

**APLIKASI DETEKSI *SAMPLE*/CITRA DARAH MENGGUNAKAN
METODE *CAPS NEURAL NETWORK* BERBASIS *ANDROID***

SKRIPSI

**MUHAMMAD RIZKY DHARMAWAN NASUTION
171401109**



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

MEDAN

2024

**APLIKASI DETEKSI *SAMPLE*/CITRA DARAH MENGGUNAKAN
METODE *CAPS NEURAL NETWORK* BERBASIS *ANDROID***

SKRIPSI

Diajukan untuk melengkapi dan memenuhi syarat memperoleh ijazah
Sarjana Ilmu Komputer

MUHAMMAD RIZKY DHARMAWAN NASUTION

171401109



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

MEDAN

2024

PERSETUJUAN

Judul : APLIKASI DETEKSI SAMPLE/CITRA DARAH
MENGGUNAKAN METODE CAPS NEURAL
NETWORK BERBASIS ANDROID

Kategori : SKRIPSI

Nama : MUHAMMAD RIZKY DHARMAWAN
NASUTION

Nomor Induk Siswa : 171401109

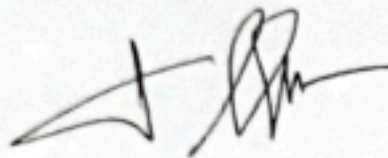
Program Studi : SARJANA (S-1) ILMU KOMPUTER

Fakultas : FAKULTAS ILMU KOMPUTER DAN
TEKNOLOGI INFORMASI UNIVERSITAS
SUMATERA UTARA

Komisi Pembimbing :

Dosen Pembimbing I

Dosen Pembimbing II



Dr. T. Henny Febriana Harumy S.Kom., M.Kom.
NIP. 198802192019032016

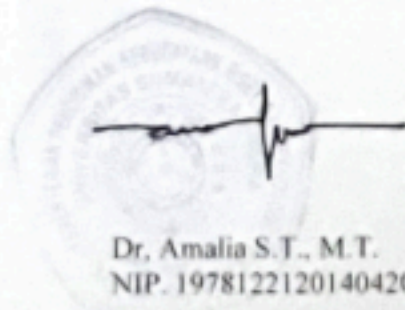


Desilia Selvida S.Kom., M.Kom.
NIP. 198912052020012001

Diketahui/Disetujui oleh

Program Studi S1 Ilmu Komputer

Ketua,



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



LEMBAR PERNYATAAN

Saya yang bertanda tangan di bawah ini menyatakan dengan sesungguhnya bahwa Skripsi saya yang berjudul “Aplikasi Deteksi *Sample/Citra* Darah Menggunakan Metode *Caps Neural Network* Berbasis *Android*” adalah benar hasil karya saya sendiri dan judul yang dimaksud belum pernah dimuat, dipublikasikan atau diteliti oleh mahasiswa lain dalam konteks penulisan Skripsi Program Studi S-1 Ilmu Komputer Fakultas Ilmu Komputer dan Teknologi Informasi Universitas Sumatera Utara. Semua Sumber data dan informasi yang diperoleh telah dinyatakan jelas dan benar apa adanya. Apabila di kemudian hari pernyataan ini tidak benar, saya bersedia menerima sanksi yang ditetapkan oleh pihak Universitas Sumatera Utara.

Medan, 24 Juni 2024
Penulis,

Muhammad Rizky Dharmawan Nasution
171401109

iii

PENGHARGAAN

Alhamdulillah, Puji dan syukur penulis panjatkan kepada Allah SWT, karena karunia Nya penulis dapat menyelesaikan penelitian serta penyusunan skripsi sebagai syarat untuk menyelesaikan pendidikan pada program studi S1 Ilmu Komputer Fakultas Ilmu Komputer dan Teknologi Informasi Universitas Sumatera Utara serta mendapat gelar sarjana.

Pada kesempatan kali ini dengan penuh kerendahan hati, penulis ingin memberikan ucapan terima kasih yang sebesar-besarnya kepada :

1. Bapak Dr. Muryanto Amin, S.Sos., M.Si selaku Rektor Universitas Sumatera Utara.
2. Ibu Dr. Maya Silvi Lydia, M.Sc. selaku Dekan Fakultas Ilmu Komputer dan Teknologi Informasi Universitas Sumatera Utara.
3. Ibu Dr. Amalia ST., M.T. selaku Ketua Program Studi Ilmu Komputer

Universitas Sumatera Utara dan selaku Dosen Pembimbing Akademik yang telah memberikan dorongan kepada penulis sehingga skripsi ini dapat terselesaikan.

4. Ibu Sri Melvani Hardi S.Kom., M.Kom selaku Sekretaris Program Studi Ilmu Komputer Universitas Sumatera Utara.
5. Ibu Dr. T. Henny Febriana Harumy S.Kom., M.Kom. sebagai Dosen Pembimbing I yang telah memberikan dukungan berupa bimbingan, kritik, serta saran, dan motivasi kepada penulis sehingga skripsi ini dapat terselesaikan.
6. Ibu Desilia Selvida S.Kom., M.Kom selaku Dosen Pembimbing II yang telah memberikan masukan, kritik serta saran kepada penulis sehingga skripsi ini dapat terselesaikan.
7. Ibu Fuzy Yustika Manik S.Kom., M.Kom. selaku Dosen Penguji I yang telah memberikan masukan, kritik serta saran kepada penulis sehingga skripsi ini dapat tersempurnakan.
8. Bapak Amer Sharif selaku Dosen Penguji II yang telah memberikan masukan, kritik serta saran kepada penulis sehingga skripsi ini dapat tersempurnakan.
9. Orang tua penulis, Indra Barkah Nasution S.E, dan yunizar Harahap yang selalu mendoakan penulis dan memberikan dorongan serta motivasi untuk dapat menyelesaikan skripsi ini.

iv

10. Kakak penulis beserta suami, Nindy Jeihan Fadillah Nasution S.Si., M.Si, dan Arief Hidayat Hasibuan S.E yang selalu mendoakan penulis dan memberikan dorongan serta motivasi untuk dapat menyelesaikan skripsi ini.
11. Saudara Yoga Dede Irawan yang telah bersedia menjadi wadah diskusi dan mendengar keluh kesah serta menjadi pendukung penulis baik secara jasa maupun memperbaiki semangat yang terkadang redup ketika skripsi ini dibuat.
12. Teman-teman Freepict yang diantaranya Muhammad Adrianto, Thoriq Ramadhana Amdanni S.Kom, Dery Ardianto S.Kom, Andre Ariesto Siregar S.Kom, Surya Maris Panensa S.Kom, Hagi Fansury, Muhammad Purnomo Adjie S.Kom, Sheilla Suhaila Matondang S.Kg., Samfriandy Gusborn Situmeang yang menjadi sarana informasi dan media penyampaian keluhan selama melakukan pengerjaan skripsi ini.
13. Semua pihak yang terlibat baik secara langsung maupun tidak langsung dalam membantu penulis menyelesaikan skripsi.

Penulis menyadari akan jauhnya kata sempurna pada skripsi yang penulis buat. Oleh karena itu, penulis akan selalu belajar lebih banyak serta dapat mengimplementasikan ilmu yang penulis dapatkan guna memberikan manfaat kepada

orang lain. Penulis berharap agar kiranya pembaca dapat memberikan kritik serta saran demi kesempurnaan skripsi ini.

Medan, 24 Juni 2024

Penulis,



Muhammad Rizky Dharmawan Nasution

171401109

v

ABSTRAK

APLIKASI DETEKSI *SAMPLE*/CITRA DARAH MENGGUNAKAN METODE *CAPS NEURAL NETWORK* BERBASIS *ANDROID*

Teknologi informasi saat ini menjadi suatu kebutuhan yang tidak dapat dipisahkan pada proses kehidupan, perkembangannya saat ini sangat dibutuhkan dalam menyelesaikan berbagai permasalahan yang muncul di kehidupan masyarakat. Gigitan nyamuk *Anopheles betina* yang terinfeksi merupakan vektor penularan penyakit ini. Parasit menyerang sel darah merah setelah menyerang tubuh melalui gigitan nyamuk dan akhirnya menetap di hati. Inovasi dalam teknologi yang berhubungan dengan kesehatan telah berkembang dengan pesat hingga saat ini. Salah satunya yaitu proses pengecekan darah untuk mengetahui parasit malaria. Penggunaan metode yang beragam tentu dapat menentukan berjalannya proses pengolahan citra, salah satu metode pengolahan citra selain yang disebutkan diatas adalah metode *Neural Network*. Salah satu model buatan tersebut adalah Jaringan Syaraf Tiruan (JST), yang terus berupaya meniru cara otak belajar sebenarnya. Penulis menggunakan metode penelitian dengan studi pustaka yang dikumpulkan dari berbagai sumber seperti buku-buku, jurnal, laporan-laporan dan tinjauan pustaka lainnya yang berkaitan dengan penelitian. Penelitian ini berfokus pada pengembangan aplikasi pendeteksi parasit malaria menggunakan Caps Neural Network (CapsNet) berbasis Android untuk citra darah. Hasil dari penelitian ini menunjukkan sistem pengenalan deteksi sampel citra darah dengan menggunakan metode *caps* dan algoritma *Convolutional Neural Network* diperoleh hasil akurasi terbaik sebesar 96.92%. Tingkat akurasi ini dipengaruhi oleh nilai *learning rate* pada proses pelatihan, selain itu tingkat resolusi gambar citra darah, jumlah data latih dan data uji serta jumlah lapisan layer pada arsitektur CNN pun berpengaruh.

Kata kunci : Android, Aplikasi, Capsule Neural Network, Darah, Malaria, Teknologi

vi

ABSTRACT

BLOOD *SAMPLE*/IMAGE DETECTION APPLICATION USING *ANDROID* BASED *NEURAL NETWORK CAPS METHOD*

Information technology has now become a necessity that cannot be separated from the process of life, its development is currently very much needed in solving various

problems that arise in people's lives. The bite of an infected female *Anopheles* mosquito is the vector for transmitting this disease. Parasites attack red blood cells after invading the body through mosquito bites and eventually settle in the liver. *Innovation in health*

related technology has developed rapidly to date. One of them is the blood checking process to find out malaria parasites. The use of various methods can certainly determine the progress of the image processing process, one image processing method other than those mentioned above is the Neural Network method. One of these artificial models is Artificial Neural Networks (ANN), which continually attempts to imitate the way the brain actually learns. The author uses research methods with literature studies collected from various sources such as books, journals, reports and other literature reviews related to research. This research focuses on developing a malaria parasite detection application using Android-based Caps Neural Network (CapsNet) for blood images. The results of this research show that the blood image sample detection recognition system using the caps method and the Convolutional Neural Network algorithm obtained the best accuracy results of 96.92%. This level of accuracy is influenced by the learning rate value in the training process, apart from that the level of blood image resolution, the amount of training data and test data and the number of layers in the CNN architecture also have an influence.

