.1** Citra Konjungtiva Anemia dan Non-anemia

Untuk data pengujian, dilakukan pengambilan langsung citra konjungtiva mata dari klinik Bidan E.br Gultom, A. Md, S. Keb. di daerah Tarutung, Tapanuli Utara. Data pengujian yang terkumpul berjumlah 40 gambar, dengan masing-masing kelas anemia dan non-anemia berjumlah 20 gambar. Pengambilan data citra dilakukan dengan menyesuaikan cara pengambilan citra yang dilakukan pada data latih dan validasi.

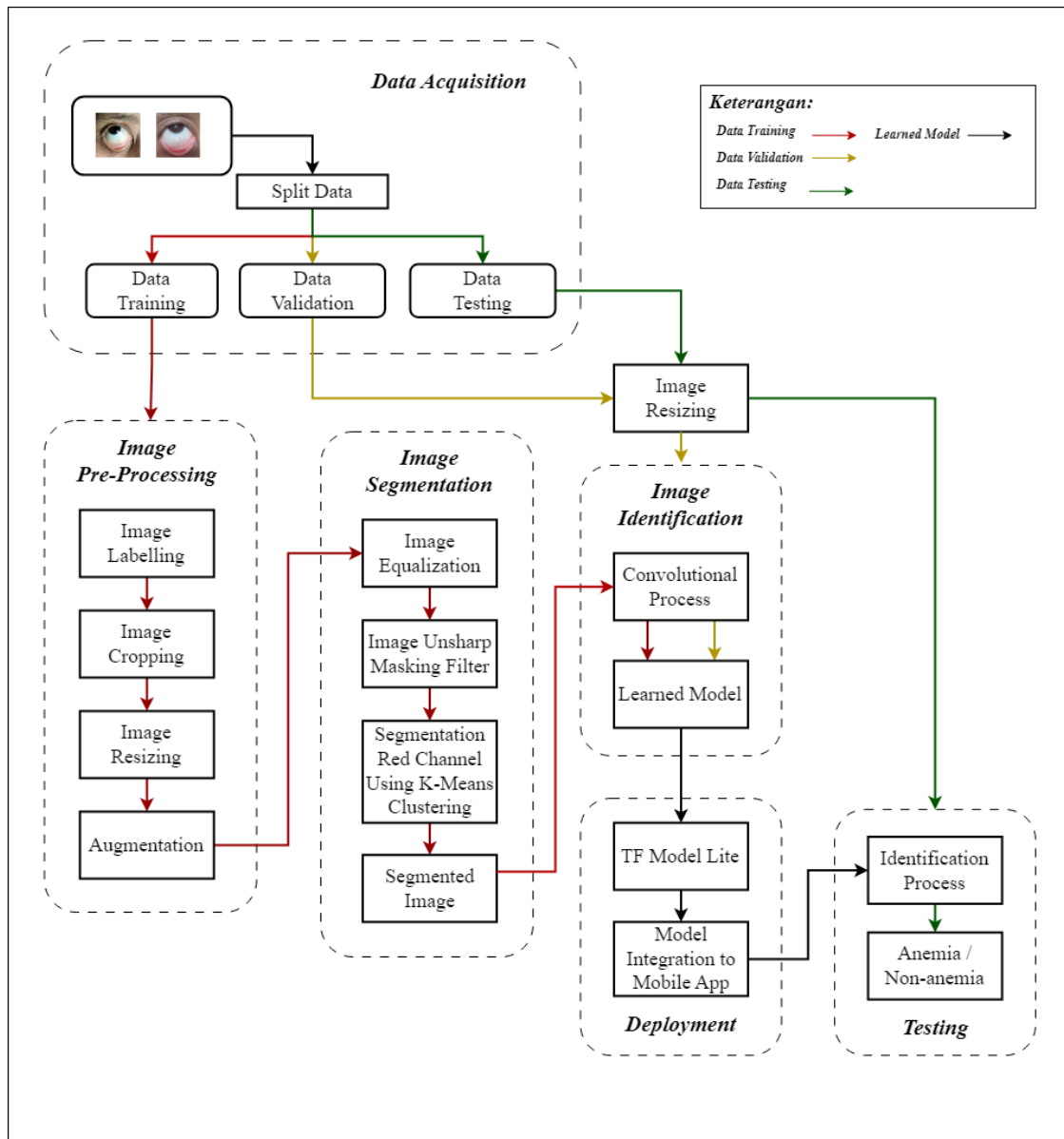
**Tabel 3.1** Pembagian Data

No	Data	Data Latih	Data Validasi	Data Uji	Jumlah
1	Citra konjungtiva anemia	160	40	20	220
2	Citra konjungtiva non-anemia	160	40	20	220
Total Data		320	80	40	440

#### 3.2 Analisis Sistem

Tahap pertama pada penelitian merupakan proses pengumpulan data citra konjungtiva mata anemia dan non-anemia. Data yang terkumpul akan dibagi kedalam tiga kelompok data, yaitu data pelatihan (*training*), data validasi (*validation*), dan data uji (*testing*).

Data latih akan memasuki tahap pra-pemrosesan data (*image pre-processing*). Proses-proses yang terjadi pada tahap ini mencakup proses *labelling* untuk pengelompokan kelas data kedalam anemia atau non-anemia, proses *cropping* untuk mengambil area konjungtiva mata yang dibutuhkan, proses *resizing* untuk mengubah ukuran piksel pada citra sehingga keseluruhan citra memiliki ukuran yang sama, dan proses *augmentation* untuk menambah variasi data. Setelah tahap pra-pemrosesan dilakukan, data latih akan memasuki tahap segmentasi (*image segmentation*). Sebelum citra disegmentasi dengan menggunakan *K-Means Clustering*, citra terlebih dahulu melewati dua proses penting, yaitu *image equalization* yang bertujuan untuk memperjelas perbedaan warna dan kontras pada citra, serta proses *image unsharp masking filter* yang bertujuan untuk meningkatkan ketajaman dan struktur detail citra. Hasil citra segmentasi akan digunakan sebagai data *input* untuk tahap selanjutnya, yaitu tahap identifikasi citra (*image identification*). Pada tahap ini terjadi proses konvolusi menggunakan metode CNN (*Convolutional Neural Netwok*) untuk mempelajari dan mengenali citra konjungtiva mata. Tahap ini akan menghasilkan model yang dilatih menggunakan data latih dan data validasi yang sebelumnya sudah melalui proses *resizing*. Model terbaik akan diubah kedalam format *tflite*, untuk diintegrasikan dengan sistem dalam bentuk *android*. Selanjutnya, model yang sudah tertanam didalam sistem *android* sudah dapat diuji dengan menggunakan data uji. Keseluruhan tahapan penelitian dirangkum pada Gambar 3.2.



**Gambar 3.2** Arsitektur Umum

### 3.2.1 Data acquisition

Pada tahap ini, peneliti mengumpulkan data citra konjungtiva mata dari Dimauro et al. (2023) dan pengambilan langsung dari klinik. Pengambilan citra dilakukan dengan menggunakan kamera *smartphone* 16 MP, dengan citra berekstensi *.jpg*. Data citra yang terkumpul akan dikelompokkan kedalam tiga kelompok data, yaitu data pelatihan (*data training*), data validasi (*data validation*), dan data uji (*data testing*).

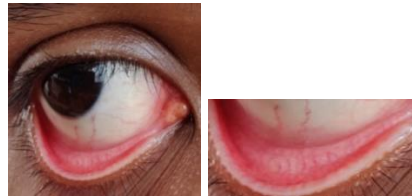
### 3.2.2 Image pre-processing

Pemrosesan gambar akan dilakukan sebelum dijadikan sebagai *input* data pada pemrosesan model. Tahap ini bertujuan untuk meningkatkan kualitas dan memperjelas

fokus objek yang digunakan pada citra, sehingga model dapat dilatih dengan baik. Terdapat tiga proses yang dilakukan, yaitu proses *cropping*, *resizing*, dan *labelling*.

#### 3.2.2.1 Cropping

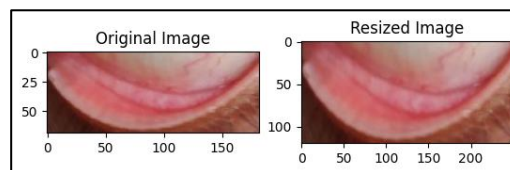
*Cropping* merupakan teknik yang digunakan dengan memotong ukuran gambar untuk memperoleh bagian yang dibutuhkan. Proses ini dilakukan dengan membuang area pada gambar yang tidak diinginkan atau diperlukan. Proses ini bertujuan untuk meningkatkan fokus pada objek atau area tertentu pada gambar, mengurangi gangguan atau elemen yang tidak relevan, dan memperbaiki komposisi visual gambar. Pada penelitian ini, proses *cropping* juga bertujuan untuk menyederhanakan area warna yang akan mengalami proses segmentasi. Contoh citra sebelum dan sesudah *cropping* terdapat pada Gambar 3.3.



**Gambar 3.3** Citra Sebelum dan Sesudah *Cropping*

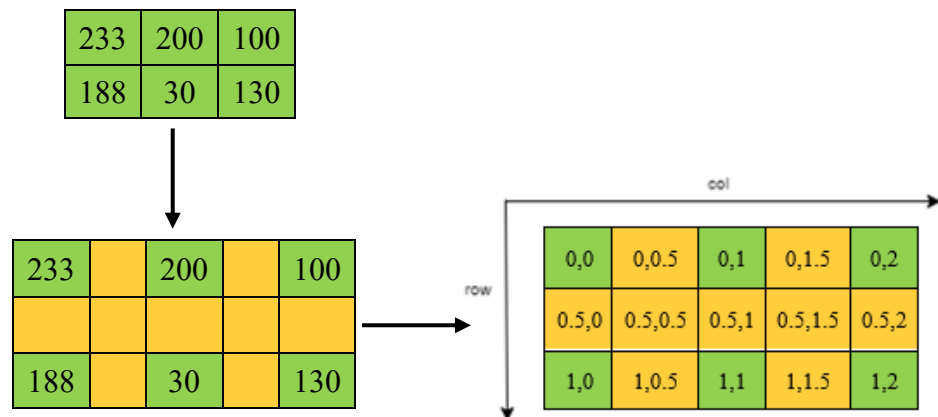
#### 3.2.2.2 Resizing

*Resizing* adalah proses mengubah ukuran gambar menjadi ukuran tertentu sebelum digunakan sebagai *input* pada model. Data citra yang telah di-*cropping* memiliki ukuran piksel dan ukuran file yang berbeda-beda. Oleh karena itu dibutuhkan proses *resizing* sehingga gambar yang digunakan kedalam sistem memiliki ukuran yang sama. Pada tahap ini digunakan teknik dasar untuk mengubah ukuran gambar menjadi 250 x 120 piksel. Pengubahan ukuran dilakukan dengan menyesuaikan skala gambar agar citra tidak terdistorsi karena aspek rasio gambar yang ikut berubah. Citra hasil *resizing* dapat dilihat pada Gambar 3.4



**Gambar 3.4** Citra Hasil *Resizing*

Tahap *resizing* dilakukan dengan menggunakan *library OpenCv* yang memiliki modul fungsi *resize* untuk mengubah ukuran gambar menjadi ukuran piksel yang diinginkan. Pada fungsi *resize*, digunakan metode *interpolation*, yaitu metode matematika untuk menemukan himpunan nilai baru pada suatu fungsi berdasarkan himpunan nilai sebelumnya pada fungsi yang sama. Adapun proses *resizing* dapat dilihat pada bagian berikut.



Citra dengan ukuran dimensi 2x3 piksel akan diubah ukurannya menjadi dimensi 3x5 piksel. Seluruh piksel dengan *background* warna kuning merupakan piksel tambahan yang dibentuk untuk menambah dimensi gambar. Nilai pada piksel tambahan merupakan perhitungan dari nilai dua piksel yang sudah diketahui. Untuk mempermudah proses perhitungan, setiap piksel akan diberikan nilai koordinat. Melalui nilai koordinat, diperoleh nilai jarak (bobot) antara piksel referensi dan piksel tambahan, yang dibutuhkan pada proses perhitungan. Adapun perhitungan yang dilakukan dapat dilihat pada rincian berikut.

Diketahui nilai referensi:

$$I(0,0) = 233, I(0,1) = 200, I(0,2) = 100, I(1,0) = 188, I(1,1) = 30, I(1,2) = 130$$

Bobot yang digunakan senilai 0.5. Maka, nilai piksel baru dapat dihitung sebagai berikut:

$$I(0,0.5) = I(0,0) \times (1-0.5) + I(0,1) \times 0.5$$

$$I(0,0.5) = 233 \times 0.5 + 200 \times 0.5$$

$$I(0,0.5) = 216$$

$$I(0,1.5) = I(0,1) \times (1-0.5) + I(0,2) \times 0.5$$

$$I(0,1.5) = 200 \times 0.5 + 100 \times 0.5$$

$$\mathbf{I(0,1.5) = 150}$$

$$I(0.5,0) = I(0,0) \times (1-0.5) + I(1,0) \times 0.5$$

$$I(0.5,0) = 233 \times 0.5 + 188 \times 0.5$$

$$\mathbf{I(0.5,0) = 210.5}$$

$$I(0.5,1) = I(0,1) \times (1-0.5) + I(1,1) \times 0.5$$

$$I(0.5,1) = 200 \times 0.5 + 30 \times 0.5$$

$$\mathbf{I(0.5,1) = 115}$$

$$I(0.5,2) = I(0,2) \times (1-0.5) + I(1,2) \times 0.5$$

$$I(0.5,2) = 100 \times 0.5 + 130 \times 0.5$$

$$\mathbf{I(0.5,2) = 115}$$

$$I(1,0.5) = I(1,0) \times (1-0.5) + I(1,1) \times 0.5$$

$$I(1,0.5) = 188 \times 0.5 + 30 \times 0.5$$

$$\mathbf{I(1,0.5) = 109}$$

$$I(1,1.5) = I(1,1) \times (1-0.5) + I(1,2) \times 0.5$$

$$I(1,1.5) = 30 \times 0.5 + 130 \times 0.5$$

$$\mathbf{I(1,1.5) = 80}$$

$$I(0.5,0.5) = I(0.5,0) \times (1-0.5) + I(0.5,1) \times 0.5$$

$$I(0.5,0.5) = 210.5 \times 0.5 + 115 \times 0.5$$

$$\mathbf{I(0.5,0.5) = 163}$$

$$I(0.5,1.5) = I(0.5,1) \times (1-0.5) + I(0.5,2) \times 0.5$$

$$I(0.5,1.5) = 115 \times 0.5 + 115 \times 0.5$$

$$\mathbf{I(0.5,1.5) = 115}$$



Nilai piksel baru yang merupakan hasil *resizing* dengan perhitungan diatas dapat dilihat pada bagian berikut.