Keyword : Android, Application, Blood, Capsule Neural Network, Technology.

vii

DAFTAR ISI

| | |
|----------------------------------|------|
| PERSETUJUAN | i |
| PERNYATAAN..... | ii |
| PENGHARGAAN | |
| iii ABSTRAK..... | |
| v <i>ABSTRACT</i> | |
| vi | |
| DAFTAR ISI | vii |
| DAFTAR TABEL | x |
| DAFTAR GAMBAR | xi |
| DAFTAR LAMPIRAN | xiii |
| BAB 1 PENDAHULUAN | 1 |
| 1.1. Latar Belakang | 1 |
| 1.2. Rumusan Masalah | 3 |
| 1.3. Batasan Masalah..... | 4 |
| 1.4. Tujuan Penelitian | 4 |
| 1.5. Manfaat Penelitian | 4 |
| 1.6. Metodologi Penelitian | 5 |
| 1.7. Sistematika Penulisan | 5 |
| BAB 2 LANDASAN TEORI | |

| | |
|---|---------------------------|
| 7 2.1. <i>Caps Neural Network</i> | 7 |
| 2.1.1. Sejarah <i>Caps Neural Network</i> | 7 2.1.2. |
| Konsep <i>Capsule Neural Network</i> | 9 |
| 2.1.3. Perhitungan <i>Caps Neural Network</i> | 10 |
| 2.2. <i>Confusion Matrix</i> | 10 |
| 2.2.1. Metode Klasifikasi Sel Darah menggunakan Jaringan Kapsul | 11 |
| 2.3. <i>Datasets</i> | 12 |
| 2.4. Malaria | 13 |
| 2.4.1. Klasifikasi Parasit Malaria..... | 15 |
| 2.4.2. Siklus Hidup Malaria..... | 16 |
| 2.5. Aplikasi | 19 |
| 2.6. <i>Android</i> | 19 |
| 2.6.1. <i>Android Studio</i> | 20 |
| | viii |
| 2.6.2. <i>Android SDK</i> | 21 |
| 2.6.3. <i>Android JDK</i> | 21 |
| 2.6.4. <i>Android Development Tools</i> | 22 |
| 2.6.5. <i>Android Virtual Device</i> | 22 |
| 2.7. Algoritma | 22 |
| 2.8. <i>Mobile</i> | 23 |
| 2.9. <i>Python</i> | 24 |
| 2.10. <i>OpenCv</i> | 24 |
| 2.11. Citra Digital..... | 25 |
| 2.11.1. Tipe-Tipe Citra Digital | 26 |
| 2.11.2. Pengolahan Citra Digital | 27 2.12. |
| <i>Machine Learning</i> | 28 2.13. |
| Penelitian Relevan..... | 31 BAB 3 |
| ANALISIS DAN PERANCANGAN SISTEM | 33 3.1. |
| Analisis Sistem..... | 33 3.2. |
| Analisis Data | 34 3.3. |
| Rancangan Arsitektur Sistem | 34 3.4. |
| Data Penelitian | 36 3.4.1. |
| <i>Data Sample</i> | 36 3.4.2. <i>Data</i> |
| <i>Augmentation</i> | 37 3.5. <i>Processing</i> |
| Data | 37 3.6. Rancang |
| Desain Aplikasi | 39 3.6.1. Tampilan |
| Rancangan <i>Dashboard</i> | 39 3.6.2. Tampilan |
| Rancangan Galeri | 40 3.6.3. Tampilan |

| | | |
|--|----|---------------------|
| Rancangan Hasil..... | 41 | BAB 4 |
| IMPLEMENTASI DAN PENGUJIAN | 42 | 4.1. |
| Implementasi | 42 | 4.1.1. |
| <i>Tampilan Aplikasi</i> | 42 | 4.2. |
| Pengujian | 47 | 4.3. |
| Analisis | 50 | 4.3.1. |
| Perolehan Data | 50 | 4.3.2. |
| <i>Analisis Processing</i> | 51 | 4.3.3. <i>Image</i> |
| <i>Preprocessing and Data Augmentation</i> | 52 | |
| | | ix |
| 4.3.4. <i>Training dan Result</i> | 52 | |
| 4.3.5. <i>Performance Matrics</i> | 54 | |
| 4.3.6. <i>Result</i> | 54 | |
| BAB 5 KESIMPULAN DAN SARAN | 59 | |
| 5.1. Kesimpulan | 59 | |
| 5.2. Saran | 59 | |
| DAFTAR PUSTAKA | 60 | |
| LAMPIRAN | | |
| 63 | | |

x

DAFTAR TABEL

| | |
|--|----|
| Tabel 2.1. Lama Siklus <i>Ekso-Eritrositik</i> dan <i>Eritrositik</i> Pada <i>Plasmodium</i> | 18 |
| Tabel 4.1. <i>Training Model</i> | 53 |
| Tabel 4.2. <i>Confusion Matrix</i> | 54 |
| Tabel 4.3. Perbandingan Akurasi Pada Semua Model..... | 54 |
| Tabel 4.4. Perbandingan Akurasi Pada Semua Model..... | 58 |

xi

DAFTAR GAMBAR

| | |
|--|----|
| Gambar 2.1. Informasi Spasial Dari Angka-Angka..... | 9 |
| Gambar 2.2. Model <i>Caps Nets</i> yang dioptimalkan | 11 |
| Gambar 2.3. Siklus Hidup <i>Plasmodium</i> | 18 |
| Gambar 2.4. Multi-disiplin <i>Machine Learning</i> | 28 |
| Gambar 2.5. Tipe Masalah Dalam <i>Data Science</i> | 30 |

| | |
|--|----|
| Gambar 3.1. Proses Klasifikasi Sel Darah | 35 |
| Gambar 3.2. Sebuah <i>Subset</i> dari Sample darah yang tidak terinfeksi Malaria yang Disediakan oleh <i>National Institute of Health</i> (NIH)..... | 36 |
| Gambar 3.3. Sebuah <i>Subset</i> dari Sample darah yang terinfeksi Malaria berdasarkan Plasmodium yang Disediakan oleh <i>National Institute of Health</i> (NIH)..... | 36 |
| Gambar 3.4. Proses Data <i>Augmentation</i> | 37 |
| Gambar 3.5. Arsitektur <i>Capsule Network</i> | 38 |
| Gambar 3.6. Tampilan Rancangan <i>Dashboard</i> | 39 |
| Gambar 3.7. Tampilan Rancangan Galeri..... | 40 |
| Gambar 3.8. Tampilan Rancangan Hasil | 41 |
| Gambar 4.1. Tampilan Menu Utama Aplikasi..... | 43 |
| Gambar 4.2. Tampilan Unggah Citra..... | 43 |
| Gambar 4.3. Tampilan Menu Galeri | 44 |
| Gambar 4.4. Tampilan Hasil Plasmodium Falciparum..... | 44 |
| Gambar 4.5. Tampilan Hasil Plasmodium Malariae..... | 45 |
| Gambar 4.6. Tampilan Hasil Plasmodium Ovale | 45 |
| Gambar 4.7. Tampilan Hasil Plasmodium Vivax | 46 |
| Gambar 4.8. Tampilan Hasil Tidak Terinfeksi | 46 |
| Gambar 4.9. Hasil Pengujian Pertama | 47 |
| Gambar 4.10. Hasil Pengujian Kedua | 48 |
| Gambar 4.11. Hasil Pengujian Ketiga | 48 |
| Gambar 4.12. Hasil Pengujian Keempat..... | 49 |
| Gambar 4.13. Hasil Pengujian Kelima | 50 |
| Gambar 4.14. <i>Processing CNN</i> | 51 |
| Gambar 4.15. Sel Darah Merah Tanpa Augmentasi | 52 |

xii

| | |
|---|----|
| Gambar 4.16. Sel Darah Merah dengan Augmentasi..... | 52 |
| Gambar 4.17. Akurasi Model <i>Transfer Training</i> Sebelum dan Sesudah <i>Fine-Tuning</i> | 53 |
| Gambar 4.18. Akurasi Klasifikasi Dataset Rasio 30 : 70..... | 55 |
| Gambar 4.19. Akurasi Klasifikasi Dataset Rasio 50 : 50..... | 56 |
| Gambar 4.20. Akurasi Klasifikasi Dataset Rasio 70 : 30..... | 57 |

xiii

BAB 1

PENDAHULUAN

1.1 Latar Belakang

Teknologi informasi saat ini menjadi suatu kebutuhan yang tidak dapat dipisahkan pada proses kehidupan, perkembangannya saat ini sangat dibutuhkan dalam menyelesaikan berbagai permasalahan yang muncul di kehidupan masyarakat. Perkembangan ini sudah mulai memasuki pada tahap perkembangan kecerdasan buatan, dimana komputer dapat memberi suatu keputusan berdasarkan Analisis dari data yang di dapatkan, sehingga hal ini dapat membantu siapapun mendapatkan kebutuhan data hasil Analisis awal lebih cepat. Perkembangannya kini merambah ke bidang medis. Berkat Teknologi Informasi, beberapa kemajuan telah dicapai di bidang administrasi kesehatan, perawatan pasien, dan studi ilmu kesehatan secara keseluruhan, yang semuanya berkontribusi terhadap pesatnya perluasan sektor kesehatan. Di

seluruh dunia, masyarakat mulai memperhatikan pelayanan kesehatan yang mengandalkan teknologi informasi. Hal ini terutama karena kemajuan teknologi memiliki potensi besar untuk memperbaiki kehidupan masyarakat. (Yanni, 2018).

Bayi, balita, dan wanita hamil sangat rentan terhadap dampak mematikan malaria, yang masih menjadi masalah kesehatan masyarakat yang utama. Malaria juga berpotensi menurunkan produktivitas dalam bekerja karena menyebabkan anemia secara langsung. Masih terdapat risiko kematian akibat malaria di Indonesia jika masyarakat tidak mendapatkan pengobatan. Menurut data Organisasi Kesehatan Dunia (WHO) pada tahun 2022, diperkirakan terdapat 811.636 kasus terkonfirmasi pada tahun 2021 (sehatnegeriku.kemkes.go.id, 2023). Hingga akhir tahun 2021, 347 kabupaten dan kota di Tanah Air telah berhasil memberantas penyakit malaria.

Jumlah ini meningkat dibandingkan tahun sebelumnya yang berjumlah 318 kabupaten dan kota. Pemberantasan penyakit malaria secara bertahap pada

tahun 2030 merupakan target program pengendalian malaria di Indonesia. (Laporan Kinerja Kesehatan Tahun 2023).

Parasit Plasmodium bertanggung jawab atas penyakit yang dikenal sebagai malaria. Gigitan nyamuk *Anopheles* betina yang terinfeksi merupakan vektor penularan penyakit ini. Parasit menyerang sel darah merah setelah menyerang tubuh melalui gigitan nyamuk dan akhirnya menetap di hati. Malaria dapat ditularkan kepada manusia tidak hanya melalui gigitan nyamuk tetapi juga melalui donor organ, transfusi darah, berbagi jarum suntik, dan janin yang terinfeksi oleh ibunya. Beberapa daerah di Indonesia masih mengidap penyakit malaria sehingga menjadi penyakit endemik. Wilayah tersebut antara lain Maluku, Nusa Tenggara Timur, Sulawesi Papua, Papua Barat, serta sebagian Kalimantan dan Sumatera.

Inovasi dalam teknologi yang berhubungan dengan kesehatan telah berkembang dengan pesat hingga saat ini. Meski demikian, beberapa kegiatan masih dilakukan dengan cara tradisional. Salah satunya yaitu proses pengecekan darah untuk mengetahui parasit malaria. Maka dari itu dianggap penting untuk saat ini dilakukan Analisis terhadap citra darah untuk mengetahui apakah darah tersebut terdapat atau terjangkit parasit malaria atau tidak perkembangan terkini dalam teknologi pengolahan citra (*image processing*) telah menarik minat banyak orang, dan mereka ingin mempelajari lebih lanjut tentang topik ini sehingga mereka dapat menerapkannya dalam kehidupan sehari-hari. Pemrosesan citra digital tidak dapat dipisahkan dari identifikasi objek kreatif seiring dengan peningkatan tersebut. (Putri, 2016).

Berbagai bidang, mulai dari seni dan penglihatan manusia hingga astronomi dan teknik, berkontribusi pada penciptaan citra, yang merupakan penggambaran objek dua dimensi di dunia yang terlihat. Ini adalah susunan dua dimensi titik atau pixel berwarna. (Hutahaean, Waluyo, dan Rais 2019).

Komputer dan manusia sama-sama dapat memperoleh manfaat dari pemrosesan gambar digital, suatu metode untuk menyempurnakan gambar diam atau bergerak sehingga lebih mudah dipahami dan digunakan (Effendi, Fitriyah, dan Effendi 2017). Area AI yang dikenal sebagai "pemrosesan

3

gambar" berfokus pada pencarian solusi terhadap masalah dengan menganalisis representasi digital dari objek fisik. Perhitungan matematis pada objek geometris atau pixel dapat dilakukan dengan menggunakan metode pencitraan. Membedakan objek dalam suatu gambar dimungkinkan dengan menetapkan nilai perbedaan yang dapat dihitung secara matematis. Warna, tekstur, dan

bentuk merupakan ciri pembeda suatu benda (Widyaningsih 2017). Informasi digital memungkinkan dilakukannya penerapan pengelompokan atau pengelompokan objek. Penggunaan metode yang beragam tentu dapat menentukan berjalannya proses pengolahan citra, salah satu metode pengolahan citra selain yang disebutkan diatas adalah metode *Neural Network*. Salah satu model buatan tersebut adalah Jaringan Syaraf Tiruan (JST), yang terus berupaya meniru cara otak belajar sebenarnya. Model matematika kognisi manusia yang menyeluruh adalah inspirasi bagi neuron. Premis dasarnya adalah bahwa pemrosesan data terjadi di blok bangunan paling dasar yang dikenal sebagai neuron. Ada beban yang terkait dengan setiap tautan yang memungkinkan sinyal berpindah antar sel saraf (Lesnussa, Sinay, dan Idah, 2017). Sinyal keluaran ditentukan dengan mengaktifkan setiap sel saraf dari total sinyal tertimbang yang masuk. Berdasarkan data historis, jaringan saraf tiruan dapat mengidentifikasi tindakan. Jaringan saraf tiruan, yang dapat belajar dari pengalaman dan bahkan dapat menghasilkan prediksi dari data yang belum dipelajari, akan menganalisis data historis.

Berdasarkan fenomena diatas penulis memiliki ide untuk mengembangkan teknologi dalam bidang kesehatan khususnya untuk mendeteksi citra darah pada sel darah untuk penyakit malaria, dan dituangkan dalam bentuk karya ilmiah dengan judul “**APLIKASI DETEKSI *SAMPLE/CITRA DARAH MENGGUNAKAN METODE CAPS NEURAL NETWORK BERBASIS ANDROID***”

1.2. Rumusan Masalah

Berdasarkan latar belakang tersebut, maka peneliti dapat merumuskan permasalahan sebagai berikut:

1. Bagaimana implementasi aplikasi deteksi *sample/citra* darah menggunakan metode *caps neural network* berbasis *android*.
2. Bagaimana visualisasi deteksi *sample/citra* darah menggunakan metode *caps neural network* berbasis *android*.

4

1.3. Batasan Masalah

Berdasarkan rumusan masalah dan tujuan penelitian diatas agar penelitian tidak terlalu melebar maka diambil batasan masalah pada penelitian ini, yakni:

1. Penelitian ini berfokus pada pembuatan aplikasi pendeteksi sel citra darah untuk penyakit malaria yang diintegrasikan dengan sistem.

2. Proses pengolahan informasi menggunakan metode *Neural Network* dengan pendekatan *Caps*, dan menggunakan Bahasa pemrograman *python* yang berbasis *android* dengan *datasets* dari *National Institutes of Health* (NIH) dengan nama malaria dengan menggunakan 100 dataset.
3. Versi *android* pada sistem yang digunakan adalah minimal versi 8 atau *android* oreo.

1.4. Tujuan Penelitian

Berikut adalah tujuan penelitian ini, yang diambil dari rumusan masalah yang telah dibahas sebelumnya:

1. Untuk mengimplementasikan model aplikasi deteksi *sample/citra* darah menggunakan metode *caps neural network* berbasis *android*.
2. Untuk memvisualisasikan cara kerja sistem saat *processing* pendeteksian *sample/citra* darah.
3. Menentukan minimal versi *android* sesuai kebutuhan sistem.

1.5 Manfaat Penelitian

Manfaat dari penelitian ini adalah untuk mengetahui klasifikasi *sample/citra* darah untuk penyakit malaria. Hasil pengembangan model aplikasi pada teknologi kesehatan yang telah terintegrasikan ini dapat memudahkan proses pendeteksian sel darah pada penyakit tertentu, sehingga identifikasi penyakit dapat diketahui sejak dini melalui sel darah yang telah diketahui oleh sistem.

5

1.6. Metodologi Penelitian

Berikut ini metodologi penelitian yang digunakan yaitu :

1. Studi Pustaka

Dalam penelitian ini, penulis menggunakan metode studi pustaka atau studi literatur untuk meninjau dan mengumpulkan berbagai referensi dari buku- buku, jurnal, laporan-laporan dan tinjauan pustaka lainnya yang memiliki hubungan dengan penelitian yang akan dilakukan.

2. Perancangan Sistem

Perancangan sistem dilakukan dalam bentuk *flowchart*.

3. Implementasi Sistem

Implementasi dari sistem yang akan dilakukan dibangun sesuai dengan perancangan yang dibuat dengan Bahasa pemrograman *python*, metode *Neural Network* yang berbasis *Android*.

4. Pengujian Sistem

Sistem yang telah dibuat akan diuji coba untuk melihat dan memastikan bahwa sistem tersebut berjalan sesuai dengan semestinya.

5. Dokumentasi

Dokumentasi yang dilakukan dalam penelitian ini dengan pembuatan laporan mengenai pembahasan program yang dirancang sebagai bukti penulis melakukan penelitian.

1.7. Sistematika Penulisan

Untuk mempermudah pembahasan pada laporan skripsi ini secara menyeluruh, maka dirumuskan sistematika penulisan yang mengacu pada pedoman penulisan yang ada. Berikut ini susunan sistematika penulisan pada laporan skripsi ini :

BAB 1 PENDAHULUAN

Bab ini memberikan penjelasan terkait latar belakang penelitian, rumusan dan batasan masalah, tujuan penelitian, manfaat penelitian, metode penelitian, serta sistematika penulisan.

6

BAB 2 LANDASAN TEORI

Bab ini memberikan penjelasan landasan teori yang digunakan dalam pembahasan penelitian ini sebagai acuan terkait dengan rancangan Aplikasi Deteksi Sample/Citra Darah Menggunakan Metode *Caps Neural Network* Berbasis *Android*.

BAB 3 ANALISIS DAN PERANCANGAN SISTEM

Bab ini memberikan penjelasan analisis terhadap permasalahan dalam penelitian serta menjelaskan gambaran terkait rancangan aplikasi Deteksi Sample/Citra Darah.

BAB 4 IMPLEMENTASI DAN PENGUJIAN SISTEM

Bab ini memberikan penjelasan terkait dengan hasil dari perancangan aplikasi Deteksi Sample/Citra Darah serta melakukan pengujian terhadap aplikasi tersebut yang dilanjutkan dengan pembahasan mengenai analisis data.

BAB 5 KESIMPULAN DAN SARAN

Bab ini memberikan penjelasan terhadap kesimpulan dan saran dari kebutuhan rancangan Aplikasi Deteksi Sample/Citra Darah Menggunakan Metode *Caps Neural Network* Berbasis *Android*.

BAB 2

TINJAUAN PUSTAKA

2.1. *Caps Neural Network*

Capsule Neural Network, atau *CapsNet*, adalah jenis arsitektur jaringan saraf tiruan dalam ilmu komputer yang dikembangkan oleh *Geoffrey Hinton* dan rekan-rekannya pada tahun 2017. *CapsNet* dirancang untuk mengatasi beberapa masalah dalam jaringan saraf konvensional, terutama dalam pengenalan objek dan pengenalan pola.