233	216	200	150	100
210.5	163	115	115	115
188	109	30	80	130

Proses *resizing* gambar pada penelitian ini dapat diterapkan dengan menggunakan *pseudocode* berikut.

```

DEFINE OpenCV module
SET new size for height 250, width 120
COUNT original size using shape function
COUNT img scale_x by dividing new size width with original width
COUNT img scale_y by dividing new size height with original height
COUNT new_height by multiplying original height with scale_y
COUNT new_width by multiplying original width with scale_x
UPDATE img size with resize function from OpenCV module

```

Contoh dari langkah-langkah *resizing*, dirincikan pada bagian berikut.

- Menghitung ukuran awal gambar

$$W_{awal} = 25 \text{ piksel}$$

$$H_{awal} = 12 \text{ piksel}$$

- Menghitung faktor skala pada sumbu x dan y

$$Scale_x = \frac{W_{baru}}{W_{awal}} = \frac{250}{25} = 10$$

$$Scale_y = \frac{H_{baru}}{H_{awal}} = \frac{120}{12} = 10$$

- Menghitung ukuran baru

$$W_{baru} = W_{awal} \times Scale_x = 25 \times 10 = 250$$

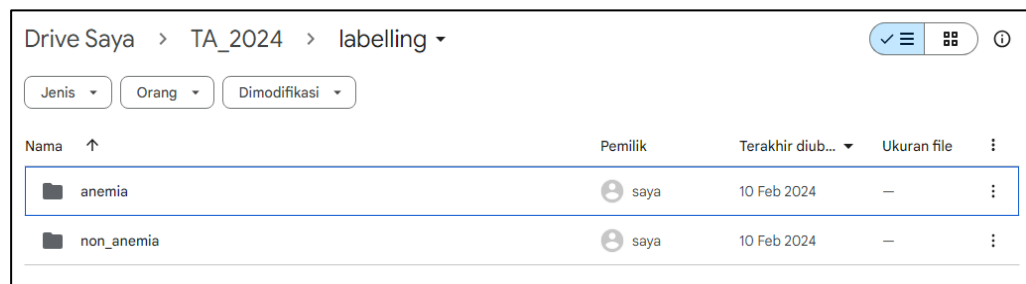
$$H_{baru} = H_{awal} \times Scale_y = 12 \times 10 = 120$$

- Menyimpan gambar dengan ukuran baru.

### 3.2.2.3 Labelling

*Labelling* merupakan proses membuat label pada data pelatihan. Proses ini bertujuan untuk mengelompokkan gambar kedalam folder pada *google drive* yang masing-masing sudah diberi nama sesuai dengan kelompok kelas citranya,

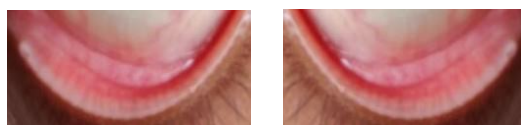
yaitu anemia dan non-anemia. Sistem akan mengakses gambar pada folder berdasarkan dengan labelnya masing-masing.



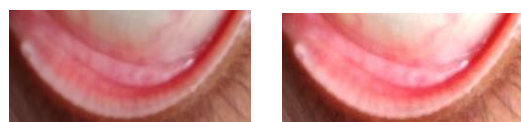
**Gambar 3.5** *Labelling*

#### 3.2.2.4 Augmentation

Augmentasi data merupakan teknik menambah variasi data melalui modifikasi atau manipulasi citra digital kedalam beberapa jenis tampilan, bentuk dan posisi yang berbeda. Teknik ini akan membantu model untuk dapat mengenali objek dalam berbagai tampilan, bentuk dan posisi. Variasi citra dapat menambah jumlah data yang digunakan. Untuk implementasi proses augmentasi pada citra, digunakan kelas *ImageDataGenerator* pada tensorflow yang memiliki beberapa parameter-parameter untuk mengubah tampilan, bentuk dan posisi citra. Penelitian ini menggunakan dua jenis augmentasi, yaitu *horizontal flip* dan *brightness range* dengan rentang pengubahan tingkat kecerahan citra berada diantara 0.8 hingga 1.3.



**Gambar 3.6** Citra Sebelum dan Sesudah *Horizontal Flip*



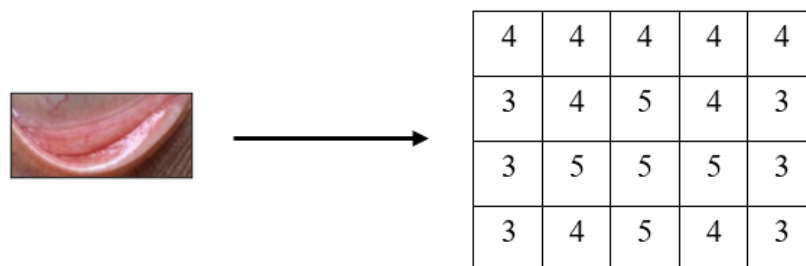
**Gambar 3.7** Citra Sebelum dan Sesudah *Brightness Range*

### 3.2.3 Image segmentation

Segmentasi citra adalah tahap mengelompokkan piksel pada citra digital ke dalam beberapa kelompok yang berbeda dengan ciri yang sama. Beberapa penelitian menggunakan metode *clustering* untuk segmentasi citra. *K-Means Clustering* merupakan salah satu algoritma *unsupervised* yang biasa digunakan untuk segmentasi area yang dibutuhkan dalam citra. Penelitian ini melakukan segmentasi citra dengan *K-Means Clustering* untuk mempermudah proses identifikasi citra konjungtiva mata. Namun, sebelum menerapkan algoritma *k-means*, terlebih dahulu digunakan teknik pemrosesan citra dengan *equalization* dan *unsharp masking filter* untuk memperjelas area citra yang dibutuhkan sebelum proses segmentasi. Sebelumnya, teknik ini sudah diujicobakan pada penelitian oleh Delgado et al. (2018) dan dinilai cukup efektif untuk membantu proses segmentasi citra konjungtiva mata secara akurat.

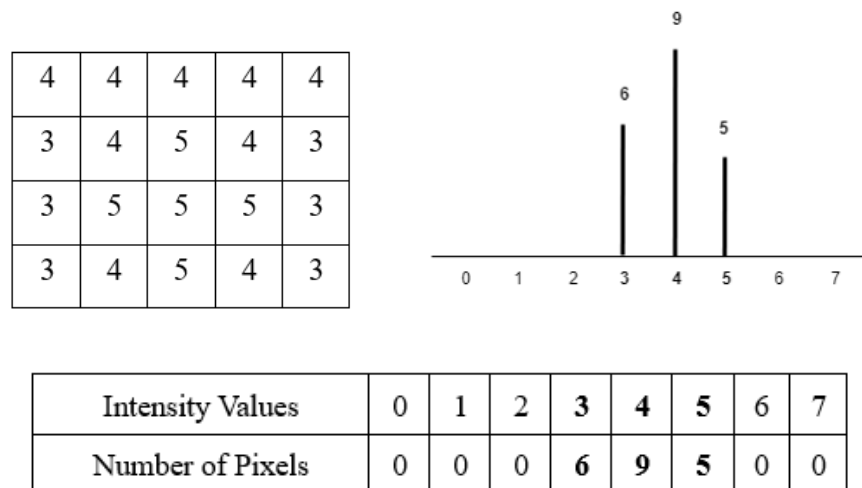
#### 3.2.3.1 Equalization

*Equalization* merupakan teknik pemrosesan gambar yang dirancang untuk memperjelas perbedaan warna dan kontras pada citra konjungtiva mata. Teknik ini menggunakan fungsi *histogram equalization* untuk meningkatkan distribusi intensitas piksel dalam citra sehingga seluruh rentang intensitas warna yang tersedia pada citra konjungtiva mata dapat digunakan secara optimal.



**Gambar 3.8** Citra Konjungtiva Mata dan Nilai Intensitas Pikselnya

Proses *equalization* melibatkan nilai Intensitas (*Intensity*) dari citra, sementara citra yang digunakan memiliki format RGB (*Red, Green, Blue*). Oleh karena itu, citra akan diubah terlebih dahulu kedalam format HSV (*Hue, Saturation, Value*) untuk memperoleh nilai intensitas pada citra. *Value* mengandung nilai intensitas yang dibutuhkan.



**Gambar 3.9** Perhitungan Variasi Intensitas Pada Piksel Citra

Gambar memiliki tiga variasi nilai intensitas yaitu 3, 4, dan 5 yang masing-masing berjumlah 6, 9, dan 5 piksel. Pada rentang 0-7, intensitas gambar hanya berada pada rentang 3-5. *Histogram equalization* bertujuan untuk mendistribusikan nilai intensitas pada citra, sehingga intensitas tidak hanya berada pada rentang 3-5. Perhitungan untuk memperoleh nilai intensitas yang lebih terdistribusi terdapat pada Tabel 3.2.

**Tabel 3.2** Perhitungan Nilai Intensitas Citra

Gray Level	Number of pixels (Np)	PDF (Np/Sum)	CDF	CDF x 7	Histogram Equalization Level
0	0	0	0	0	0
1	0	0	0	0	0
2	0	0	0	0	0
3	6	$6/20 = 0.3$	0.3	2.1	2
4	9	$9/20 = 0.45$	0.75	5.25	5
5	5	$5/20 = 0.25$	1	7	7
6	0	0	1	7	7
7	0	0	1	7	7

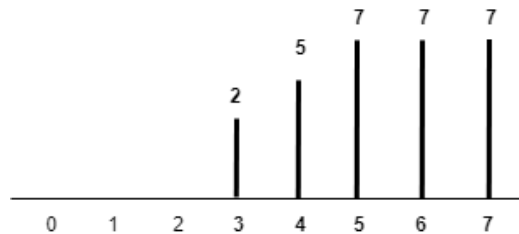
\*PDF = Probability Distribution Function

\*CDF = Cumulative Distribution Function

\*Np Sum = 20

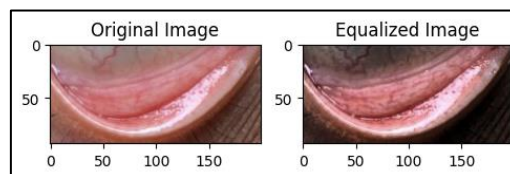
Perhitungan dilakukan dengan menggunakan jumlah *gray level* atau nilai intensitas citra untuk menghitung nilai PDF (*Probability Distribution Function*). PDF dapat dicari dengan membagi jumlah masing-masing *gray level* dengan total

jumlah *gray level*. Kemudian nilai PDF digunakan untuk menghitung nilai CDF (*Cumulative Distribution Function*). CDF dapat dicari dengan menambahkan nilai CDF awal (bernilai 0) secara bersilang dengan nilai PDF selanjutnya. Kemudian nilai CDF akan dikalikan dengan nilai maksimum *gray level* (bernilai 7). Hasilnya kemudian akan dibulatkan sebagai distribusi baru pada citra. Grafik distribusi yang baru dapat dilihat pada bagian dibawah ini.



Intensity Values	0	1	2	3	4	5	6	7
Number of Pixels	0	0	0	2	5	7	7	7

Pendistribusian nilai intensitas piksel dengan perhitungan fungsi *histogram equalization* akan mengurangi nilai intensitas piksel yang sudah ada, untuk didistribusikan ke area dengan nilai intensitas piksel yang lebih rendah. Hal ini mengakibatkan warna pada area citra yang awalnya memiliki nilai intensitas yang rendah, menjadi lebih gelap dari warna aslinya. Nilai intensitas piksel pada area konjungtiva mata memiliki nilai intensitas yang cukup tinggi, sehingga pengurangan nilai intensitas tidak terlalu mempengaruhi warna asli citra. Perubahan dari citra asli ke citra hasil *equalized image* dapat dilihat pada Gambar 3.10 berikut.



**Gambar 3.10** Perubahan Citra Asli ke Citra Hasil Equalization

Pemrosesan citra dengan teknik *equalization* menggunakan fungsi *histogram equalization* pada modul *OpenCv*, dapat juga diterapkan dengan menggunakan *pseudocode* berikut. Format gambar yang sebelumnya telah diubah dalam bentuk HSV akan kembali diubah kedalam format dalam bentuk RGB.

DEFINE OpenCv module

CONVERT img format to HSV by using `cvtColor` function from OpenCv module

APPLY equalization by using `equalizeHist` function from OpenCv module

CONVERT img format to BGR by using `cvtColor` function from OpenCv module

### 3.2.3.2 Unsharp masking filter

*Unsharp Masking Filter* merupakan salah satu teknik pengolahan gambar yang bertujuan untuk meningkatkan ketajaman tepi dan memperjelas struktur detail pada gambar. Teknik ini akan membantu proses segmentasi untuk menentukan area warna pada citra konjungtiva. Hasil citra proses *equalization* akan digunakan sebagai *input* untuk proses *unsharp masking filter*. Proses *unsharp masking filter* dijelaskan pada langkah-langkah berikut.

a. Langkah 1: Blur citra asli.

Citra asli akan diblur dengan menggunakan fungsi *gaussian blur*, untuk mengurangi *noise* pada citra dan menghasilkan citra yang halus dan kurang tajam. Berikut contoh perhitungan fungsi *gaussian blur* pada citra.