Menurut (en.wikipedia.org, 2023) Perbedaan utama antara *CapsNet* dan jaringan saraf konvensional adalah penggunaan kapsul (kapsul) sebagai unit pemrosesan dasar. Kapsul yang merupakan kelompok neuron yang bekerja sama untuk mengidentifikasi fitur-fitur objek dalam berbagai orientasi dan proporsi. Kapsul-kapsul ini juga mampu mengidentifikasi hubungan spasial antara fitur-fitur tersebut, yang membuat *CapsNet* lebih mampu mengatasi permasalahan transformasi dalam pengenalan objek, seperti rotasi atau pergeseran objek. Jaringan kapsul (*Capsule Network*) yang dikembangkan baru-baru ini berpotensi mengatasi beberapa kekurangan jaringan saraf konvolusional tradisional. Dengan mengganti aktivasi skalar standar dengan *vektor*, dan dengan menghubungkan *neuron* buatan dengan cara baru, jaringan kapsul bertujuan untuk menjadi perkembangan besar berikutnya untuk aplikasi visi komputer.

2.1.1. *Sejarah Caps Neural Network*

Pada tahun 2000, *Geoffrey Hinton* dkk. menggambarkan sistem pencitraan yang menggabungkan segmentasi dan pengenalan ke dalam satu proses inferensi menggunakan pohon parse. Apa yang disebut jaringan kredibilitas menggambarkan distribusi gabungan atas variabel laten dan kemungkinan pohon penguraian.

Sistem itu terbukti berguna pada *database* digit tulisan tangan MNIST (*Modified National Institute of Standards and Technology database*). (en.wikipedia.org, 2023). Mekanisme perutean dinamis untuk jaringan kapsul

diperkenalkan oleh *Hinton* dan timnya pada tahun 2017. Pendekatan tersebut diklaim dapat mengurangi tingkat kesalahan pada MNIST dan mengurangi ukuran set pelatihan. Hasilnya diklaim jauh lebih baik daripada CNN dengan angka yang sangat tumpang tindih. Dalam ide awal *Hinton*, satu kolom mini akan mewakili dan mendeteksi satu entitas multidimensi. (Sabour dkk, 2017).

2.1.2. Konsep Capsule Neural Network

Kapsul (***Capsule***) adalah sekumpulan neuron yang aktif secara individual untuk berbagai properti suatu jenis objek, seperti posisi, ukuran, dan warna. Secara formal, kapsul adalah sekumpulan *neuron* yang secara kolektif menghasilkan vektor aktivitas dengan satu elemen untuk setiap *neuron* guna menampung nilai instansiasi neuron tersebut (misalnya, rona). (Sabour dkk, 2017). Program grafis menggunakan nilai instansiasi untuk menggambar suatu objek. *Caps Nets* mencoba untuk mendapatkan ini dari masukan mereka. Probabilitas kehadiran entitas dalam masukan tertentu adalah panjang vektor, sedangkan orientasi vektor mengkuantifikasi properti kapsul. (Srihari & Sargur, 2017). Neuron buatan secara tradisional menghasilkan aktivasi skalar dan bernilai nyata yang secara longgar mewakili kemungkinan suatu observasi. *Capsnet* menggantikan detektor fitur keluaran skalar dengan kapsul keluaran vektor dan pengumpulan maksimal dengan perutean berdasarkan perjanjian. (Sabour dkk, 2017). Karena kapsul bersifat independen, ketika beberapa kapsul menyatu, kemungkinan deteksi yang benar jauh lebih tinggi. Sekelompok minimal dua kapsul yang mempertimbangkan entitas enam dimensi akan setuju dalam 10% secara kebetulan hanya sekali dalam sejuta percobaan. Ketika jumlah dimensi meningkat, kemungkinan terjadinya kesepakatan di cluster yang lebih besar dengan dimensi yang lebih tinggi menurun secara eksponensial. (Sabour dkk, 2017).

9

Lapisan yang lebih tinggi dalam jaringan menerima keluaran yang dibundel dari kapsul tingkat yang lebih rendah dan memprosesnya. Selain menghasilkan pose berdimensi tinggi (20-50+), sebuah *cluster* membuat kapsul yang lebih tinggi mengirimkan kemungkinan pengamatan yang tinggi bahwa suatu entitas ada di sana. Kapsul tingkat yang lebih tinggi mengabaikan *outlier*, berkonsentrasi pada *cluster*. Hal ini mirip dengan transformasi *Hough*, *RHT* (*Randomized Hough Transform*) dan *RANSAC* (*Random Sample Consensus*) dari pemrosesan gambar digital klasik. (Sabour dkk, 2017).

Menurut P. Lalit Shah dkk (2018) Jaringan kapsul meningkatkan kinerja CNN dengan memaksakan struktur pada CNN dan mengelompokkan neuron ke dalam kapsul. Kapsul adalah kumpulan neuron yang keluarannya memberi kita sifat berbeda dari fitur serupa. Outputnya adalah matriks 4x4 yang merupakan vektor 16 dimensi, di mana entri dalam matriks(atau vektor) mewakili informasi seperti koordinat x dan y dari fitur, skala dan ketebalan, kemiringan lokal, lebar dan transisi, dan lainnya *characteristics*. In pengenalan digit, digit diwakili oleh lapisan kapsul. Satu kapsul di lapisan keluaran sesuai dengan satu digit. Setiap entri dalam matriks 4x4 berbeda dan mewakili properti digit seperti nampak dalam gambar bawah ini.

| | |
|-----------------------|-----------------------|
| Scale and thickness | 6 6 6 6 6 6 6 6 6 6 6 |
| Localized part | 6 6 6 6 6 6 6 6 6 6 6 |
| Stroke thickness | 5 5 5 5 5 5 5 5 5 5 5 |
| Localized skew | 4 4 4 4 4 4 4 4 4 4 4 |
| Width and translation | 3 3 3 3 3 3 3 3 3 3 3 |
| Localized part | 2 2 2 2 2 2 2 2 2 2 2 |

Gambar 2.1: Informasi Spasial Dari Angka-Angka.

Sumber : (P. Lalit Shah dkk, 2018)

Gambar di atas menggambarkan informasi spasial tentang angka yang disimpan di setiap entri seperti bagian yang terlokalisasi, skala dan ketebalan, lebar dan terjemahan, kemiringan dan bagian yang terlokalisasi, dll. Informasi ini menentukan apa yang terjadi pada angka ketika ada perubahan dalam satu dimensi keluaran kapsul.

10

2.1.3. Perhitungan Caps Neural Network

Menurut Riti, dkk (2023) *neural network* dibagi berdasarkan *layer-layer* yaitu *input layer*, *hidden layer* dan *output layer*. Setiap *node* pada masing-masing layer memiliki suatu *error rate*, yang akan digunakan untuk proses *training*. *Neural network* dengan *layer-layer* memiliki konsep kerja sebagai berikut : *input layer* menunggu *user* memasukan *input* ke masing-masing *node* nya, setelah masing-masing *node* di *input layer* memperoleh data yang dibutuhkan maka akan dikalikan dengan *weight*-nya menghasilkan sum (jumlah) atau yang lebih dikenal dengan akumulator dengan rumus $NET = O_1 W_1 + O_2 W_2 + \dots + O_n W_n = \sum O_i W_i$, lalu akumulator tersebut akan dimasukkan kedalam Fungsi Aktivasi yang digunakan, rumusnya adalah $OUT = F (NET)$, untuk *Backpropagation* umumnya menggunakan fungsi *sigmoid biner* dan fungsi *sigmoid bipolar*.

2.2. Confusion Matrix

Menurut Riti, dkk (2023) Dalam *Confusion Matrix* terdapat empat istilah sebagai representasi hasil dari proses klasifikasi, keempat istilah itu *True Positive* (TP), *True Negative* (TN), *False Positive* (FP) dan *False Negative* (FN). Nilai *True Negative* (TN) merupakan jumlah data negatif yang terdeteksi dengan benar oleh sistem, sedangkan *False Positive* (FP) merupakan data positif namun terdeteksi sebagai data negatif/terklasifikasi salah oleh sistem. Sementara itu, *True Positive* (TP) merupakan jumlah data positif yang terklasifikasi dengan benar oleh sistem, *False Negative* (FN) merupakan jumlah data negatif namun terklasifikasi sebagai data positif/terdeteksi salah oleh sistem.

Berdasarkan nilai *True Negative* (TN), *False Positive* (FP), *False Negative* (FN) dan *True Positive* dapat diperoleh nilai akurasi, presisi dan *recall*. Nilai akurasi menggambarkan seberapa akurat sistem dapat mengklasifikasi data dengan benar, nilai presisi menggambarkan jumlah data kategori positif yang diklasifikasi dengan benar dibagi dengan total data yang diklasifikasi positif. Sementara *recall* menunjukkan berapa persen data kategori positif yang terklasifikasi dengan benar oleh sistem. Dengan rumus :

11

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN}$$

$$\frac{10000 + 10000}{10000 + 10000 + 10000 + 10000} = \frac{20000}{40000}$$

$$\frac{10000}{10000 + 10000} = \frac{10000}{20000}$$

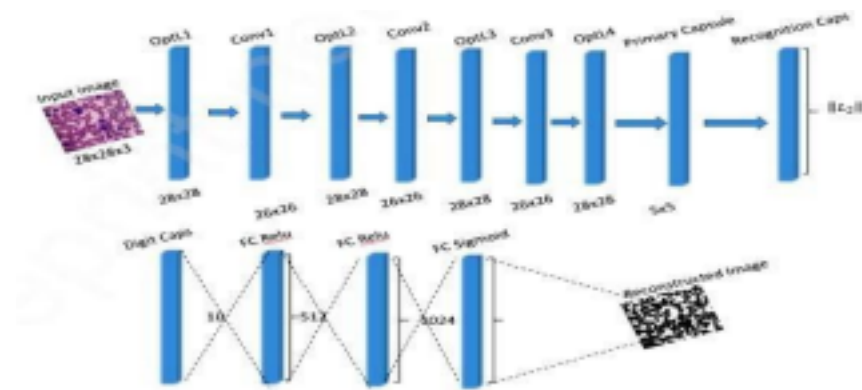
$$\text{Specificity} = \frac{TN}{TN + FP}$$

$$F1 = 2 \cdot \frac{Precision \cdot Recall}{Precision + Recall}$$

$$\text{FPR} = \frac{FP}{FP + TN}$$

2.2.1. Metode Klasifikasi Sel Darah menggunakan Jaringan Kapsul Arsitektur klasifikasi sel darah dikemukakan oleh Yaw Afriyie dkk (2022) dalam jurnalnya dengan algoritma yang dimodifikasi untuk mendeteksi dan mengklasifikasikan sel darah pada manusia, dan kegunaan *Caps Nets* dalam

klasifikasi sel darah. Setelah mengklasifikasikan sel darah menggunakan *Caps Nets*, jaringan *decoder* digunakan dalam rekonstruksi gambar. Metode yang digunakan Yaw Afriyie dkk mengadopsi perutean dinamis (Sabour dkk., 2017) dan menyajikan fungsi *squash* yang dioptimalkan dan algoritma penguat yang didasarkan pada perutean dinamis seperti yang nampak dalam gambar 2.2.



Gambar 2.2 : Model *Caps Nets* yang dioptimalkan
Sumber : Yaw Afriyie dkk (2022)

12

Pada gambar 2.2 diatas terdiri dari kapsul yang menggunakan tiga lapisan konvolusional, dan empat lapisan yang diperkuat sehingga memberikan fitur yang lebih saling memperkuat, hal itu dipetakan ke dalam kapsul utama. Gambar diubah ukurannya menjadi 28 x 28 untuk mendeteksi dan mengklasifikasikan sel darah manusia seperti yang terlihat dalam arsitektur. Karena setiap sel darah dapat membawa informasi yang lebih bermakna, Yaw Afriyie dkk memperkenalkan lapisan penguat untuk mengekstrak fitur yang lebih relevan.

Gambar masukan dipasok ke lapisan yang dimodifikasi dengan kernel 3 x 3 untuk menghasilkan peta fitur 28 x 28. Oleh karena itu, *Conv1* menerima masukan dari lapisan termodifikasi 1 yang memiliki 96, 26 x 26 kernel dan Relu dengan langkah 1. Ini menghasilkan keluaran berdimensi 96, 26 x 26 yang kemudian disuplai ke lapisan termodifikasi kedua. *Output* dari yang kedua lapisan yang dimodifikasi menghasilkan keluaran 96,24 x 24 yang selanjutnya dipasok ke *Conv2* dengan dimensi 128,24 x 24 yang berfungsi sebagai masukan ke lapisan yang dimodifikasi berikutnya.

Terdapat blok pra-pemrosesan dan lapisan normalisasi batch di antara lapisan yang dimodifikasi dan lapisan konvolusional. Untuk menghasilkan peta aktivasi baru, dilakukan dengan menjalankan perkalian berdasarkan pixel dalam blok pra pemrosesan dengan mengalikan setiap pixel peta aktivasi dengan angka yang berkisar antara 30 hingga 50.

Normalisasi batch (De dll., 2016; Yaw Afriyie dkk, 2022) diperkenalkan dalam arsitektur yang menghubungkan ke lapisan yang terhubung

sepenuhnya. Kapsul primer terdiri dari 5 komponen kapsul dengan dimensi lima. Ada empat kapsul di lapisan kapsul kelas (tergantung pada jumlah kategori dalam kumpulan data). Jaringan dekoder berisi tiga lapisan yang terhubung penuh (FC) dengan 512, 1024, dan 25600 *neuro* (yang juga bergantung pada kumpulan data) di lapisan pertama, kedua dan ketiga.

2.3. *Datasets*

Datasets terdiri dari dua kelas berbeda dari gambar apusan darah tipis eritrosit di mana satu kelas terinfeksi malaria dan yang lainnya sehat. Sampel-sampel ini dikumpulkan dari repositori *National Institutes of Health (NIH)* di mana data ini

13

tersedia untuk umum untuk penelitian (R. Sivaramakrishnan, 2018). Menurut Madhu dkk. (2020) dalam jurnalnya mengemukakan bahwa terdapat 27.558 gambar secara kolektif untuk kedua kelas tanpa ketidakseimbangan kelas dalam data. Sampel-sampel ini didistribusikan dengan resolusi pixel varian dari

gambar ke gambar. Jadi, sampel dibentuk ulang secara isotop menjadi resolusi 128×128 dan dengan tiga saluran (RGB) agar sesuai dengan jaringan saraf yang dirancang.

2.4. *Malaria*

Malaria adalah suatu bentuk penyakit yang disebabkan oleh infeksi yang disebabkan oleh parasit. Penularan parasit ini terjadi melalui gigitan nyamuk, khususnya spesies *Anopheles*. Parasit menginfeksi manusia melalui hati sebelum bersiap memakan sel darah merah, seperti yang diungkapkan oleh Kementerian Kesehatan RI dalam yankes.kemkes.go.id (2022) di daerah tropis, penyakit ini banyak ditemukan. Parasit bersel tunggal yang menular dari keluarga *Plasmodium*, *Haemosporida*, dan *Protozoa* menyebabkan malaria. *Plasmodium falciparum*, *Plasmodium malariae*, *Plasmodium vivax*, dan *Plasmodium ovale* merupakan empat spesies *Plasmodium* yang dapat menginfeksi manusia. *Plasmodium* hadir dalam berbagai jenis, masing-masing memiliki gejala dan pilihan pengobatan yang unik. Menurut Julia Fitriany dan Ahmad Sabiq (2018) gejala malaria ada dua kategori, yaitu gejala ringan dan gejala berat.

1. Gejala Ringan (Malaria Tanpa Komplikasi)

Gejala yang dialami oleh penderita malaria tingkat sedang ternyata sangat menyakitkan. Malaria umumnya bermanifestasi dengan suhu

tinggi dan menggigil, namun bisa juga menimbulkan gejala lain seperti sakit kepala, mual, muntah, diare, dan nyeri sendi atau otot. Lokasi asal parasit dan sistem kekebalan tubuh pasien berperan dalam timbulnya gejala. Demam adalah gejala utama malaria, infeksi yang disebabkan oleh parasit *Plasmodium*. Terjadinya demam diduga berhubungan dengan skizogoni, dampak *glicosil fosfatidilinositol* (GPI), produksi sitokin atau racun lain, atau pecahnya merozoit atau skizon. Parasitemia

14

dapat menyebabkan berbagai gejala, termasuk demam, yang mungkin tidak ada pada kasus tertentu (misalnya di daerah hiperendemik). 2.

Gejala Malaria Berat (Malaria Dengan Komplikasi)

Jika pemeriksaan laboratorium, seperti Persiapan Darah Tepi atau *Rapid Diagnostic Test* (RDT), menemukan parasit malaria dalam darah Menurut Julia Fitriany dan Ahmad Sabiq (2018) gejala malaria ada dua kategori, yaitu gejala ringan dan gejala berat.

3. Gejala Ringan (Malaria Tanpa Komplikasi)

Gejala yang dialami oleh penderita malaria tingkat sedang ternyata sangat menyakitkan. Malaria umumnya bermanifestasi dengan suhu tinggi dan menggigil, namun bisa juga menimbulkan gejala lain seperti sakit kepala, mual, muntah, diare, dan nyeri sendi atau otot. Lokasi asal parasit dan sistem kekebalan tubuh pasien berperan dalam timbulnya gejala.

Demam adalah gejala utama malaria, infeksi yang disebabkan oleh parasit *Plasmodium*. Terjadinya demam diduga berhubungan dengan skizogoni, dampak *glicosil fosfatidilinositol* (GPI), produksi sitokin atau racun lain, atau pecahnya merozoit atau skizon. Parasitemia dapat menyebabkan berbagai gejala, termasuk demam, yang mungkin tidak ada pada kasus tertentu (misalnya di daerah

hiperendemik). 4. Gejala Malaria Berat (Malaria Dengan Komplikasi)

Jika pemeriksaan laboratorium, seperti Persiapan Darah Tepi atau *Rapid Diagnostic Test* (RDT), menemukan parasit malaria dalam darah pasien dan ia juga mengalami setidaknya satu dari gejala atau akibat berikut, maka pasien tersebut dianggap menderita penyakit malaria:

- 1) Gangguan kesadaran dengan derajat yang berbeda-beda (dari koma hingga kehilangan kesadaran yang lebih ringan dengan

gejala termasuk delirium, ucapan salah, tidur terus menerus, keheningan, dan perubahan perilaku sebagai contoh).

- 2) Kelemahan umum; tidak bisa duduk atau berdiri.
- 3) Kejang.
- 4) Panas.

15

- 5) Rona kekuningan pada mata atau badan.
- 6) Indikator dehidrasi, seperti mata berair, kulit kendur, bibir kering, dan penurunan produksi urin.
- 7) Pendarahan pada hidung, gusi, atau saluran cerna.
- 8) Sesak napas atau napas cepat.
- 9) Terus-menerus muntah dan tidak bisa mengonsumsi makanan atau cairan apa pun.
- 10) Warna urin dapat berkisar dari teh tua hingga hampir hitam.
- 11) Keluaran urin sedikit atau tidak ada sama sekali.
- 12) Tangan yang warnanya sangat pucat dan kadar hemoglobin di bawah 5 g% menunjukkan anemia.

2.4.1. Klasifikasi Parasit Malaria

Plasmodium falciparum, *Plasmodium vivax*, *Plasmodium ovale*, *Plasmodium malariae*, dan *Plasmodium knowlesi* adalah lima spesies parasit yang menyebabkan penyakit yang disebut malaria, sebagaimana dinyatakan oleh *World Health Organization* (WHO).

a. *Plasmodium Falciparum*

Bentuk *Plasmodium falciparum* yang tumbuh paling cepat serta menyerang sel darah merah dan menghambat aliran darah, yang dapat menyebabkan kelumpuhan otak (cerebral palsy) dan anemia, menjadikannya jenis yang paling berbahaya. Malaria jenis ini tumbuh subur di iklim hangat dan lembap dan paling umum terjadi di beberapa negara, termasuk Indonesia dan Afrika.

b. *Plasmodium Vivax*

Plasmodium ini dapat ditemukan di seluruh dunia baik di daerah tropis maupun subtropis. Siklus seksual yang ditopang oleh sel darah merah terjadi setiap 48 jam. Menimbulkan sedikit kasus demam tertian yang terjadi setiap tiga hari. Hati manusia berpotensi menjadi inang "*hipnozoit*" bagi parasit ini, yang berarti ia mungkin tidak aktif selama berbulan-bulan atau bahkan bertahun-tahun sebelum muncul kembali.

c. *Plasmodium Ovale*

Plasmodium ovale yang sangat mirip dengan *Plasmodium vivax*, adalah parasit malaria yang umum di Afrika, khususnya di Afrika Barat dan kepulauan Pasifik Barat. Pada manusia, parasit penyebab malaria *ovale* juga dikenal sebagai malaria tertiana *ovale* jinak dapat tertidur di hati.

d. *Plasmodium Malariae*

Dapat menyebabkan malaria yang disebabkan oleh *Plasmodium* atau *Quartane*. Demam setiap empat hari disebabkan oleh siklus 72 jam pada sel darah merah.

e. *Plasmodium Knowlesi*

Parasit baru ini hanya dapat ditemukan di Asia Tenggara dan disebarkan melalui babi dan monyet yang terinfeksi, baik yang berekor besar maupun kecil. Siklus pengembangannya bisa sangat parah dan berulang hanya dalam 24 jam. *Plasmodium falciparum* dan

Plasmodium malariae dapat disalah artikan sebagai *P.knowlesi*. Jika lebih dari satu jenis plasmodium menginfeksi seorang pasien, kondisi ini disebut infeksi campuran. Mayoritas kasus melibatkan kombinasi *Plasmodium falciparum* dan *Plasmodium vivax* atau *malariae*.