80	120	160
40	90	140
10	60	110

Filter *low-pass* (Gaussian Blur 3x3):

1	2	1
2	4	2
1	2	1

Perhitungan *gaussian blur*:

$$\text{Pixel (1,1)} = (1 \times 80 + 2 \times 120 + 1 \times 160 + 2 \times 40 + 4 \times 90 + 2 \times 140 + 1 \times 10 + 2 \times 60 + 1 \times 110) / 16 = 109$$

$$\text{Pixel (1,2)} = (1 \times 120 + 2 \times 160 + 1 \times 0 + 2 \times 90 + 4 \times 140 + 2 \times 0 + 1 \times 60 + 2 \times 110 + 1 \times 0) / 16 = 148$$

$$\text{Pixel (1,3)} = (1 \times 160 + 2 \times 0 + 1 \times 0 + 2 \times 140 + 4 \times 0 + 2 \times 0 + 1 \times 110 + 2 \times 0 + 1 \times 0) / 16 = 36$$

$$\text{Pixel (2,1)} = (1 \times 40 + 2 \times 90 + 1 \times 120 + 2 \times 10 + 4 \times 60 + 2 \times 110 + 1 \times 0 + 2 \times 0 + 1 \times 0) / 16 = 51$$

$$\text{Pixel (2,2)} = (1 \times 90 + 2 \times 140 + 1 \times 60 + 2 \times 120 + 4 \times 160 + 2 \times 0 + 1 \times 40 + 2 \times 110 + 1 \times 0) / 16 = 138$$

$$\text{Pixel (2,3)} = (1 \times 140 + 2 \times 0 + 1 \times 0 + 2 \times 110 + 4 \times 0 + 2 \times 0 + 1 \times 0 + 2 \times 0 + 1 \times 0) / 16 = 28$$

$$\text{Pixel (3,1)} = (1 \times 10 + 2 \times 60 + 1 \times 110 + 2 \times 0 + 4 \times 40 + 2 \times 90 + 1 \times 0 + 2 \times 140 + 1 \times 0) / 16 = 38$$

$$\text{Pixel (3,2)} = (1 \times 60 + 2 \times 110 + 1 \times 0 + 2 \times 90 + 4 \times 140 + 2 \times 0 + 1 \times 0 + 2 \times 0 + 1 \times 0) / 16 = 38$$

$$\text{Pixel (3,3)} = (1 \times 110 + 2 \times 0 + 1 \times 0 + 2 \times 160 + 4 \times 0 + 2 \times 0 + 1 \times 0 + 2 \times 0 + 1 \times 0) / 16 = 32$$

b. Langkah 2: Buat mask

Mask merupakan citra hasil pengurangan nilai piksel pada citra asli dengan nilai piksel pada citra *blur*, sehingga dihasilkan citra yang berisi detail-detail tepi yang dihapus pada proses *gaussian blur*.

c. Langkah 3: Tambahkan mask ke citra asli

Hasil mask kemudian akan ditambahkan kembali ke citra asli, sehingga menghasilkan citra asli dengan detail tepi yang lebih jelas. Perhitungannya dijelaskan pada bagian berikut.

Perhitungan untuk hasil *unsharp masking*:

$$\text{Pixel (1,1)} = 80 + (80 - 109) = 51$$

$$\text{Pixel (1,2)} = 120 + (120 - 148) = 92$$

$$\text{Pixel (1,3)} = 160 + (160 - 36) = 284$$

$$\text{Pixel (2,1)} = 40 + (40 - 51) = 29$$

$$\text{Pixel (2,2)} = 90 + (90 - 138) = 42$$

$$\text{Pixel (2,3)} = 140 + (140 - 28) = 252$$

$$\text{Pixel (3,1)} = 10 + (10 - 38) = -18$$

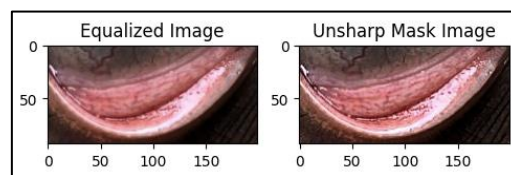
$$\text{Pixel (3,2)} = 60 + (60 - 38) = 82$$

$$\text{Pixel (3,3)} = 110 + (110 - 32) = 188$$

Melalui perhitungan yang dilakukan, maka diperoleh hasil piksel dengan teknik *unsharp masking filter* sebagai berikut.

51	92	284
29	42	252
-18	82	188

Penggunaan fungsi *gaussian blur* akan menghasilkan citra blur sehingga seluruh citra akan memiliki detail-detail citra yang halus. Pada citra ini, detail tepi pada citra tidak lagi terlihat. Untuk memperoleh detail tepi pada citra, maka digunakan teknik penambahan piksel citra hasil *blur* dengan citra aslinya, sehingga dihasilkan mask dengan detail tepi citra yang lebih jelas. Untuk hasil akhirnya, citra mask akan ditambahkan dengan citra asli piksel untuk menampilkan citra hasil *unsharp masking filter* dengan detail tepi yang lebih tajam. Perubahan citra *equalization* ke citra *unsharp masking filter* dapat dilihat pada Gambar 3.11 berikut.



**Gambar 3.11** Perubahan Citra Equalization ke Hasil Unsharp Masking Filter

Teknik pengolahan citra dengan *unsharp masking filter* dapat diterapkan dengan menggunakan *pseudocode* berikut.

```

DEFINE OpenCv module
BLUR img using GaussianBlur function from OpenCv module
COMBINE blur img and original img using addWeighted function from OpenCv module

```

Setelah proses pengolahan citra dengan teknik *equalization* dan *unsharp masking* dilakukan, maka citra dapat disegmentasi. Keberhasilan algoritma *K-Means Clustering* untuk melakukan segmentasi citra ditentukan oleh nilai klusternya (*k*). Penelitian ini menggunakan metode *Elbow* dalam menentukan nilai *k* yang tepat pada segmentasi citra yang digunakan. Metode *Elbow* adalah metode penentuan nilai kluster terbaik melalui analisa penurunan nilai variasi data terhadap rentang nilai kluster yang



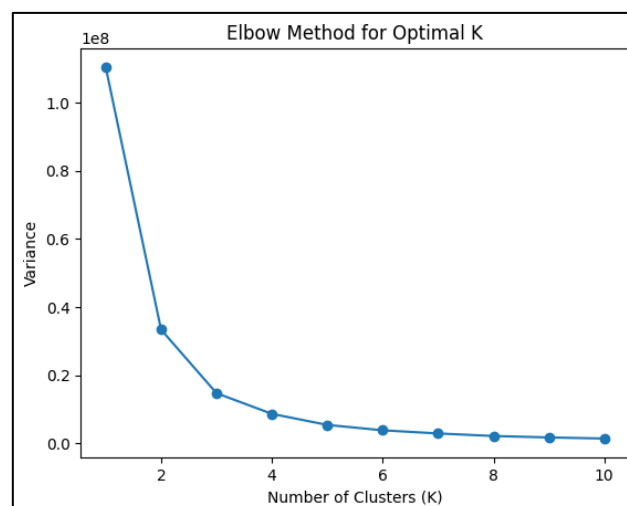
diujikan (Arifah et al., 2023). Pada metode ini, dilakukan uji coba segmentasi dengan jumlah kluster yang berbeda untuk dapat memperoleh nilai variasi datanya. Penerapan metode *Elbow* dapat menggunakan *pseudocode* berikut.

```

DEFINE OpenCv module
DEFINE variable variances
DEFINE variable k start from 1
CREATE calculate_variance function to calculate the value of the variances
CREATE k-means clustering function with k-means criteria
APPLY calculate_variance function to calculate set of data variances in each
cluster formed in k-means clustering function.
LOOP FOR EACH k in range 1 to 10 to collect variances value for each cluster

```

Percobaan dilakukan dengan menggunakan nilai kluster 1 hingga 10. Segmentasi dengan percobaan beberapa kluster tersebut akan menghasilkan nilai variasi data yang dapat dianalisa melalui grafik *Elbow*. Penentuan kluster terbaik dapat dilihat dari penurunan nilai variasi data yang signifikan antara satu kluster dengan kluster lain, yang membentuk sudut siku (*elbow*) pada grafik (Muningsih & Kiswati, 2018). Dari grafik *Elbow* diperoleh nilai kluster 2 yang menghasilkan penurunan nilai variasi data yang signifikan dan membentuk siku (*elbow*) pada grafik. Melalui analisa tersebut, maka penelitian ini akan menggunakan nilai  $k = 2$  untuk proses segmentasi citra. Grafik *Elbow* dari hasil percobaan yang dilakukan terdapat pada Gambar 3.9.



**Gambar 3.12** Grafik *Elbow*

Penggunaan nilai  $k = 2$  akan menghasilkan citra segmentasi dengan menggunakan metrik *Euclidean*. Berikut adalah contoh perhitungan manual *K-Means Clustering* untuk data citra 4x5 piksel menggunakan metrik *Euclidean*.

a. Langkah 1: Inisialisasi centroid awal

Pilih secara acak 2 centroid awal dari data input:

Centroid 1: [20,30,40,50]

Centroid 2: [25,35,45,55]

b. Langkah 2: Menghitung jarak *Euclidean*

Hitung jarak *Euclidean* antara setiap data dengan centroid yang telah dipilih:

Data 1: [20,30,40,50]

$$\text{Jarak centroid 1: } \sqrt{(20 - 20)^2 + (30 - 30)^2 + (40 - 40)^2 + (50 - 50)^2} = 0$$

$$\text{Jarak centroid 2: } \sqrt{(20 - 25)^2 + (30 - 35)^2 + (40 - 45)^2 + (50 - 55)^2} = 10$$

Data 2: [25,35,45,55]

$$\text{Jarak centroid 1: } \sqrt{(25 - 20)^2 + (35 - 30)^2 + (45 - 40)^2 + (55 - 50)^2} = 10$$

$$\text{Jarak centroid 2: } \sqrt{(25 - 25)^2 + (35 - 35)^2 + (45 - 45)^2 + (55 - 55)^2} = 0$$

Data 3: [22,32,42,52]

$$\text{Jarak centroid 1: } \sqrt{(22 - 20)^2 + (32 - 30)^2 + (42 - 40)^2 + (52 - 50)^2} = 4$$

$$\text{Jarak centroid 2: } \sqrt{(22 - 25)^2 + (32 - 35)^2 + (42 - 45)^2 + (52 - 55)^2} = 6$$

Data 4: [21,31,41,51]

$$\text{Jarak centroid 1: } \sqrt{(21 - 20)^2 + (31 - 30)^2 + (41 - 40)^2 + (51 - 50)^2} = 2$$

$$\text{Jarak centroid 2: } \sqrt{(21 - 25)^2 + (31 - 35)^2 + (41 - 45)^2 + (51 - 55)^2} = 8$$

Data 5: [24,34,44,55]

$$\text{Jarak centroid 1: } \sqrt{(24 - 20)^2 + (34 - 30)^2 + (44 - 40)^2 + (54 - 50)^2} = 8$$

$$\text{Jarak centroid 2: } \sqrt{(24 - 25)^2 + (34 - 35)^2 + (44 - 45)^2 + (54 - 55)^2} = 2$$

c. Langkah 3: Assign data ke kluster terdekat

Assign setiap data ke kluster terdekat berdasarkan jarak yang dihitung sebelumnya:

Data 1: Assign ke Kluster 1 (Centroid 1)

Data 2: Assign ke Kluster 2 (Centroid 2)

Data 3: Assign ke Kluster 1 (Centroid 1)

Data 4: Assign ke Kluster 1 (Centroid 1)

Data 5: Assign ke Kluster 2 (Centroid 2)

d. Langkah 4: Menghitung ulang centroid baru

Hitung rata-rata dari setiap kluster untuk mendapatkan centroid baru:

Kluster 1:

Data 1: [20,30,40,50]

Data 3: [22,32,42,52]

Data 4: [21,31,41,51]

Nilai rata-rata:  $[(20+22+21)/3, (30+32+31)/3, (40+42+41)/3, (50+52+51)/3] = [21,31,41,51]$ .

Centroid baru untuk Kluster 1: [21,31,41,51]

Kluster 2:

Data 2: [25,35,45,55]

Data 5: [24,34,44,54]

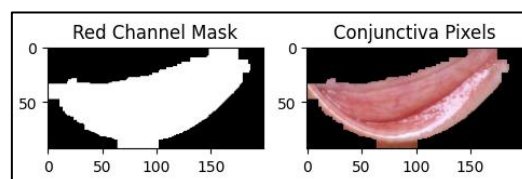
Nilai rata-rata:  $[(25+24)/2, (35+34)/2, (45+44)/2, (55+54)/2] = [24.5, 34.5, 44.5, 54.5]$ .

Centroid baru untuk Kluster 2: [24.5,34.5,44.5,54.5]

e. Langkah 5: Ulangi langkah 2 – 4 sampai konvergen

Ulangi langkah 2-4 dengan menggunakan centroid baru yang dihasilkan sebagai centroid awal, hingga tidak ada perubahan kluster atau konvergen.

Segmentasi yang dilakukan pada citra, akan mengelompokkan piksel pada citra kedalam dua kluster, sehingga ditampilkan citra biner yang memiliki warna putih dan hitam. Piksel pada area konjungtiva mata kemudian akan dikembalikan sehingga dihasilkan citra hasil segmentasi yang dapat dilihat pada Gambar 3.13.



**Gambar 3.13** Citra Hasil Segmentasi

Proses segmentasi dapat diterapkan dengan menggunakan *pseudocode* berikut.

```

DEFINE OpenCV module
DEFINE red_channel to stores image red channel
CHANGE red_channel dimension to be one dimensional matrix using reshape function
CHANGE red_channel matrix type to be float
DEFINE k-means criteria using OpenCv functions
APPLY k-means using kmeans function from OpenCv module
CONVERT centers value type into uint8
DEFINE segmented_red_channel to stores segmented red_channel
CHANGE segmented_red_channel shape to get red channel using reshape function
DEFINE binary_mask with Otsu method using OpenCv module
APPLY morphologyEx function from OpenCv module into binary_mask

```

Hasil dari proses segmentasi kemudian akan dievaluasi dengan menggunakan metrik penilaian *Dice coefficient*. Metrik penilaian ini akan mengukur kesamaan antara dua set data yang dalam konteks segmentasi citra adalah citra hasil segmentasi prediksi dan *ground truth*. *Dice coefficient* memiliki relevansi yang cocok untuk hasil segmentasi berupa *binary segmentation*. Metrik ini juga memberikan penilaian yang sensitif terhadap kesamaan antara hasil segmentasi dan *ground truth*, sehingga memberikan gambaran yang jelas untuk kualitas segmentasi citra. Proses penggunaan metode dapat dilihat pada pseudocode berikut.

```

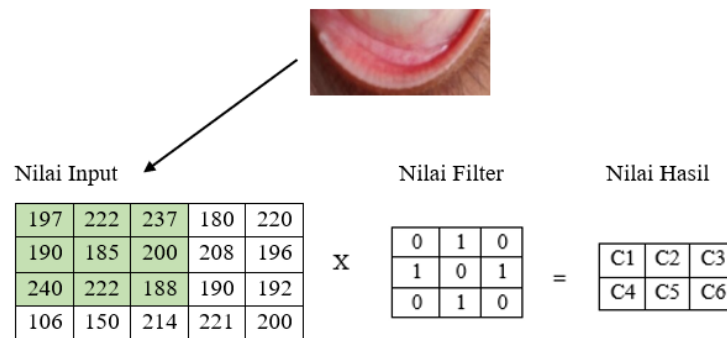
DEFINE Numpy module
DEFINE dice coefficient function
DEFINE segmented red channel as the segmented mask result from segmentation
DEFINE ground truth mask as the ground truth mask
DEFINE dice score as the result of dice coefficient function applied to segmented
red channel and ground truth mask
PRINT dice score.