2.4.2. Siklus Hidup Malaria

Proses siklus hidup parasit *Plasmodium* terjadi di dalam tubuh nyamuk *Anopheles* betina dan juga manusia. Dalam hal reproduksi, *Plasmodium* memiliki dua siklus hidup yang berbeda: siklus *sporogoni*, yang merupakan tahap seksual, dan tahap *skizogoni*, yang merupakan tahap aseksual.

- a. Siklus Hidup *Plasmodium* di Tubuh Manusia (Siklus Aseksual) Sel-sel pada hati akan terinfeksi oleh parasit yang masuk ke dalam tubuh melalui gigitan nyamuk *Anopheles* (sporozoit) yang terinfeksi.

Parasit ini kemudian akan melakukan reproduksi aseksual hingga berkembang menjadi skizon. Skizon akan pecah dan menghasilkan *merozoit* dalam jumlah besar, seringkali antara 2.000 dan 40.000

tergantungan spesiesnya. Menjadi "*merozoit*" yang matang mungkin

memerlukan waktu sepuluh hingga empat belas hari hingga banyak siklus (siklus ekso-eritrositik).

Setelah itu, merozoit akan masuk ke aliran darah dan menginfeksi sel darah merah sebelum menyebar lebih jauh. Ada beberapa parasit yang tetap tidak aktif di hati dan dapat aktif kembali. Parasit ini terlihat pada *P. vivax* dan *P. ovale*. Tidak semuanya menyebar ke aliran darah. Karena adanya vakuola yang terdapat pada sel parasit, merozoit yang menginfeksi sel darah merah pada akhirnya akan berevolusi menjadi parasit yang berbentuk cincin. Hal ini karena sel inti terletak di perbatasan trofozoit. Ketika trofozoit matang, mereka menjadi lebih besar, sehingga bentuk cincin menjadi lebih mudah terlihat. Setelah itu, *trofozoit* bereplikasi secara aseksual dengan membagi nukleus menjadi *skizon*, yang berkisar antara sepuluh hingga tiga puluh inti, bergantung pada spesies parasitnya.

Setelah selesainya siklus *eritrositik*, *skizon* dewasa akan meledak, melepaskan sejumlah besar merozoit muda yang akan menginfeksi lebih banyak sel darah merah. Siklus replikasi bertanggung jawab atas pecahnya dan hancurnya sejumlah besar eritrosit. Manifestasi gejala klinis disebabkan oleh replikasi dan penghancuran yang berulang-ulang. Rentang waktu antara gigitan nyamuk yang terinfeksi dan timbulnya gejala klinis disebut masa inkubasi intrinsik.

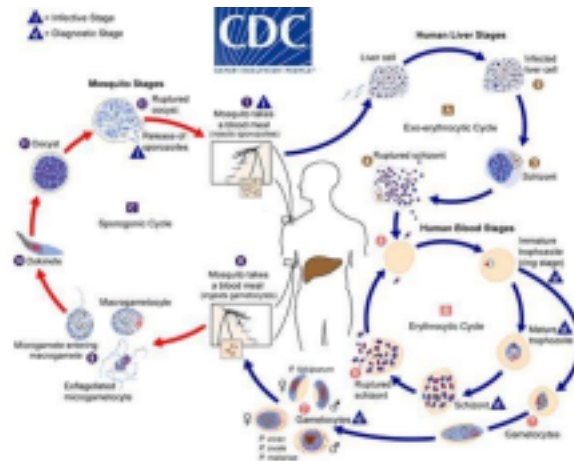
Beberapa *trofozoit* mengalami beberapa putaran replikasi, yang pada akhirnya menghasilkan perkembangan gamet jantan (*mikrogametosit*) dan gamet betina (*makrogamet*). Ketika nyamuk menghisap darah manusia yang terinfeksi parasit, maka parasit tersebut akan berkembang di dalam tubuh serangga tersebut. Ini adalah tahap di mana parasit akan tertular oleh nyamuk.

18

b. Siklus Hidup Plasmodium di Tubuh Nyamuk (Pembiasaan Seksual/ *Sporogoni*)

Baik *mikrogamet* maupun *makrogamet* dihirup dari darah manusia yang terinfeksi. Perkawinan silang akan terjadi antara jantan (*mikrogamet*) dan betina (*makrogamet*) untuk menghasilkan zigot. Zigot kemudian akan matang dan memanjang menjadi ookinet, yang selanjutnya akan masuk ke dinding lambung (usus tengah) dan akhirnya menjadi *ookista*. *Ookista* yang telah mencapai

kematangan akan pecah sehingga menghasilkan ribuan sporozoit baru. *Sporozoit* ini kemudian akan menyebar ke seluruh tubuh nyamuk, termasuk kelenjar ludah. Proses perkembangan dari zigot menjadi *sporozoit* memerlukan waktu antara 12 hingga 14 hari, yang disebut dengan waktu *inkubasi ekstrinsik*.



Gambar 2.3 Siklus Hidup Plasmodium
Sumber : cdc.gov

Berikut adalah tabel yang menunjukkan berapa lama siklus *ekso eritrositik* dan *eritrositik* untuk masing-masing spesies plasmodium.

Tabel 2.1 Lama Siklus *Ekso-Eritrositik* dan *Eritrositik* Pada Plasmodium

| Spesies | Siklus <i>Ekso Eritrositik</i> (hari) | Siklus <i>Eritrositik</i> (Jam) | Jumlah <i>Merozoit</i> dalam <i>Shcizon</i> |
|----------------------|---------------------------------------|---------------------------------|---|
| <i>P. falciparum</i> | 5-7 | 48 | 40.000 |
| <i>P. vivax</i> | 6-8 | 48 | 10.000 |
| <i>P. ovale</i> | 9 | 48 | 15.000 |
| <i>P. malariae</i> | 14-16 | 72 | 2000 |

2.5. Aplikasi

Setyawan dan Munari (2020) menyatakan bahwa aplikasi adalah jenis perangkat lunak yang secara langsung memanfaatkan kemampuan komputer untuk menjalankan tugas yang diinginkan pengguna. Aplikasi dikembangkan untuk mempermudah pekerjaan, baik itu penerapan, penggunaan, maupun penambahan data.

Aplikasi adalah program yang menyediakan tujuan tertentu, seperti mengedit dokumen, mengontrol Windows dan game, dll. (Rachmad Hakim S., 2018). Misalnya, aplikasi penggajian, aplikasi aset tetap, dan sebagainya merupakan contoh aplikasi yang didefinisikan oleh Harip Santoso

(2017) sebagai kelompok *file* yang berupaya menjalankan fungsi terkait tertentu. Aplikasi berarti "implementasi suatu aplikasi untuk digunakan", dari situlah istilah "aplikasi" berasal.

Aplikasi seperti yang dijelaskan oleh Dhanta dan dikutip oleh Azhar (2019), merupakan program yang dikembangkan oleh perusahaan komputer untuk melakukan aktivitas tertentu. Contoh program tersebut termasuk *Microsoft Word dan Excel*. "Aplikasi" berarti implementasi suatu aplikasi untuk digunakan, dari situlah istilah aplikasi berasal. Istilah aplikasi mengacu pada proses mentransfer informasi dari satu bentuk ke bentuk lain, baik itu data, masalah, atau pekerjaan, untuk menerapkan perubahan pada objek atau masalah tersebut dengan cara baru dengan tetap menjaga kualitas esensialnya.

2.6. Android

Menurut Syafaat dalam (Ariyanto, 2018), *Android* adalah seperangkat aplikasi, *middleware*, dan sistem informasi yang dirancang untuk digunakan pada perangkat seluler.

Menurut Ii dalam (Ariyanto, 2018), *Android* merupakan OS berbasis *Unix* untuk komputer tablet layar sentuh dan telepon seluler. Namun Android menjadi platform yang berinovasi dengan pesat seiring perkembangannya. Google, kekuatan pendorong di belakangnya, merupakan bagian integral dari hal ini. *Android* dikembangkan menjadi platform oleh *Google* setelah perusahaan membelinya.

20

Google Inc. adalah pengembang *Android*. *Open source* dan dibangun pada platform *Linux* adalah *Android*. Tujuan awal dari sistem operasi *Android* adalah untuk memberi daya pada perangkat seluler yang mendukung layar sentuh. *Gadget* tambahan yang menjalankan OS *Android*, termasuk kamera digital, jam tangan pintar, sistem navigasi, TV, dan kacamata pintar, kini sedang dalam pengembangan.

Karena *Android* bersifat *open source*, hal ini menarik minat para pengembang aplikasi yang ingin membuat aplikasi untuk smartphone yang menggunakan sistem operasi Android. *Google Play Store* adalah toko aplikasi yang kini menawarkan lebih dari satu juta aplikasi untuk diunduh pengguna *Android*. Mengingat hal di atas, dapat dikatakan bahwa *Android* adalah salah satu dari banyak sistem operasi yang sedang dikembangkan saat ini yang berbasis *Linux*.

2.6.1. Android Studio

Firly (2018) menjelaskan bahwa *Android Studio* dirancang untuk memfasilitasi pembuatan sistem operasi *Google Android*. Ini adalah IDE (*Integrated Development Environment*) resmi, atau lingkungan pengembangan terintegrasi. Dasar dari aplikasi ini adalah *IntelliJ IDEA JetBrains*. Sebagai alternatif dari *Eclipse*, atau ADT, *Android Studio* adalah *Integrated Development Environment* (IDE) utama untuk membuat aplikasi *Android* asli.

Android Studio IDE dibangun di atas *IntelliJ IDEA* dan dirancang khusus untuk membuat aplikasi *Android*. Ketika membuat aplikasi *Android*, *Android Studio* memiliki kemampuan tambahan yang dapat membantu Anda menjadi lebih produktif, seperti:

- a. Sistem pembuatan versi serbaguna yang dibangun dengan *Gradle*.
- b. Emulator yang efisien dan penuh fitur.
- c. Satu tempat untuk membuat kode untuk semua ponsel dan *tablet Android*.
- d. *Instant Run*, yang memungkinkan Anda memperbarui aplikasi yang ada tanpa membuat APK.
- e. Menggunakan integrasi *GitHub* dan templat kode untuk membangun fitur aplikasi bersama dan mengimpor sampel kode.