```

### 3.2.4 Image identification dengan Convolutional Neural Network

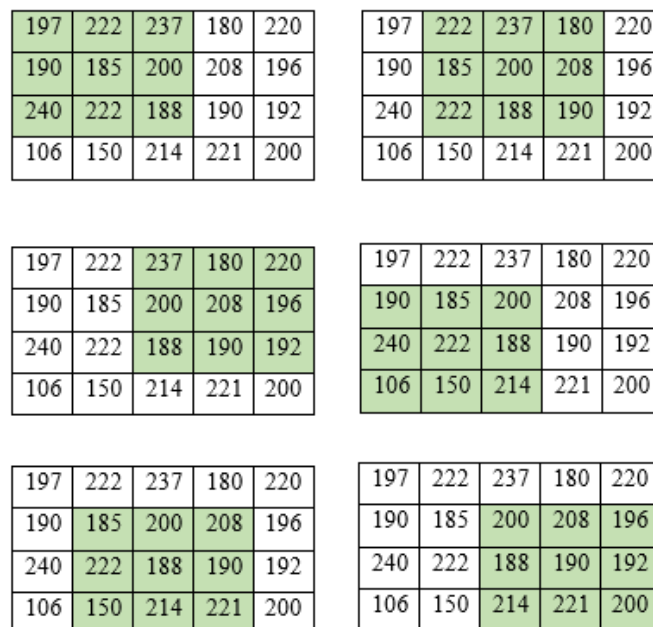
Identifikasi citra konjungtiva mata pada penelitian ini memakai algoritma *Convolutional Neural Network* (CNN). Proses identifikasi terjadi dalam dua proses, yaitu proses pada *feature extraction layer* dan *fully connected layer*. Terdapat dua jenis lapisan pada *feature extraction layer*, yaitu *convolutional layer* dan *pooling layer*. Pada *convolutional layer* terjadi proses konvolusi yang berfungsi untuk mengekstrak fitur-fitur penting pada citra. Proses konvolusi memanfaatkan kernel atau filter dengan ukuran tertentu untuk mendapatkan informasi sebagai representasi fitur penting pada citra yang digunakan pada proses identifikasi. Pada penelitian ini, digunakan empat

*convolutional layer* dengan masing-masing *layer* memiliki jumlah filter sebanyak 32, 64, 128, dan 256 berukuran 3 x 3. Masing-masing filter pada setiap lapisan akan diterapkan pada bagian citra untuk memperoleh *feature map* sebagai hasil fitur yang diekstraksi dan menjadi *input* untuk lapisan dan proses konvolusi berikutnya. Proses konvolusi digambarkan pada Gambar 3.14.



**Gambar 3.14** Proses Perhitungan Filter *Convolutional Layer*

Pada penerapan filter terhadap nilai *input* yang merupakan bagian dari citra, akan dilakukan pergeseran filter secara berurutan dari kiri atas matriks input dengan *stride* sebesar 1. *Stride* merupakan jumlah langkah yang digunakan filter untuk berpindah posisi. Proses pergeseran filter digambarkan pada Gambar 3.15.



**Gambar 3.15** Pergeseran Filter Pada Proses Konvolusi

Proses perhitungan antara nilai piksel *input* dengan nilai filter merupakan proses perkalian (*dot product*) yang akan menghasilkan nilai-nilai untuk *feature map*. Perhitungan pada proses konvolusi dapat dilihat pada rincian berikut.

$$C1 = (197 \times 0) + (222 \times 1) + (237 \times 0) + (190 \times 1) + (185 \times 0) + (200 \times 1) + (240 \times 0) + (222 \times 1) + (188 \times 0) = 222 + 190 + 200 + 222 = \mathbf{834}$$

$$C2 = (222 \times 0) + (237 \times 1) + (180 \times 0) + (185 \times 1) + (200 \times 0) + (208 \times 1) + (222 \times 0) + (188 \times 1) + (190 \times 0) = 237 + 185 + 208 + 188 = \mathbf{818}$$

$$C3 = (237 \times 0) + (180 \times 1) + (220 \times 0) + (200 \times 1) + (108 \times 0) + (196 \times 1) + (188 \times 0) + (190 \times 1) + (192 \times 0) = 180 + 200 + 196 + 190 = \mathbf{766}$$

$$C4 = (190 \times 0) + (185 \times 1) + (200 \times 0) + (240 \times 1) + (222 \times 0) + (188 \times 1) + (106 \times 0) + (150 \times 1) + (214 \times 0) = 185 + 240 + 188 + 150 = \mathbf{763}$$

$$C5 = (185 \times 0) + (200 \times 1) + (208 \times 0) + (222 \times 1) + (188 \times 0) + (190 \times 1) + (150 \times 0) + (214 \times 1) + (221 \times 0) = 200 + 222 + 190 + 214 = \mathbf{826}$$

$$C6 = (200 \times 0) + (208 \times 1) + (196 \times 0) + (188 \times 1) + (190 \times 0) + (192 \times 1) + (214 \times 0) + (221 \times 1) + (200 \times 0) = 208 + 188 + 192 + 221 = \mathbf{809}$$

Dengan perhitungan diatas, maka diperoleh hasil matriks *feature map* dari citra contoh sebagai berikut.

834	818	766
763	826	809

*Output* yang dihasilkan pada konvolusi pertama merupakan representasi fitur non-linearitas yang kompleks sehingga diperlukan fungsi aktivasi *ReLU* untuk memungkinkan model dapat memahami dan merepresentasikan pola-pola non-linear tersebut. Selanjutnya *feature map* akan digunakan sebagai *input* pada lapisan *pooling* dengan tujuan untuk mengurangi dimensi spasial dari fitur yang telah diekstrak dengan tetap mempertahankan informasi penting didalamnya. Pada penelitian ini digunakan *MaxPooling* dengan ukuran 2 x 2, untuk mengurangi dimensi pada fitur dengan mengambil nilai maksimum pada setiap jendela *pooling*. Hasil dari lapisan *pooling*

kemudian akan digunakan sebagai *input* untuk *convolutional layer* atau lapisan konvolusi berikutnya. Pada penelitian ini, dilakukan empat kali proses konvolusi, sesuai dengan jumlah lapisan konvolusi yang digunakan. Hasil akhir dari keempat konvolusi digunakan sebagai data masukan pada proses kedua, yaitu *fully connected layer*.

Lapisan-lapisan pada *fully connected layer* mengandung *neuron* aktivitas yang saling terhubung antara lapisan sebelumnya dengan lapisan berikutnya. *Feature map* yang dihasilkan pada proses konvolusi memiliki bentuk multidimensional array sehingga diperlukan “*flatten*” atau pengubahan *feature map* menjadi bentuk vektor satu dimensi yang digunakan sebagai *input* pada proses yang terjadi dalam *layer*. Pada penelitian ini, terdapat dua buah lapisan dengan masing-masing lapisan terdiri atas 256 *neuron* dan 1 *neuron*, serta penambahan lapisan *dropout* dengan nilai sebesar 0.3. Setiap *neuron* pada lapisan saling terhubung dengan lapisan sebelumnya. Lapisan pertama beserta fungsi aktivasi *ReLU* berfungsi untuk melakukan prediksi, sementara untuk lapisan terakhir merupakan lapisan output yang berfungsi untuk menghasilkan hasil prediksi model. Lapisan *dropout* bertujuan untuk mengurangi kemungkinan model mengalami *overfitting*. Lapisan *output* memakai fungsi aktivasi *sigmoid* untuk menghasilkan *output* yang berada pada rentang 0 dan 1 yang disesuaikan dengan tugas identifikasi pada penelitian.

#### 3.2.4.1 Training process

Model yang dibangun membutuhkan pelatihan untuk dapat mengenali citra yang akan diidentifikasi. Pada penelitian ini dilakukan pelatihan model pada 320 citra latih dan 80 citra validasi, dengan ukuran citra 250 x 120 piksel. Pada proses sebelumnya, data latih akan mengalami proses augmentasi untuk menambah jumlah data dengan variasi yang lebih banyak. Untuk melakukan proses tersebut, digunakan *ImageDataGenerator* sebagai sebuah kelas dalam pustaka *tensorflow* untuk melakukan augmentasi data citra sebelum digunakan pada pelatihan model. Terdapat beberapa argumen pada kelas *ImageDataGenerator* untuk konfigurasi data citra, yaitu:

- *Rescale*, merupakan argumen yang digunakan untuk menormalisasikan data gambar dengan membagi setiap nilai piksel pada citra dengan nilai 255, sehingga menghasilkan nilai piksel antara 0 dan 1. Penggunaan

argumen ini bertujuan untuk menjaga stabilitas pelatihan dan membantu model untuk dapat mencapai konvergensi lebih cepat.

- *Horizontal flip*, merupakan argumen untuk proses augmentasi yang memungkinkan gambar untuk diputar secara horizontal. Penggunaan argumen ini memberikan kemampuan pada model untuk mengenali citra latih dalam posisi yang berbeda secara horizontal.
- *Brightness range*, merupakan argumen yang menentukan rentang kecerahan pada citra secara acak, dengan nilai rentang kecerahan dimulai dari 0.8 hingga pada 1.3. Nilai rentang yang diberikan tidak mempengaruhi perubahan warna citra secara signifikan. Penggunaan argumen ini akan menambah pengenalan model terhadap variasi rentang kecerahan gambar.

Implementasi argumen-argumen dari kelas *ImageDataGenerator* akan disimpan kedalam variabel *train\_generator* untuk data latih dan *val\_generator* untuk data validasi, dengan masing-masing nilai *batch size* sama dengan 64. *Batch size* merupakan jumlah sampel data yang dilatih dalam satu iterasi (*epoch*). Artinya, pada setiap iterasi, terdapat 64 citra yang diambil secara acak untuk diproses sebelum bobot model diperbaharui. Nilai *class mode* yang digunakan adalah “*binary*” sebagaimana kasus pada penelitian ini merupakan kasus klasifikasi biner (identifikasi).

Data yang sudah diproses dan siap dilatihkan pada model, akan mengalami proses *compiling* dengan menggunakan *tensorflow*. Terdapat tiga parameter penting pada proses ini, yaitu parameter *loss*, *optimizer*, dan *metrics*, dengan rincian sebagai berikut.

- *Loss*, merupakan fungsi kerugian yang mengukur baik buruknya hasil prediksi model. Fungsi ini bertujuan untuk mengarahkan model untuk memperoleh nilai yang mendekati nilai yang sebenarnya. Pada penelitian, digunakan *binary cross-entropy*, yang merupakan salah satu jenis fungsi kerugian untuk kasus identifikasi.
- *Optimizer*, merupakan algoritma yang membantu model untuk dapat memperbaharui bobotnya sehingga dapat memprediksi hasil dengan lebih baik. Perubahan yang dilakukan mengacu pada kesalahan yang diperoleh dari fungsi *loss*. Fungsi optimasi menggunakan *Adam* dengan nilai



parameter *learning rate* sama dengan 0.001 ( $1e-3$ ). *Learning rate* merupakan parameter pada fungsi optimasi yang digunakan untuk menentukan jumlah langkah yang diambil *optimizer* ketika memperbaharui bobot model berdasarkan gradien fungsi kerugian.

- *Metrics*, merupakan parameter evaluasi yang memberikan informasi tambahan tentang kinerja model. Pada penelitian ini, digunakan metrik akurasi untuk menampilkan nilai prediksi benar yang diperoleh dari pelatihan model.

Pada proses pelatihan model, data citra akan menjadi *input* pada jaringan model yang dibangun untuk menghasilkan *output* berupa hasil prediksi model. Tahap ini disebut sebagai *forward propagation*, dimana data citra akan melewati setiap lapisan pada jaringan untuk diproses dan menghasilkan *output* yang digunakan sebagai *input* untuk lapisan berikutnya. Ketika setiap *output* diperoleh, hasil prediksi akan dibandingkan dengan label yang sebenarnya (*ground truth*) untuk menghitung seberapa jauh prediksi tersebut dari nilai yang sebenarnya. Tahap ini dilakukan dengan fungsi kerugian (*loss function*) yang pada penelitian ini memanfaatkan *binary cross-entropy*. Nilai kerugian tersebut akan diperbaharui sampai model mencapai titik konvergensinya, dengan mengubah bobot-bobot dalam jaringan. Pengubahan bobot melibatkan perhitungan gradien dari fungsi kerugian terhadap bobot dalam jaringan. Gradien akan memberikan informasi tentang seberapa banyak dan dalam arah mana model harus mengubah setiap bobot agar mengurangi nilai kerugian. Oleh karena itu, setiap lapisan akan mengirimkan gradien kesalahan ke lapisan sebelumnya, dimana setiap lapisan akan mengalikan gradien tersebut dengan turunan dari fungsi aktivasi yang digunakan pada lapisan tersebut. Proses tersebut, dinamakan sebagai *backpropagation* dimana proses akan berjalan mundur dari *output* kepada *input*. Pembaharuan bobot akan dilakukan oleh *optimizer*, dalam penelitian ini menggunakan *Adam*. Proses tersebut akan diulang sesuai dengan jumlah iterasi (*epoch*) yang dideklarasikan. Pada penelitian ini digunakan iterasi sebesar 35 *epoch*. Setiap iterasi akan membawa model perlahan-lahan ke titik konvergensi dan memberikan prediksi yang lebih akurat.

#### 3.2.4.2 *Learned model*

*Learned model* merupakan model hasil pembelajaran yang telah melalui proses pelatihan yang berfungsi untuk melakukan prediksi citra yang belum pernah dilihat oleh model melalui sistem, pada penelitian ini disebutkan sebagai data uji. Model dengan performa terbaik akan disimpan kedalam format *h5* dan akan dikonversi kedalam format *tflite* agar dapat diintegrasikan pada lingkungan android

Proses pelatihan model menggunakan algoritma CNN dapat diterapkan dengan menggunakan *pseudocode* berikut ini.

```
IMPORT libraries
```

```
DEFINE train_datagen as ImageDataGenerator with rescaling, horizontal_flip,
brightness_range augmentation parameters.
```

```
DEFINE validation_datagen as ImageDataGenerator with rescaling augmentation
parameter.
```

```
DEFINE checkpoint_callback using ModelCheckpoint function to automatically
saving the best model.
```

```
DEFINE early_stopping_callback using EarlyStopping function to automatically
stopping training with monitoring validation loss
```

```
DEFINE reduce_lr_callback using ReduceLRonPlateau function to automatically
reduce learning rate value with monitoring validation loss
```

```
CREATE train_generator by applying train_datagen to training data.
```

```
CREATE validation_generator by applying validation_datagen to validation data.
```

```
CREATE cnn model using convolutional layers, pooling layers, dropout layers,
fully connected layers, input shape.
```

```
COMPILE model using Adam optimizer, binary_crossentropy loss, accuracy
metrics.
```

```
SET number of epochs, train_steps_per_epoch, validation_steps_per_epoch.
```

```
TRAIN model using fit method specifying train_generator, valid_generator,
callbacks.
```

#### 3.2.5 *Deployment*

*Deployment* merupakan proses integrasi model dalam format *tflite* kedalam sistem dengan berbasis *android*. File model akan diupload langsung kedalam folder dengan

nama tensorflow, untuk kemudian menghasilkan kode berupa parameter yang tersimpan didalam model. Kode tersebut akan ditambahkan pada kode pembuatan aplikasi, dan disesuaikan sehingga model dapat bekerja ketika aplikasi dijalankan.

### 3.2.6 *Testing process*

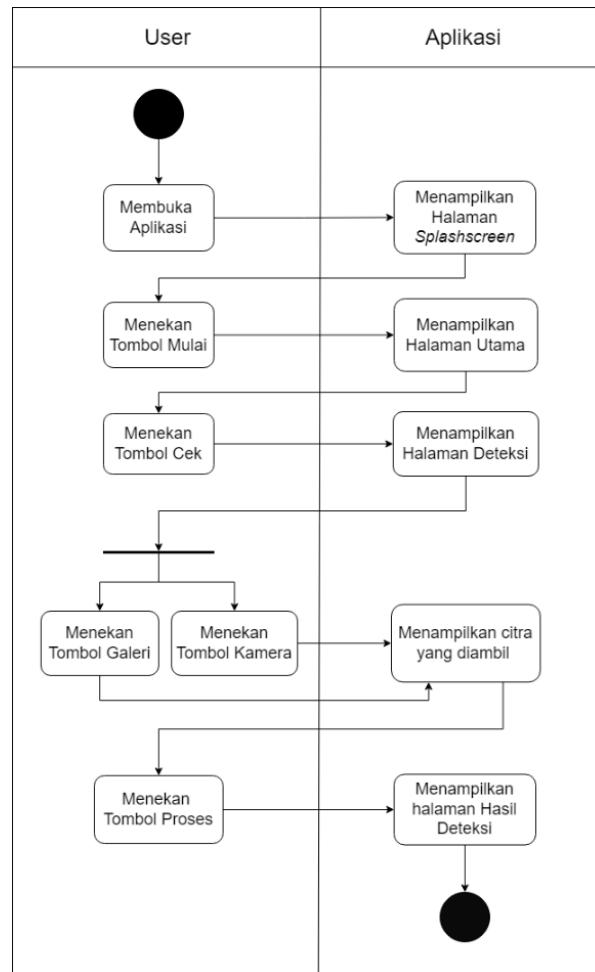
*Testing process* merupakan proses pengujian sistem dengan data uji, dalam penelitian ini digunakan data uji sebanyak 40 gambar dengan masing-masing kategori 20 citra konjungtiva anemia dan 20 citra konjungtiva non-anemia. Cara pengambilan citra konjungtiva mata disesuaikan dengan cara pemeriksaan medis untuk memperoleh hasil identifikasi yang tepat. Adapun tahap yang dilakukan, yaitu:

- Memastikan konjungtiva mata tidak mengalami sentuhan atau tekanan sebelum pemeriksaan. Hal ini dikarenakan konjungtiva mata bersifat sensitif sehingga warnanya sangat mudah mengalami perubahan.
- Memastikan ruangan / tempat pengambilan citra memiliki sumber pencahayaan yang baik (disarankan dilakukan pada ruang terbuka), sehingga diperoleh citra konjungtiva mata yang jelas.
- Melakukan pemeriksaan dengan menarik ke bawah, bagian bawah kelopak, secara perlahan hingga konjungtiva mata dapat terlihat dengan jelas. Kemudian, gambar konjungtiva mata dapat diambil dan digunakan pada aplikasi.

Setelah proses pengambilan gambar dilakukan dengan benar, citra yang diperoleh dapat digunakan pada aplikasi. Model akan dipanggil untuk melakukan proses identifikasi pada citra. Hasil prediksi kemudian akan dimunculkan beserta dengan informasi penanganan / pencegahan dasarnya.

## 3.3 Perancangan Antarmuka Sistem

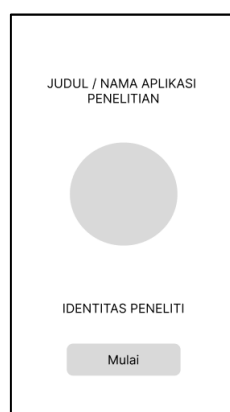
Tahap ini dilakukan untuk menampilkan hasil proses identifikasi kedalam tampilan yang lebih mudah dan nyaman digunakan oleh pengguna. Desain terdiri atas enam halaman, yaitu halaman *splashscreen*, utama, deteksi, hasil deteksi, panduan penggunaan aplikasi dan informasi anemia. Perancangan aplikasi dilakukan sesuai dengan diagram aktivitas yang menggambarkan alur proses penggunaan aplikasi.



**Gambar 3.16** Activity Diagram

### 3.3.1 Rancangan tampilan splashscreen

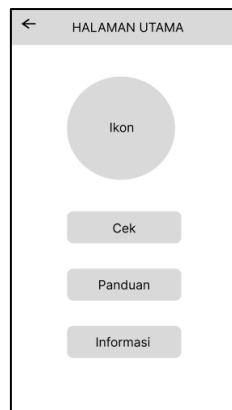
Pada saat pengguna membuka aplikasi, halaman *splashscreen* menjadi halaman awal yang akan ditampilkan pada aplikasi. Halaman ini memuat judul/nama aplikasi, logo universitas, identitas peneliti, dan tombol “Mulai” untuk menuju kehalaman utama.



**Gambar 3.17** Rancangan Halaman *Splashscreen*

### 3.3.2 Rancangan tampilan halaman utama

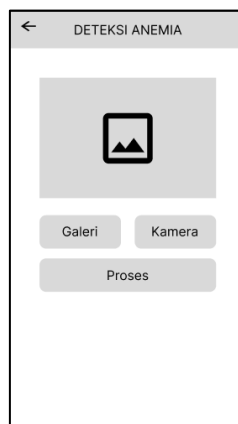
Pada halaman utama terdapat tiga tombol yaitu tombol “Cek” yang dapat membawa pengguna ke halaman deteksi anemia, tombol “Panduan” yang akan mengarahkan pengguna ke halaman panduan penggunaan aplikasi, serta tombol “Informasi” untuk membawa pengguna ke halaman yang menyajikan informasi seputar anemia.



**Gambar 3.18** Rancangan Halaman Utama

### 3.3.3 Rancangan tampilan halaman deteksi

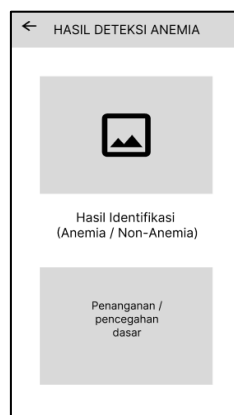
Pengguna dapat melakukan deteksi anemia menggunakan citra konjungtiva mata pada halaman ini. Pengguna dapat menekan tombol “Galeri” untuk mengambil gambar pada perangkat yang digunakan. Pengguna juga dapat mengambil gambar konjungtiva mata secara langsung, dengan menekan tombol “Kamera” untuk membuka kamera perangkat yang digunakan dan menangkap gambar. Untuk memulai proses identifikasi, pengguna dapat menekan tombol “Proses”.



**Gambar 3.19** Rancangan Halaman Deteksi

### 3.3.4 Rancangan tampilan halaman hasil deteksi

Tampilan pada halaman hasil deteksi akan muncul ketika pengguna menekan tombol “Proses” pada halaman deteksi. Apabila gambar diprediksi sebagai “Anemia” maka halaman akan menampilkan gambar, hasil prediksi dan solusi penanganan dasarnya. Sebaliknya jika diprediksi sebagai “Non-anemia” maka halaman akan menampilkan gambar, hasil prediksi dan solusi pencegahan dasarnya. Untuk kembali melakukan identifikasi, pengguna dapat menekan *icon arrowback* pada sudut kiri tampilan, untuk kembali ke halaman deteksi.



**Gambar 3.20** Rancangan Halaman Hasil Deteksi

## BAB 4

### IMPLEMENTASI DAN PENGUJIAN

#### 4.1 Implementasi Sistem

Pembentukan model untuk proses identifikasi anemia melalui citra konjungtiva mata menggunakan *K-Means Clustering* dan *Convolutional Neural Network (CNN)*, akan diterapkan kedalam sistem berbasis *android* yang dibentuk menggunakan bahasa pemrograman *python* dan *kotlin*. Pembentukan sistem membutuhkan sejumlah data pendukung, perangkat keras (*hardware*), perangkat lunak (*software*) dengan spesifikasi tertentu, serta beberapa modul pendukung lainnya.

##### 4.1.1 Perangkat keras dan perangkat lunak

Sistem dibangun dengan menggunakan perangkat keras dan lunak berikut.

- CPU: *Processor Intel® Core™ i3-1115G4*
- GPU: *NVIDA GeForce MX330*
- RAM: *8GB DDR4*
- Kapasitas *Harddisk 1TB*
- Sistem Operasi *Windows 11 Home Single Language 64-bit*

Sistem yang dibangun juga menggunakan beberapa pendukung lainnya sebagai berikut.

- *Google Colab RAM 12.7GB T4 GPU*
- *Android Studio Giraffe*
- *Python3 version 3.10.12*
- *Tensorflow version 2.15.0*
- *OpenCV version 4.8.0*
- *Numpy version 1.25.2*
- *Java version 17.0.2*

##### 4.1.2 Implementasi data

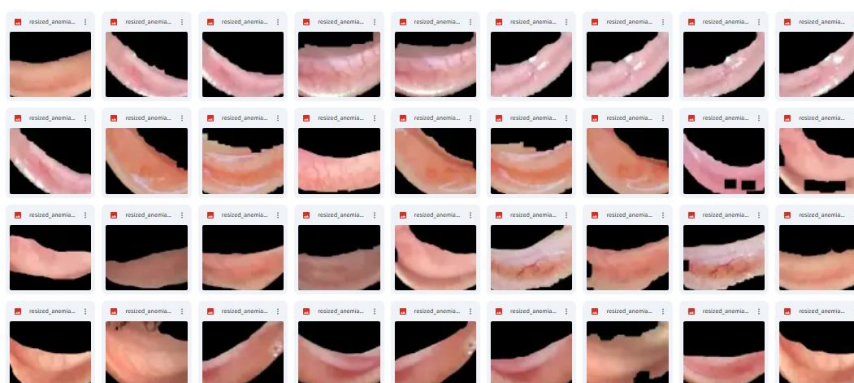
Penggunaan data diperoleh melalui penelitian oleh Dimauro et al. (2023) dan pengambilan secara langsung melalui klinik kesehatan di daerah kecamatan Tarutung, Tapanuli Utara. Data berjumlah 440 gambar dengan pembagian 220 citra konjungtiva anemia dan 220 citra konjungtiva non-anemia, yang diambil dengan menggunakan

kamera 16 MP pada ruang dengan pencahayaan yang baik. Sebelum digunakan sebagai *input* pada model CNN, data terlebih dahulu melalui proses *image pre-processing* dan *image segmentation*. Pada pra-pemrosesan gambar, dilakukan teknik augmentasi untuk menambah variasi data. Augmentasi dilakukan dengan dua jenis augmentasi dan menghasilkan jumlah data baru sebanyak 1600 citra latih dan 400 citra validasi.

**Tabel 4.1** Pembagian Data Setelah Augmentasi

No	Data	Data Latih	Data Validasi	Data Uji	Jumlah
1	Citra konjungtiva anemia	800	200	20	1020
2	Citra konjungtiva non-anemia	800	200	20	1020
<b>Total Data</b>		<b>1600</b>	<b>400</b>	<b>40</b>	<b>2040</b>

Citra yang digunakan sebagai *input* pada model adalah citra yang sudah mengalami segmentasi. Contoh citra yang digunakan untuk pelatihan model dimuat pada Gambar 4.1 dan 4.2



**Gambar 4.1** Citra Konjungtiva Anemia



**Gambar 4.2** Citra Konjungtiva Non-anemia



#### 4.1.3 Implementasi desain interface

Implementasi desain tampilan pada sistem disesuaikan pada tampilan yang telah dirancang pada bab sebelumnya. Tampilan akhir sistem dapat dilihat pada penjelasan berikut.

##### 4.1.3.1 Tampilan halaman *splashscreen*

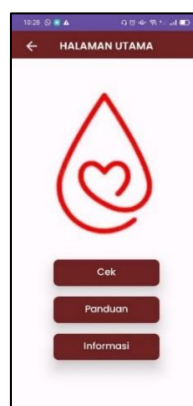
Tampilan awal aplikasi akan menampilkan halaman *splashscreen* yang memuat judul penelitian, logo, identitas peneliti dan instansi peneliti. Pengguna dapat menekan tombol “Mulai” untuk memulai dan mengakses halaman lainnya pada aplikasi.



**Gambar 4.3** Tampilan Halaman *Splashscreen*

##### 4.1.3.2 Tampilan halaman utama

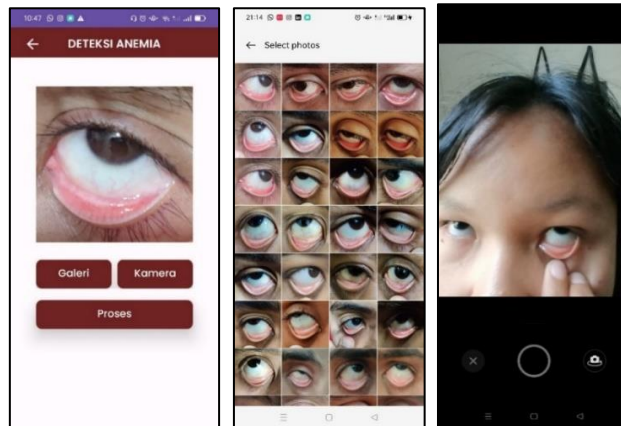
Tampilan pada halaman utama menampilkan tiga tombol utama dalam aplikasi, yaitu tombol “Cek” yang mengarah pada halaman deteksi, tombol “Panduan” yang mengarah pada halaman panduan penggunaan aplikasi, dan tombol “Informasi” yang mengarah pada halaman informasi anemia.