21

- f. Sumber daya pengujian yang komprehensif.
- g. Alat untuk linting yang memperbaiki masalah kecepatan, kegunaan, dan kompatibilitas antar versi.
- h. selain *NDK* dan *C++*.
- i. Integrasi *Google Cloud Messaging* dan *App Engine* menjadi mudah dengan dukungan bawaan untuk *Google Cloud Platform*.

2.6.2. *Android SDK*

Yuniar Supardi (2015) menyatakan bahwa *Android SDK* adalah seperangkat alat yang digunakan pengembang untuk membangun aplikasi untuk *platform Android* di Java. *Android Software Development Kit* (SDK) adalah seperangkat alat yang dapat digunakan pengembang untuk membuat aplikasi yang berjalan pada *platform Android*. Serangkaian alat lengkap untuk mengembangkan aplikasi disertakan dengan *Android SDK*. Termasuk dalam kit pengembangan perangkat lunak (SDK) *Android* adalah *debugger*, perpustakaan, emulator untuk perangkat seluler, dokumentasi, contoh kode, dan tutorial.

Android Software Development Kit (SDK) adalah API (*Application Programming Interface*) yang Safaat H. (2011) nyatakan penting untuk memulai pengembangan aplikasi Java pada *platform Android*. Sistem operasi

seluler Google, middleware, dan aplikasi penting semuanya merupakan bagian dari rangkaian perangkat lunak Android. Saat ini, *Android Software Development Kit* (SDK) ditawarkan sebagai seperangkat alat dan API untuk mulai membuat aplikasi berbasis Java untuk *platform Android*.

2.6.3. *Android JDK*

Java adalah teknologi yang diperkenalkan Sun Microsystems pada pertengahan tahun 1990an (DeCoster (2012)). *Java* adalah kumpulan teknologi yang dikembangkan oleh Sun yang memungkinkan programmer membangun dan menjalankan aplikasi pada komputer lokal dan jarak jauh.

Java Development Kit (JDK) *Sun Microsystems* adalah *compiler* dan *interpreter* yang memungkinkan pengguna membuat program Java. Instalasi *Android SDK* tidak dapat dimulai kecuali *Java Development Kit* (JDK) diinstal pada PC.

22

2.6.4. *Android Development Tools*

Android Development Tools, atau yang lebih sering dikenal dengan *Eclipse Plugins*, merupakan jembatan antara lingkungan pengembangan terintegrasi (IDE) *Eclipse* dan kit pengembangan perangkat lunak (SDK) Android, seperti yang diungkapkan oleh Yuniar Supardi (2015). Dengan ADT, *Eclipse User* dapat membangun aplikasi Android, merancang antarmuka pengguna, mengintegrasikan komponen yang dibangun pada kerangka API Android, melakukan debug aplikasi, dan menggabungkan aplikasi Android.

2.6.5. *Android Virtual Device*

AVD adalah platform untuk mengembangkan dan menguji aplikasi *Android*; itu beroperasi pada mesin virtual, (Safaat H., 2011). *Android Virtual Device Driver* (AVD) adalah pengaturan yang memungkinkan kita menjalankan perangkat *Android* berdasarkan model yang dipilih, antara lain *Android 1.5* atau *2.2*.

2.7 Algoritma

Algoritma merupakan cara penyelesaian masalah dengan instruksi atau kegiatan yang disusun secara teratur dan sistematis. Manusia dapat menciptakan karya yang semakin kompleks dan canggih sebagai hasil dari kemajuan ilmu pengetahuan dan teknologi. Meskipun komputer pada umumnya dapat mengungguli manusia dalam hal kecepatan komputasi,

mereka masih perlu diajari cara menyelesaikan masalah oleh manusia menggunakan algoritma yang telah ditentukan sebelumnya. Selain penggunaannya dalam pemecahan masalah berbasis komputer, algoritma mempunyai aplikasi praktis dalam penyelesaian masalah-masalah umum yang melibatkan serangkaian prosedur. Kita bisa mendapatkan beberapa definisi "algoritma" dari berbagai tempat, dan itu akan membantu kita memahami apa itu algoritma. Pada tahun 825 M, dalam karyanya *Al-Jabr Wa-al Muqabla*, Abu Ja'far Muhammad Ibnu Musa Al Khawarizmi mula-mula menggunakan istilah algoritmik dan ritmis, yang akhirnya menjadi kata algoritma. Algoritma didefinisikan dalam pemrograman komputer sebagai metode yang

23

menggunakan serangkaian langkah untuk menyelesaikan masalah secara sistematis dan terorganisir (Jando & Nani, 2018).

(Kani, 2020) menyatakan bahwa algoritma adalah upaya untuk memecahkan masalah dengan menyusun serangkaian tindakan secara sistematis dan logis untuk menghasilkan keluaran tertentu. Beberapa contoh dari berbagai macam algoritma adalah:

1. Algoritma Sekuensial (*Sequence Algorithm*)

Algoritma yang tahapannya diurutkan secara berurutan dari awal hingga akhir disebut *Sequence algorithm* atau algoritma sekuensial. Metode perebusan air adalah contoh yang baik dari jenis algoritma sekuensial ini. Mengikuti prosedur dalam urutan khusus ini sangatlah penting.

2. Algoritma Perulangan (*Looping Algorithm*)

Algoritma yang mengulangi dirinya sendiri atau menjalankan proses yang sama berulang kali disebut *Looping Algorithm*. Kita juga harus melakukan hal yang sama berulang kali untuk menyelesaikan tantangan yang kita hadapi.

3. Algoritma Percabangan atau Bersyarat (*Conditional Algorithm*)

Jika persyaratan tertentu terpenuhi, algoritma melanjutkan ke tahap berikutnya; jenis algoritma ini dikenal sebagai *Conditional Algorithm*.

2.8. Mobile

Kata “mobile” digunakan untuk mencirikan aplikasi komunikasi pada perangkat nirkabel yang kecil dan portable (Prakarsya, A., 2019). Perangkat seluler menggunakan lebih sedikit daya dibandingkan komputer desktop, dan

konsumen memilih perangkat kompak karena portabilitas dan kemudahan penggunaan saat bepergian.

Istilah "*mobile*" menggambarkan perangkat lunak yang dapat dijalankan pada perangkat elektronik portable, karena kata *mobile* berarti gerakan. Anda dapat melakukan apa saja dengan bantuan aplikasi seluler; mereka membuat berbelanja, belajar, bekerja dari rumah, menjelajahi web, dan banyak lagi menjadi lebih mudah.

24

2.9. *Python*

Salah satu bahasa pemrograman tersebut adalah *Python*, yang menawarkan struktur data tingkat tinggi dan interaktif serta berorientasi objek. *Python* adalah bahasa pemrograman interpretatif yang melayani berbagai tujuan dan menekankan pada keterbacaan kode, membuat sintaksisnya lebih mudah dipahami. Orang-orang mengklaim bahwa *Python* adalah bahasa yang kuat dengan sintaksis yang mudah dipahami. Dalam hal efisiensi kesederhanaan pengembangan, dan kompatibilitas sistem, bahasa pemrograman *Python* dibuat khusus untuk membuat hidup programmer lebih mudah. Scripting dan pengembangan aplikasi independen adalah dua dari banyak kegunaan *Python* (A. Nur Syahrudin dan T. Kurniawan, 2018).

Linux/Unix, Windows, Mac OS X, Java Virtual Machine, dan banyak lagi sistem lainnya yang kompatibel dengan bahasa pemrograman *Python*. *Python* hadir dengan banyak lisensi berbeda, masing-masing dengan nomor versinya sendiri. Namun secara teori, bahkan untuk penggunaan komersial, *Python* gratis untuk digunakan. Baik Definisi *Open Source* maupun Lisensi Publik Umum (GPL) kompatibel dengan lisensi *Python* (R. R. Saragih, 2019).

Karena tata bahasa *Python* lebih mirip dengan bahasa manusia daripada bahasa mesin, ini adalah bahasa pemrograman yang mudah dipelajari dan dipahami oleh pemula. *Python* merupakan bahasa pemrograman yang mirip dengan bahasa Inggris. Lebih sedikit baris kode berarti bahwa *Python*

lebih mudah dipahami oleh pendatang baru, terutama anak-anak.

2.10. *OpenCv*

Proyek *OpenCv* (*Open Source Computer Vision Library*) dimulai oleh *Intel* dan saat ini dikelola oleh *Willow Garage* dan *Itseez*. Tujuan utamanya adalah untuk memfasilitasi pemrosesan gambar dinamis secara *real-time*. Perangkat lunak *OpenCV* didistribusikan di bawah lisensi BSD yang lebih liberal, yang memberikan lebih banyak kebebasan dibandingkan GPL dan memungkinkan

penggunaan komersial tanpa batas tanpa memerlukan pengungkapan kode sumber. Antarmuka pengguna *OpenCV* kompatibel dengan beberapa platform dan bahasa, termasuk *Windows, Linux, Android, dan C++, C+, Python, dan*

25

Java. *OpenCV* memprioritaskan aplikasi *real-time* dan dibuat agar efisien secara komputasi.

2.11. Citra Digital

Berbagai bidang, mulai dariseni dan penglihatan manusia hingga astronomi dan teknik, berkontribusi pada penciptaan gambar, yang merupakan penggambaran objek dua dimensi di dunia yang terlihat. Menurut Hutahaean, Waluyo, dan

Rais (2019), itu adalah susunan dua dimensi berupa titik atau pixel berwarna. Citra digital dua dimensi dibuat dengan mengambil sampel pixel individual dari gambar analog dua dimensi yang berkesinambungan. Ada array dalam gambar digital yang berisi urutan bit tertentu yang mewakili nilai nyata dan kompleks. Pixel, (*picture element*), terletak pada perpotongan baris dan kolom dalam matriks dua dimensi $f(x,y)$ yang merepresentasikan citra digital. Matriks tersebut mempunyai M kolom dan N baris. Berikut representasi fungsi matematika dari suatu gambar $f(x,y)$ yang diperoleh dari rumus yang diberikan di atas:

$$f(x,y) \approx \begin{bmatrix} f(0,0) & f(0,1) & \dots & f(0,M-1) \\ f(1,0) & f(1,1) & \dots & f(1,M-1) \\ \vdots & \vdots & \vdots & \vdots \\ f(N-1,0) & f(N-1,1) & \dots & f(N-1,M-1) \end{bmatrix}$$

$$\begin{matrix} 0 \leq x \leq \\ M-1 \end{matrix}$$

$$\begin{matrix} 0 \leq y \leq \\ N-1 \end{matrix}$$

$$\begin{matrix} 0 \leq f(x,y) \\ \leq G-1 \end{matrix}$$

Dimana:

M = Jumlah pixel baris (*row*) pada *array* citra

N = Jumlah pixel kolom (*column*) pada *array* citra

G = Nilai skala keabuan (*gray level*)

26

Besarnya nilai M,N dan G pada umumnya merupakan perpangkatan dari dua.

$$M = 2^m; N = 2^n; G = 2^k$$

Dimana n, k adalah bilangan bulat positif. Skala abu-abu (grayscale) adalah interval $(0, G)$. Proses digitalisasi menentukan besar kecilnya G . Sebagai aturan umum, 0 (nol) menunjukkan intensitas hitam dan 1 (satu) menunjukkan intensitas putih. Dengan 28 bit per pixel, atau 256 tingkat derajat abu-abu, nilai G citra 8-bit adalah 256.