**Gambar 4.4** Tampilan Halaman utama

#### 4.1.3.3 Tampilan halaman deteksi

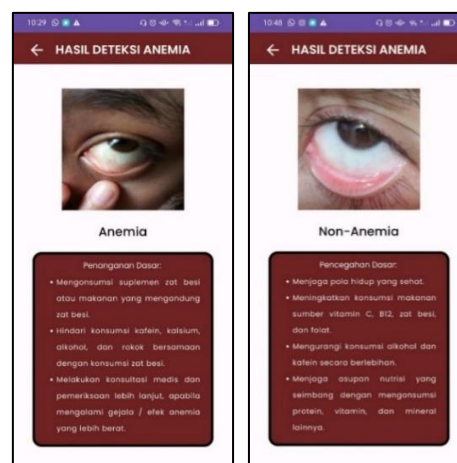
Halaman deteksi merupakan halaman untuk melakukan proses identifikasi citra yang diunggah oleh pengguna. Terdapat tiga tombol yang dapat digunakan, yaitu tombol “Galeri” untuk mengambil gambar dari perangkat, tombol “Kamera” untuk menangkap gambar secara langsung, dan tombol “Proses” untuk memulai proses identifikasi pada gambar yang digunakan.



**Gambar 4.5** Tampilan Halaman Deteksi, Galeri, Kamera

#### 4.1.3.4 Tampilan halaman hasil deteksi

Pada halaman ini, hasil identifikasi yang ditampilkan adalah hasil prediksi citra ke dalam kelas anemia atau non-anemia, serta pencegahan / penanganan dasar yang dapat dilakukan pengguna.



**Gambar 4.6** Tampilan Halaman Hasil Deteksi

#### 4.1.3.5 Tampilan halaman panduan

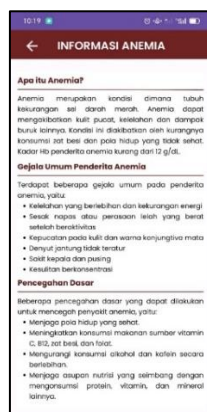
Halaman panduan menampilkan panduan pengambilan citra yang benar serta tutorial penggunaan aplikasi yang dapat digunakan pengguna sebagai arahan selama menggunakan aplikasi.



**Gambar 4.7** Tampilan Halaman panduan

#### 4.1.3.6 Tampilan halaman informasi anemia

Pada halaman informasi, diberikan beberapa informasi tentang anemia, gejala umum, dan pencegahan dasar yang dapat dilakukan untuk mencegah anemia.



**Gambar 4.8** Tampilan Halaman Informasi Anemia

## 4.2 Prosedur Operasional

Sistem yang dibangun terdiri atas halaman *splashscreen*, halaman utama, halaman deteksi dan hasilnya, halaman panduan, dan halaman informasi tentang anemia. Fitur utama pada sistem ini adalah fitur deteksi yang dapat diakses melalui tombol “Deteksi” pada halaman utama. Sistem akan mengarahkan pengguna ke halaman deteksi, dimana terdapat tiga tombol yaitu, tombol Galeri, Kamera, dan Proses. Jika pengguna ingin

melakukan deteksi dengan gambar yang tersimpan pada perangkat yang digunakan, maka pengguna dapat menekan tombol “Galeri” dan sistem akan mengakses ruang galeri pada perangkat. Namun apabila pengguna ingin mengambil gambar secara langsung, pengguna dapat menekan tombol “Kamera” untuk mengizinkan sistem mengakses kamera pada perangkat. Sistem akan menampilkan gambar yang diambil oleh pengguna. Jika gambar sudah sesuai, pengguna dapat menggunakan tombol “Proses” untuk memulai proses identifikasi pada gambar yang diunggah. Pada tahap ini, sistem akan mengenali citra dengan memanfaatkan kinerja model yang sebelumnya sudah diintegrasikan dengan sistem. Hasil identifikasi akan ditampilkan pada halaman hasil deteksi. Selain hasil identifikasi, sistem juga memberikan informasi tentang penanganan dan pencegahan dasar yang dapat dilakukan oleh pengguna sesuai dengan hasil deteksi. Apabila pengguna ingin melakukan identifikasi kembali, maka pengguna dapat menekan tombol *icon arrowback* pada sudut kiri atas halaman. Sistem juga menyediakan halaman panduan yang dapat diakses oleh pengguna untuk dapat mempelajari cara penggunaan aplikasi. Halaman panduan dapat ditampilkan menggunakan tombol “Panduan” yang ada pada halaman utama. Selain itu, sistem juga menyediakan halaman informasi yang bertujuan untuk memberikan edukasi tentang penyakit anemia, gejala umum dan pencegahan dasar yang dapat dilakukan. Halaman informasi dapat diakses dengan menekan tombol “Informasi” pada halaman utama.

#### **4.3 Pelatihan Sistem**

Pelatihan sistem dibangun dengan menggunakan dua algoritma, yaitu *K-Means Clustering* dan CNN (*Convolutional Neural Network*). Algoritma *k-means* bertujuan untuk melakukan segmentasi pada citra latih. Proses ini bertujuan untuk mengurangi *noise* dan memfokuskan model pada area yang akan diidentifikasi pada citra. Untuk mempermudah proses segmentasi, dilakukan dua pra-pemrosesan gambar dengan dua teknik penting yaitu *image equalization* dan *image unsharp masking filter*. Pada proses *equalization*, digunakan fungsi *histogram equalization* untuk melakukan distribusi intensitas piksel pada citra, sementara *unsharp masking filter* menggunakan bantuan fungsi *gaussian blur* untuk mempertajam tepi citra. Hasil dari kedua proses akan digunakan pada proses segmentasi citra. Pada implementasi segmentasi, terdapat beberapa parameter penting, yaitu:

- ‘k’: merupakan nilai yang menentukan jumlah pusat kluster yang akan dibentuk. Pada penelitian digunakan nilai  $k=2$  sebagai nilai kluster terbaik dari uji coba kluster dengan menggunakan metode *Elbow*.
- ‘centers’: merupakan pusat kluster untuk masing-masing kelompok warna. Pada penelitian ini terdapat dua pusat kluster yang terbentuk. Proses pemilihan pusat kluster akan dilakukan secara acak dengan percobaan sebanyak 10 kali. Nilai terbaik akan dijadikan pusat kluster sebagai acuan untuk mengelompokkan nilai-nilai piksel yang dekat dengan pusat kluster. Pusat kluster akan terus diperbaharui sampai proses klustering berhenti.
- ‘criteria’: merupakan aturan yang diberikan untuk membatasi proses klustering. Pada penelitian ini digunakan dua kriteria, yaitu jumlah iterasi maksimum dan akurasi minimum yang diinginkan. Pada penelitian ini, proses klustering akan berhenti ketika mencapai iterasi maksimum 100 dan nilai perubahan rata-rata pusat kluster mencapai minimum 0.2.

Hasil segmentasi dijadikan sebagai data *input* untuk proses pelatihan model dengan arsitektur CNN. Sebelum memulai proses pelatihan dilakukan penentuan beberapa parameter yang dapat mempengaruhi performa model. Pada penelitian ini, dilakukan beberapa percobaan pada nilai dari parameter *batch size* dan *learning rate*.

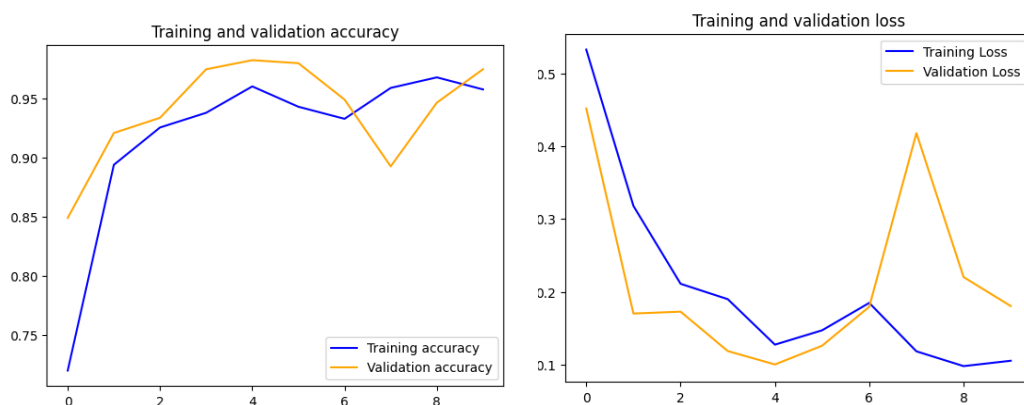
Menurut Rochmawati et al. (2021), pemilihan nilai *batch size* dan *learning rate* dapat mempengaruhi hasil *training* model. Penggunaan *batch size* yang terlalu kecil atau besar dapat mempengaruhi lama waktu proses komputasi dan kemungkinan model mengalami *overfitting*. Penggunaan nilai *learning rate* juga dapat menyebabkan *underfitting* apabila menggunakan nilai yang terlalu besar, dan *overfitting* jika menggunakan nilai yang terlalu kecil. Penentuan nilai *batch size* yang tepat dapat ditemukan dengan melakukan percobaan pada beberapa nilai *batch size*. Sementara untuk nilai *learning rate*, dapat dilakukan dengan pendekatan *cyclic learning rate* oleh Smith et al. (2018), dimana akan dilakukan penurunan nilai *learning rate* secara berkala.

Pada penelitian ini, dilakukan beberapa percobaan pada *batch size* 30, 50, dan 64. Pada proses pelatihan, digunakan beberapa fungsi *callback* yaitu *EarlyStopping* dan *ReduceLrOnPlateau*. Penggunaan *EarlyStopping* bertujuan untuk menghentikan proses pelatihan apabila tidak ada penurunan nilai *validation loss* dalam lima *epoch* berturut-turut. Sementara penggunaan *ReduceLrOnPlateau* bertujuan untuk menerapkan

pendekatan *cyclic learning rate* dengan mengurangi nilai *learning rate* awal secara berkala apabila tidak ada penurunan nilai *validation loss* dalam tiga *epoch* berturut-turut. Pada penelitian ini, digunakan nilai awal *learning rate* 0.001 dengan jumlah *epoch* 35. Hasil *training* pada masing-masing percobaan akan menghasilkan *output* yang berbeda-beda. Pada penelitian ini digunakan nilai *validation loss* sebagai metrik penilaian untuk mengukur performa model. Metrik penilaian ini akan mengukur seberapa baik model dalam mengenali data validasi sebagai data yang belum pernah dilihat oleh model melalui penurunan nilai *validation loss* secara signifikan pada proses pelatihan model.

#### 4.3.1 Percobaan nilai batch size 30

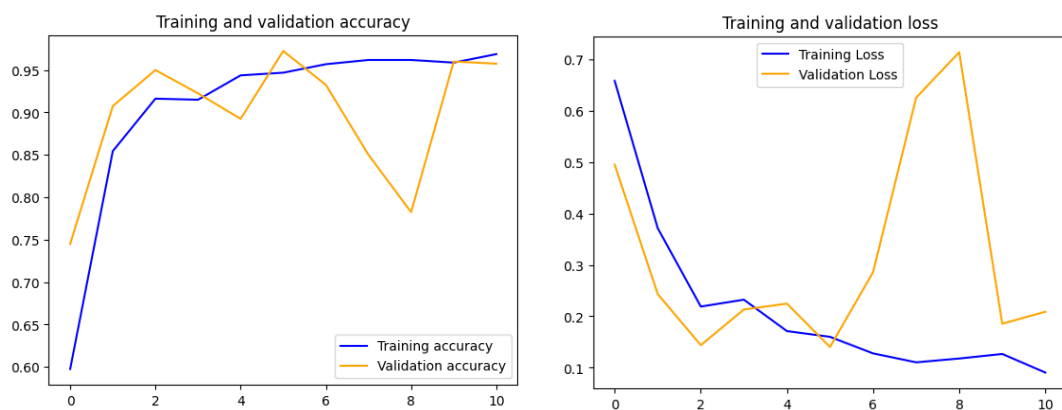
Proses *training* model dengan nilai *batch size* 30 menghasilkan 530 *total steps*. Penurunan nilai *validation loss* terjadi pada *epoch* 1 hingga *epoch* 5 dengan nilai terendahnya mencapai 0.1228. Nilai *validation loss* mengalami perubahan yang tidak stabil pada *epoch* 6 hingga *epoch* 8. Karena tidak ditemukannya penurunan nilai *validation loss* pada tiga *epoch* secara berturut-turut, maka fungsi *ReduceLrOnPlateau* akan aktif untuk menurunkan nilai *learning rate* menjadi 0.0001. Namun, penurunan *learning rate* tidak mempengaruhi perubahan nilai pada *validation loss* pada lima *epoch* berturut-turut sehingga pelatihan dihentikan pada *epoch* 10, dengan fungsi *EarlyStopping*. Dari analisa grafik yang dihasilkan pada Gambar 4.9, model masih mengalami kenaikan dan penurunan nilai *validation loss* yang tidak stabil yang memungkinkan model masih mengalami *overfitting*.



**Gambar 4.9** Grafik Akurasi dan Loss Batch Size 30

#### 4.3.2 Percobaan nilai batch size 50

Proses *training* model dengan *batch size* 50 menghasilkan 352 *total steps*. Pada proses pelatihan model ini, ditemukan penurunan *validation loss* pada *epoch* 1 hingga *epoch* 6 dengan nilai terendahnya sebesar 0.1402. Pada *epoch* berikutnya, perubahan pada nilai *validation loss* terlihat tidak stabil selama tiga *epoch* berturut-turut, sehingga diperlukan fungsi *ReduceLrOnPlateau* untuk mengurangi nilai *learning rate* menjadi 0.0001. Namun pada lima *epoch* berikutnya, tidak ditemukan penurunan yang baik pada nilai *validation loss* sehingga fungsi *EarlyStopping* menghentikan pelatihan pada *epoch* 11. Dari analisa grafik pada Gambar 4.10, perubahan pada nilai *validation loss* masih tidak stabil sehingga memungkinkan model mengalami *overfitting*.

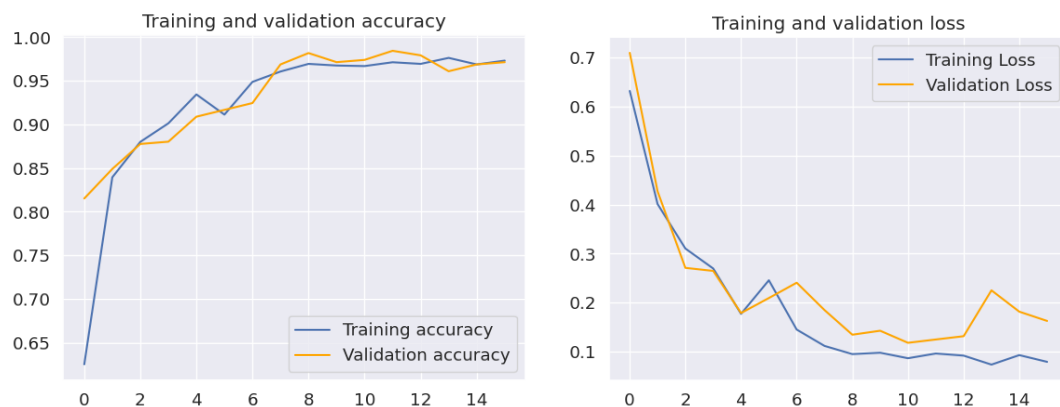


**Gambar 4.10** Grafik Akurasi dan Loss Batch Size 50

#### 4.3.3 Percobaan nilai batch size 64

Proses *training* model dengan *batch size* 64 menghasilkan 400 *total steps*. Penurunan nilai *validation loss* ditemukan pada *epoch* 1 hingga *epoch* 5 dengan nilai terendahnya yaitu 0.1789. Namun, terjadi kenaikan dan penurunan yang kurang stabil pada *epoch* 6 hingga *epoch* 8 sehingga dilakukan pengurangan nilai *learning rate* menjadi 0.0001 oleh fungsi *ReduceLrOnPlateau*. Penurunan nilai *learning rate* mempengaruhi perubahan nilai *validation loss*, sehingga terjadi penurunan yang signifikan pada *epoch* 9 sehingga diperoleh nilai *validation loss* sebesar 0.1346. Penurunan nilai *validation loss* kembali terjadi pada *epoch* 11 dengan nilai *validation loss* menjadi 0.1181. Pada tiga *epoch* berikutnya, tidak ditemukan penurunan nilai *validation loss*, sehingga fungsi *ReduceLrOnPlateau* kembali menurunkan nilai *learning rate* menjadi 0.00001. Namun, pada dua *epoch* berikutnya, tidak ditemukan penurunan nilai *validation loss* sehingga fungsi *EarlyStopping* menghentikan proses pelatihan pada *epoch* 16. Dari analisa grafik

pada Gambar 4.11, ditemukan perubahan nilai *validation loss* yang cukup stabil setelah beberapa penurunan nilai *validation loss* yang terjadi selama proses pelatihan. Penurunan nilai *validation loss* juga diikuti dengan peningkatan nilai akurasi yang cukup baik, dengan menggambarkan performa model yang lebih baik dari percobaan pelatihan model lainnya.



**Gambar 4.11** Grafik Akurasi dan *Loss Batch Size 64*

Dari percobaan yang dilakukan, beberapa model yang dihasilkan akan digunakan untuk memperoleh nilai *validation precision* dan *validation recall* sebagai bagian dari parameter pembandingan yang digunakan untuk menemukan model terbaik. Dari perbandingan yang dilakukan, ditemukan model terbaik dengan nilai parameter *batch size 64*, *learning rate 0.0001*, dan jumlah *epoch 11* yang menghasilkan *400 total steps*. Dari analisa yang dilakukan, penggunaan *batch size* yang lebih besar dapat menambah variasi kelompok data yang akan dipelajari oleh model pada satu *epoch*, sehingga penggunaan *batch size 64* lebih optimal untuk digunakan. Adapun hasil perbandingan nilai parameter yang digunakan pada percobaan proses *training* dirangkumkan pada Tabel 4.2.

**Tabel 4.2** Perbandingan *Training*





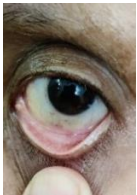
No	Parameter	Epoch	Val loss	Val Precision	Val Recall
1	Batch size 30 dan lr 0.001	5	0.1228	0.7907	0.7075
2	Batch size 50 dan lr 0.001	6	0.1402	0.8494	0.8125
3	Batch size 64 dan lr 0.001	5	0.1789	0.8404	0.7975
4	Batch size 64 dan lr 0.0001	9	0.1346	0.8494	0.8125
5	<b>Batch size 64 dan lr 0.0001</b>	<b>11</b>	<b>0.1181</b>	<b>0.9376</b>	<b>0.9375</b>






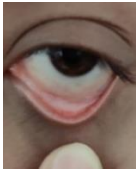

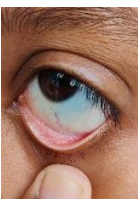


#### 4.4 Pengujian Sistem

Pengujian sistem pada data baru bertujuan untuk mengukur kemampuan sistem dalam mengidentifikasi penyakit anemia melalui warna pada konjungtiva mata. Pengujian dilakukan dengan data uji berjumlah 40 data, dengan masing-masing 20 data untuk setiap kelasnya. Pengujian ini dilakukan dengan menggunakan model terpilih yang memiliki nilai *learning rate* 0.0001, *batch size* 64, dengan jumlah *epoch* 11. Hasil pengujian untuk 20 data uji dirangkumkan pada Tabel 4.3. Untuk pengujian lengkapnya dapat dilihat pada halaman lampiran.








**Tabel 4.3** Pengujian Identifikasi Anemia

No	Gambar	Aktual	Prediksi	Status
1		Anemia	Anemia	Benar
2		Anemia	Anemia	Benar
3		Anemia	Anemia	Benar
4		Anemia	Anemia	Benar
5		Anemia	Anemia	Benar

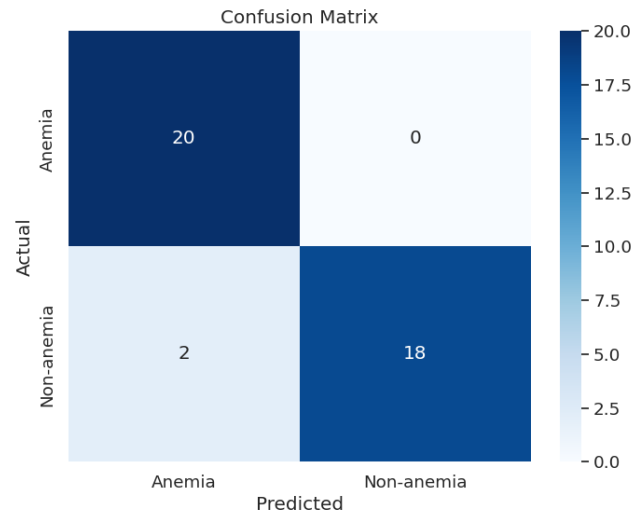
---

6		Anemia	Anemia	Benar
7		Anemia	Anemia	Benar
8		Anemia	Anemia	Benar
9		Anemia	Anemia	Benar
10		Anemia	Anemia	Benar
11		Non-anemia	Anemia	Salah
12		Non-anemia	Non-anemia	Benar
13		Non-anemia	Non-anemia	Benar

---

14		Non-anemia	Non-anemia	Benar
15		Non-anemia	Non-anemia	Benar
16		Non-anemia	Anemia	Salah
17		Non-anemia	Non-anemia	Benar
18		Non-anemia	Non-anemia	Benar
19		Non-anemia	Non-anemia	Benar
20		Non-anemia	Non-anemia	Benar

Pengujian pada 40 data gambar uji menghasilkan kesalahan prediksi pada 2 gambar. Keberhasilan dan kesalahan sistem dalam mengidentifikasi area konjungtiva mata pada citra digambarkan melalui *confusion matrix* pada Gambar 4.12.



**Gambar 4.12** *Confusion Matrix*

*Confusion matrix* akan menghasilkan nilai *true positive* (TP), *false positive* (FP), dan *false negative* (FN) yang dapat dilihat pada Tabel 4.4.

**Tabel 4.4** Perhitungan Hasil Uji

	Prediksi		
	TP	FP	FN
Anemia	20	0	0
Non-anemia	18	2	0

Pada pengujian data, ditemukan sebanyak 38 data bernilai *true positive* dan 2 data bernilai *false postive*. Nilai yang dihasilkan akan digunakan untuk menghitung nilai *precision*, *recall*, dan *F-1 Score*.

- *Precission*

Rumus perhitungan *precision* dapat dilihat pada persamaan 2.4

$$Precision \text{ anemia} = \frac{20}{20} = 1 \times 100\% = 100\%$$

$$Precision \text{ non-anemia} = \frac{18}{20} = 0.9 \times 100\% = 90\%$$

- *Recall*

Rumus perhitungan *recall* dapat dilihat pada persamaan 2.5

$$Recall \text{ anemia} = \frac{20}{20} = 1 \times 100\% = 100\%$$

$$Recall \text{ non-anemia} = \frac{18}{18} = 1 \times 100\% = 100\%$$

- *F-1 Score*

Rumus perhitungan *F-1 Score* dapat dilihat pada persamaan 2.6

$$F-1 \text{ Score anemia} = 2 \times \frac{1 \times 1}{1 + 1} = 1 \times 100\% = 100\%$$

$$F-1 \text{ Score non-anemia} = 2 \times \frac{0.9 \times 1}{0.9 + 1} = 0.947 \times 100\% = 94.7\%$$

Hasil perhitungan nilai *precision*, *recall*, dan *f-1 score* pada penelitian dapat dilihat pada Tabel 4.5.

**Tabel 4.5** Nilai *Precision*, *Recall*, dan *F-1 Score*

	<i>Precision</i>	<i>Recall</i>	<i>F1-Score</i>
Anemia	100%	100%	100%
Non-anemia	90%	100%	94.7%

Dari pengujian dan perhitungan nilai-nilai matrik penilaian yang dilakukan, maka diperoleh nilai akurasi dengan perhitungan rumus pada persamaan 2.3.

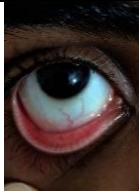





$$Accuracy = \frac{\text{Jumlah benar}}{\text{Total data}} \times 100\% = \frac{38}{40} \times 100\% = 95\%$$

Sistem yang dibangun dengan metode *K-Means Clustering* dan CNN pada penelitian identifikasi anemia melalui citra konjungtiva berhasil melakukan identifikasi dengan akurasi sebesar 95%. Sistem merupakan aplikasi berbasis *android* yang dapat mengidentifikasi citra konjungtiva mata melalui unggahan foto dari galeri perangkat atau pengambilan secara langsung menggunakan kamera. Meskipun sistem sudah bekerja dengan baik, masih terdapat 5% kesalahan yang dilakukan oleh sistem. Kesalahan dalam mengidentifikasi warna pada citra konjungtiva mata dapat disebabkan oleh tidak terlalu signifikannya perbedaan warna pada beberapa citra konjungtiva mata anemia dan non-anemia. Faktor lainnya adalah penggunaan cahaya yang digunakan pada saat pengambilan citra.

Pada Tabel 4.6 dilakukan percobaan penggunaan cahaya yang berbeda-beda pada citra. Pada percobaan pertama dengan citra non-anemia, tingkat cahaya dengan ukuran 50 - 100 lux masih dapat diprediksi dengan benar. Namun pada ukuran 500 – 1000 lux, sistem melakukan kesalahan prediksi. Kemudian dilakukan percobaan kedua dengan citra konjungtiva anemia. Pada tingkat cahaya 50 – 100 lux, sistem tidak

melakukan prediksi dengan benar. Namun, pada penggunaan cahaya  $\pm 500$  lux, sistem dapat melakukan prediksi dengan benar.

**Tabel 4.6** Percobaan Berdasarkan Cahaya Pengambilan

Gambar	Tingkat cahaya	Aktual	Prediksi	Status
	$\pm 50$ lux	Non-anemia	Non-anemia	Benar
	$\pm 100$ lux	Non-anemia	Non-anemia	Benar
	$\pm 500$ lux	Non-anemia	Anemia	Salah
	$\pm 1000$ lux	Non-anemia	Anemia	Salah
	$\pm 50$ lux	Anemia	Non-anemia	Salah
	$\pm 100$ lux	Anemia	Non-anemia	Salah



± 500 lux

Anemia

Anemia

Benar

Dari pengujian yang dilakukan, ditemukan faktor-faktor yang dapat mempengaruhi kerja sistem yaitu, kemiripan beberapa warna citra konjungtiva mata anemia dan non-anemia, kualitas citra serta pencahayaan yang digunakan.

1. Faktor pertama adalah perbedaan warna yang tidak signifikan pada beberapa citra anemia dan non-anemia. Penurunan kadar Hb yang mempengaruhi perubahan warna pada citra konjungtiva mata menghasilkan beberapa citra dengan warna yang sedikit mirip.
2. Faktor kedua adalah kualitas dan kejelasan citra yang digunakan. Pengambilan gambar yang *blur* dan tidak jelas dapat mempengaruhi kesalahan sistem dalam mengidentifikasi warna pada citra konjungtiva mata. Pengambilan data dengan kamera 16 MP menghasilkan citra yang jelas. Hal ini memungkinkan model melakukan kesalahan prediksi terhadap citra uji yang diidentifikasi.
3. Faktor ketiga adalah pencahayaan yang digunakan. Pada cara pengambilan citra secara medis, disarankan menggunakan cahaya alami (pemeriksaan pada ruang yang terbuka). Sementara pada proses pengujian ditemukan satu citra dengan prediksi salah. Pengujian yang dilakukan pada Tabel 4.6 menunjukkan beberapa kesalahan prediksi sistem pada citra dengan pencahayaan yang berbeda. Hal ini disebabkan oleh sifat warna yang sensitif, khususnya pada pencahayaan yang digunakan. Pada percobaan yang dilakukan, ditemukan kesalahan prediksi pada pengambilan citra anemia dengan pencahayaan yang minim. Sementara pada pengujian citra non-anemia, ditemukan kesalahan prediksi dengan pencahayaan yang berlebihan. Peningkatan dan penurunan cahaya yang signifikan dapat mempengaruhi warna asli citra. Pada penelitian digunakan jenis augmentasi *brightness range* dengan tujuan untuk menambah variasi data pada rentang kecerahan citra. Namun penggunaan augmentasi ini masih dinilai baik, karena menggunakan rentang kecerahan yang sedikit sehingga tidak mempengaruhi warna asli citra.

## **BAB 5**

### **PENUTUP**

#### **5.1 Kesimpulan**

Pengujian pada penelitian identifikasi citra konjungtiva mata menggunakan *K-Means Clustering* dan CNN (*Convolutional Neural Network*) menghasilkan kesimpulan sebagai berikut.

1. Penggunaan metode *K-Means Clustering* dan CNN untuk identifikasi anemia melalui citra konjungtiva mata dapat bekerja dengan baik dengan akurasi mencapai 95%.
2. Segmentasi dengan *K-Means Clustering* menghasilkan citra dengan *noise* lebih sedikit sehingga mempermudah CNN untuk melakukan pengenalan terhadap warna pada citra konjungtiva mata.
3. Model dengan performa terbaik diperoleh dengan nilai parameter *learning rate* 0.0001, *batch size* 64, dan *total steps* 400.
4. Faktor-faktor yang dapat mengakibatkan kesalahan prediksi model terhadap citra adalah pencahayaan yang digunakan serta kualitas dan kejelasan gambar pada saat pengambilan gambar secara langsung.

#### **5.2 Saran**

Beberapa saran untuk penelitian berikutnya.

1. Menggunakan dataset dengan kualitas yang baik dan jumlah yang banyak.
2. Memperhatikan beberapa faktor seperti cara pengambilan citra dan cahaya yang digunakan pada pengambilan citra secara langsung, untuk memperoleh citra gambar yang jelas dan baik.
3. Melakukan pengujian pada metode segmentasi dan klasifikasi yang berbeda dengan metode pada penelitian ini.
4. Membangun sistem deteksi secara *real-time* dengan metode *object detection*







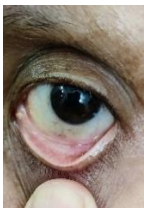


## DAFTAR PUSTAKA

- Amalia, E., Lamada, M., Kaswar, A. B., & Andayani, D. D. (2023). Klasifikasi Penyakit Anemia Berbasis Citra Palpebral Konjungtiva Dengan Algoritma Transfer Learning. *Jurnal Media Elektrik*, 20(2), 2721–9100. <https://ojs.unm.ac.id/mediaelektrik/article/view/44503>.
- Andaur, C. (2021). Using K-Means Clustering For Image Segmentation. *Medium.com*, 21 Januari 2021 (diakses 18 Februari 2024).
- Arifah, S., Swedia, E. R., & Septian, M. R. D. (2023). Analisis Perbandingan Algoritma Clustering Dalam Melakukan Segmentasi Warna Pada Citra Jajan Tradisional. *Sebatik*, 27(1), 70-76. <https://doi.org/10.46984/sebatik.v27i1.2273>.
- Asare, J. W., Appiahene, P., Arthur, E. J., Korankye, S., Afrifa, S., & Donkoh, E. T. (2023). Detection of anemia using conjunctiva images. *Medicine in Novel Technology and Devices*, 18. <https://doi.org/10.1016/j.medntd.2023.100237>
- Asiyah, S., Tritasmoro, I.I., & Sofia, S. (2022). Anemia Detection Through Conjunctiva on Eyes Using Principal Component Analysis Method and K-Nearest Neighbor. *Proccedings – 2022 8th Internasional Conference on Science and Technology, ICST 2022*. <https://doi.org/10.1109/ICST56971.2022.10136254>
- Azmi, K., Defit, S., & Putra Indonesia YPTK Padang Jl Raya Lubuk Begalung-Padang-Sumatera Barat, U. (2023). Implementasi Convolutional Neural Network (CNN) Untuk Klasifikasi Batik Tanah Liat Sumatera Barat. *Jurnal Unitek*, 16(1), 2023.
- Banik, P. P., Saha, R., & Kim, K. D. (2020). An Automatic Nucleus Segmentation and CNN Model based Classification Method of White Blood Cell. *Expert Systems with Applications*, 149. <https://doi.org/10.1016/j.eswa.2020.113211>
- Chai, A. L., Huang, O. Y., Rakočević, R., & Chung, P. (2021). Critical iron deficiency anemia with record low hemoglobin: a case report. *Journal of Medical Case Reports*, 15(1). <https://doi.org/10.1186/s13256-021-03024-9>
- Delgado, G., Roman, A., Alva, A., Saldivar, B., Zimic, M., Barrientos, F., & Salgado, M. (2018). *Method for the Automatic Segmentation of the Palpebral Conjunctiva using Image Processing*. <https://www.mathworks.com/help/images/index.html>
- Dimauro, G., Griseta, M. E., Camporeale, M. G., Clemente, F., Guarini, A., & Maglietta, R. (2023). An intelligent non-invasive system for automated diagnosis of anemia exploiting a novel dataset. *Artificial Intelligence in Medicine*, 136.
- Febrianti, A. (2018). *Penerapan Metode K-Means Clustering dan Support Vector Machine (SVM) dalam Identifikasi Api pada Citra Warna Digital*. UIN Syarif Hidayatullah. <http://dx.doi.org/10.24014/coreit.v6i1.9001>
- Fitriany, J., Saputri, A. I., Ilmu, S., & Anak, K. (2018). Anemia Defisiensi Besi. In *Jurnal Averrous* (Vol. 4, Issue 2).

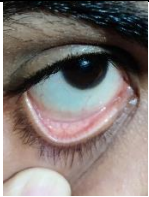
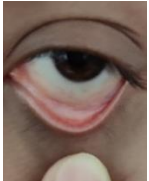
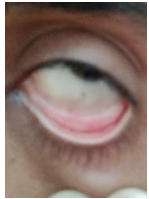




- Gede, I., Suryawan, T., Putu, I., Eka, A., & Udayana, D. (2019). Optimasi Convolutional Neural Network Untuk Deteksi Covid-19 Pada X-Ray Thorax Berbasis Dropout. *Jurnal Teknologi Informasi Dan Ilmu Komputer (JTIK)*. <https://doi/10/25126/jtiik.202295143>
- Gonzales, R. C., Woods, R. E. (2018). *Digital Image Processing. 4th Edition*. Pearson Education: New York.
- Guntoro, I., Marisa, M., D., & Hidayati, R. (2022). Penerapan Dropout Pada Jaringan Saraf Tiruan Backpropagation Dalam Mengklasifikasikan Tingkat Fine Fuel Moisture Code (FFMC) Untuk Kebakaran Hutan Dan Lahan. *Jurnal Komputer Dan Aplikasi*.
- Handoyo, S. Y. (2006). Gambaran Umum Sistem Penglihatan (mata). *Anatomi Fisiologi Untuk Paramedis, 19-45*. Diakses dari [https://repository.unimus.ac.id/1072/3/BAB II.pdf](https://repository.unimus.ac.id/1072/3/BAB%20II.pdf)
- Hermawati, F., A. (2013). *Pengolahan Citra Digital: Konsep & Teori*. Yogyakarta: Andi.
- Islam, M. K., Ali, M.S., Miah, M.S., Rahman, M. M., Alam, M. S., & Hossain, M.A. (2021). Brain Tumor Detection in MR Image Using Superpixels, Principal Component Analysis And Template Based K-Means Clustering Algorithm. *Machine Learning with Applications*. <https://doi.org/10.1016/j.mlwa.2021.100044>
- Liu, Z., Yang, Ding, F., Xu, Y., & Han, X. (2021). Background Dominant Colors Extraction Method Based On Color Image Quicck Fuzzy C-Means Clustering Algorithm. *Defence Technology*. <https://doi.org/10.1016/j.dt.2020.10.002>
- Kemkes.go.id (2018, April 11). Diakses pada 20 November 2023, dari [https://kesmas.kemkes.go.id/assests/upload/dir\\_519d41d8cd98f00/files/Hasil-risikesdas-2018\\_1274.pdf](https://kesmas.kemkes.go.id/assests/upload/dir_519d41d8cd98f00/files/Hasil-risikesdas-2018_1274.pdf)
- Mehmood, N., Khan, S. J., & Rashid, M. (2022). K-means Clustering based Color Segmentation on Vitiligo Skin Lesion. *2022 International Conference on Emerging Trends in Smart Technologies, ICETST 2022*. <https://doi.org/10.1109/ICETST55735.2022.9922940>
- Mishra, M. (2020). *Convolutional Neural Networks, Explained. towardsdatascience.com*, 27 Agustus 2020 (diakses 18 Februari 2024).
- Monika, K. M., Vignesh, A. N., Kumari, C. U., Kumar, M. N. V. S. S., & Lydia, E. L. (2020). Skin cancer detection and classification using machine learning. *Materials Today: Proceedings*, 33, 4266–4270. <https://doi.org/10.1016/j.matpr.2020.07.366>
- Muhayati, A., Ratnawati, D., Ilmu Kesehatan, F., & Pembangunan Nasional Veteran Jakarta Jalan Limo Raya Kelurahan Limo Kecamatan Limo Kota, U. (2019). Hubungan Antara Status Gizi dan Pola Makan dengan Kejadian Anemia Pada Remaja Putri. *Jurnal Ilmiah Ilmu Keperawatan Indonesia*.

- Muningsih, E., & Kiswati, S. (2018). Sistem Aplikasi Berbasis Optimasi Metode Elbow Untuk Penentuan Clustering Pelanggan. *JOUTICA*, 3(1). <https://doi.org/https://doi.org/10.30736/jti.v3i1.196>
- Noor, B.B., Anwar, M.S., & Mrinmoy, D. (2019). Comparative Study Between Decision Tree, SVM, and KKN to Predict Anaemic Condition. *Proceedings – 2022 IEEE Internasional Conference on Biomedica Engineering Computer and Information*. <https://doi.org/10.1109/BECITHCON48839.2019.9063188>
- Raghav, P. 2018. Understanding of Convolutional Neural Network (CNN) – Deep Learning. *Medium.com*, 4 Maret 2018 (diakses 18 Februari 2024).
- Rasywir, E., Sinaga, R., Pratama, Y., Dinamika, U., & Jambi, B. (2020). Analisis dan Implementasi Diagnosis Penyakit Sawit dengan Metode Convolutional Neural Network (CNN). *Jurnal Informatika Dan Komputer*, 22(2). <https://doi.org/10.31294/p.v2i2>
- Rochmawati, N., Hidayati, H. B., Yamasari, Y., Peni, H., Tjahyaningtijas, A., Yustanti, W., & Prihanto, A. (2021). Analisa Learning rate dan Batch size Pada Klasifikasi Covid Menggunakan Deep learning dengan Optimizer Adam. *Journal Information Engineering and Educational Technology*.
- Sabilla, I. A. (2020). Arsitektur Convolutional Neural Network (CNN) Untuk Klasifikasi Jenis Dan Kesegaran Buah Pada Neraca Buah. Institut Teknologi Sepuluh Nopember.
- Salsabila, G. A., Sa'Idah, S., & Rustam. (2022). Anemia Detection Through Conjunctiva on Eyes Using Histogram of Oriented Gradients and Decision Tree Classification. *Proceedings - 2022 8th International Conference on Science and Technology, ICST 2022*. <https://doi.org/10.1109/ICST56971.2022.10136290>
- Sena, S. 2017. Pengenalan Deep Learning Part 7: Convolutional Neural Network (CNN). *Medium.com*, 13 November 2017 (diakses 18 Februari 2024).
- Setiawan, W., & Purnama, A. (2020). Tobacco Leaf Images Clustering Using DarkNet19 And K-Means. *Proceeding – 6th Information Technology Internasional Seminar, ITIS 2020*, 269-273. <https://doi.org/10.1109/ITIS50118.2020.9321035>
- Sheth, T. N., Choudhry, N. K., Bowes, M., & Detsky, A. S. (1997). The Relation of Conjunctival Pallor to the Presence of Anemia. In *J GEN INTERN MED* (Vol. 12).
- Smith, L. N. (2018). A disciplined approach to neural network hyper-parameters: Part 1-learning rate, batch size, momentum, and weight decap.*arXiv preprint arXiv:1803.09820*.
- World Health Organization. (2023, May 1). Anaemia. World Health Organization. Diakses pada 26 Oktober 2023, dari <https://www.who.health-topics/anaemia>.
- Yuliatin.(2018). Kehamilan. Jilid I. Buku Kedokteran. EGC. Jakarta.

## LAMPIRAN


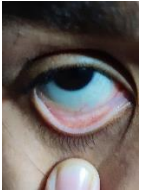
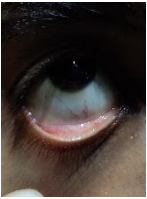



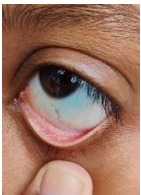
No	Gambar	Aktual	Prediksi	Status
1		Anemia	Anemia	Benar
2		Anemia	Anemia	Benar
3		Anemia	Anemia	Benar
4		Anemia	Anemia	Benar
5		Anemia	Anemia	Benar
6		Anemia	Anemia	Benar
7		Anemia	Anemia	Benar

---

8		Anemia	Anemia	Benar
9		Anemia	Anemia	Benar
10		Anemia	Anemia	Benar
11		Anemia	Anemia	Benar
12		Anemia	Anemia	Benar
13		Anemia	Anemia	Benar
14		Anemia	Anemia	Benar

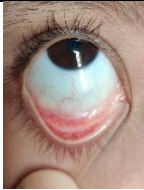

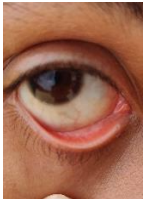

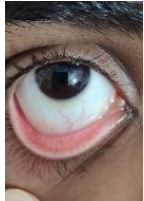


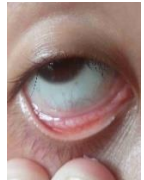
---

---

15		Anemia	Anemia	Benar
16		Anemia	Anemia	Benar
17		Anemia	Anemia	Benar
18		Anemia	Anemia	Benar
19		Anemia	Anemia	Benar
20		Anemia	Anemia	Benar
21		Non-anemia	Anemia	Salah

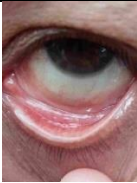






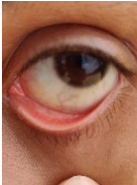
---

---

22		Non-anemia	Non-anemia	Benar
23		Non-anemia	Non-anemia	Benar
24		Non-anemia	Non-anemia	Benar
25		Non-anemia	Non-anemia	Benar
26		Non-anemia	Anemia	Salah
27		Non-anemia	Non-anemia	Benar
28		Non-anemia	Non-anemia	Benar
29		Non-anemia	Non-anemia	Benar

---




---

30		Non-anemia	Non-anemia	Benar
31		Non-anemia	Non-anemia	Benar
32		Non-anemia	Non-anemia	Benar
33		Non-anemia	Non-anemia	Benar
34		Non-anemia	Non-anemia	Benar
35		Non-anemia	Non-anemia	Benar
36		Non-anemia	Non-anemia	Benar
37		Non-anemia	Non-anemia	Benar

---



---

38		Non-anemia	Non-anemia	Benar
39		Non-anemia	Non-anemia	Benar
40		Non-anemia	Non-anemia	Benar

---