Mengubah citra analog menjadi citra *digital* disebut digitasi. Digitasi adalah transformasi benda nyata seperti gambar, teks, atau audio menjadi informasi digital yang dapat digunakan untuk penyimpanan dan pemrosesan. Dalam matriks dua dimensi, citra *digital* diterjemahkan ke dalam kisi-kisi elemen pixel. Saluran warna diwakili oleh nilai numerik yang ditetapkan untuk setiap pixel. Komputer sering kali mengurangi angka-angka di setiap angka dan menyimpannya secara berurutan untuk alasan pemrosesan atau kompresi.

2.11.1 Tipe-Tipe Citra Digital

Ada berbagai macam citra *digital* yaitu:

a. Citra Biner

Tidak adanya nilai derajat lain selain hitam dan putih menciptakan apa yang dikenal sebagai *binary image* atau citra biner. Nilai pixel objek adalah 1, sedangkan nilai pixel latar belakang adalah 0. Saat menampilkan gambar, pixel dengan nilai 0 berarti putih dan pixel dengan nilai 1 berarti hitam.

b. Citra Keabuan

Pada citra skala abu-abu (*grayscale*), nilai intensitas berkisar antara 0 (hitam) hingga 255 (putih) untuk setiap pixel.

c. Citra Warna

Setiap pixel dalam citra berwarna (*RGB*) berisi informasi tentang warna gambar. Format seperti *RGB*, *CMYK*, *HSV*, dan masih banyak lagi tersedia untuk gambar berwarna. Model warna RGB mencakup

27

berbagai macam warna yang diciptakan dengan menggabungkan merah, hijau, dan biru. Kisaran nilai untuk monitor komputer adalah dari 0 hingga 255.

2.11.2. Pengolahan Citra Digital

Bidang studi yang disebut *Digital Image Processing* (DIP) mengkaji beberapa

metode pengolahan citra digital. Representasi visual apa pun, baik diam (gambar) maupun bergerak (video), itulah yang dimaksud dengan istilah ini. Pada saat yang sama, "*digital*" di sini berarti komputer yang digunakan untuk memproses gambar. Pemrosesan data dua dimensi mencakup pemrosesan citra *digital* dalam arti yang lebih luas. Komputer digital hanya dapat memproses gambar yang direpresentasikan secara numerik dengan nilai diskrit. Digitalisasi gambar adalah proses mengubah fungsi lanjutan menjadi nilai diskrit. Meningkatkan kualitas gambar untuk memudahkan interpretasi manusia atau mesin adalah tujuan utama pemrosesan gambar.

Digitalisasi mengacu pada proses mengubah gambar analog menjadi gambar digital. Istilah "digitasi" mengacu pada transformasi materi analog (seperti gambar, potongan teks, atau suara) menjadi format elektronik untuk penyimpanan dan pemrosesan lebih lanjut. Pemrosesan data dua dimensi lebih luas disebut sebagai pemrosesan citra digital. Bidang studi yang dikenal dengan pengolahan citra *digital* berfokus pada berbagai metode pengolahan citra digital. Representasi visual apa pun, baik diam (gambar) atau bergerak (video), adalah maksud dari istilah ini. Penelitian ini memanfaatkan metode berikut untuk pengolahan citra *digital*.

a. *Resizing*

Resizing merupakan proses mengubah ukuran panjang dan lebar citra *digital* pada arah vertikal dan/atau *horizontal*

b. *Mean Substraction*

Mean subtraction digunakan sebagai sarana untuk menghindari masalah variasi pencahayaan pada foto masukan kumpulan data. Selama pelatihan, kami mengambil nilai pixel rata-rata dari kumpulan data pelatihan dan menghapusnya dari setiap gambar untuk mengelola fluktuasi intensitas dan normalisasi. Pada langkah ini, tiga

28

variabel disiapkan, yaitu: μ_R , μ_G , dan μ_B . Setelah itu, setelah gambar disiapkan melalui jaringan saat ini, prosedur untuk menghilangkan nilai rata-rata dari setiap *input channel* pada gambar masukan dijalankan:

$$R = (R - \mu_R) \quad G = (G - \mu_G) \quad B = (B - \mu_B)$$

c. *Scaling*

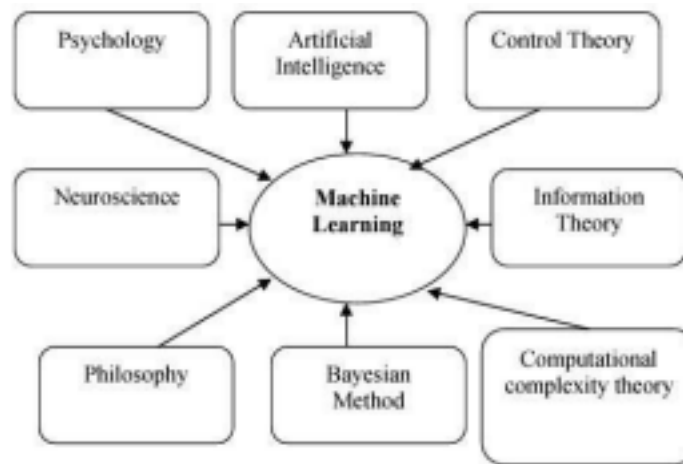
Agar jaringan *Machine Learning* dapat memproses gambar berdasarkan gambar masukan dan mengkoordinasikan *scaling* kotak pembatas yang sedang diproses pada masukan, tampilan, dan ukuran

gambar yang disimpan, perlu dilakukan penskalaan standar deviasi berdasarkan set pelatihan. Dalam proses normalisasi, faktor *scaling*, σ , disertakan.

$$R = (R - \mu_R) / \sigma \quad G = (G - \mu_G) / \sigma \quad B = (B - \mu_B) / \sigma$$

2.12. Machine Learning

Machine Learning (ML) sering dikenal sebagai pembelajaran mesin, adalah bidang interdisipliner yang diambil dari berbagai disiplin ilmu dan bidang penelitian. Model simulasi pembelajaran mesin (ML) memiliki banyak kesamaan dengan statistik komputasi, bidang yang terutama berkaitan dengan penggunaan komputer untuk menghasilkan prediksi. Selain itu, hal ini terkait dengan optimasi matematis, yang menjembatani kesenjangan antara statistik dan model, aplikasi, dan pengembangan kerangka kerja.



Gambar 2.4: Multi-disiplin *Machine Learning*
Sumber : Jafar Alzubi dkk (2018)

29

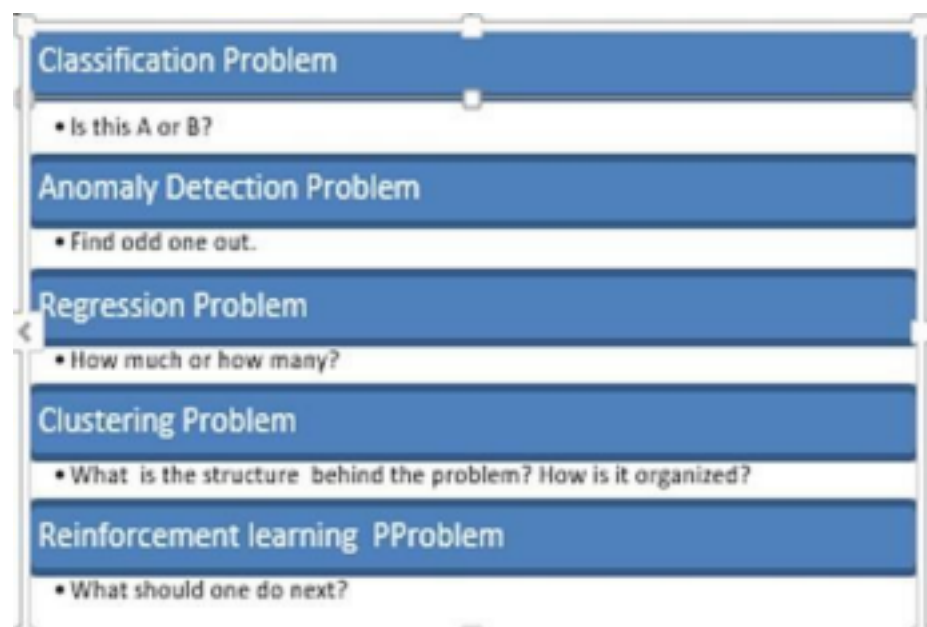
Masalah dunia nyata memiliki kompleksitas tinggi sehingga menjadikannya kandidat yang sangat baik untuk penerapan ML. Pembelajaran mesin memiliki banyak aplikasi potensial dalam bidang ilmu komputer, termasuk namun tidak terbatas pada: pemfilteran spam email, perdagangan saham online, diagnosis medis, prediksi lalu lintas, pengenalan karakter, rekomendasi produk, deteksi wajah dan bentuk, dan deteksi penipuan jaringan sosial.

Jafar Alzubi dkk (2018) menjelaskan bahwa *Machine Learning* berdasarkan pembelajaran dan pengalaman yang terus meningkat, penting untuk memungkinkan komputer melakukan aktivitas canggih tanpa interaksi manusia. Hal ini akan membantu komputer memahami kompleksitas situasi dan perlunya kemampuan beradaptasi. Secara umum tugas manusia dapat diklasifikasi menjadi dua sebagai berikut :

- a. Tugas yang dilakukan manusia sehari-hari, namun perhatian utamanya adalah melakukan tugas tersebut dengan sempurna dan melaksanakannya sesuai program yang ditentukan dengan baik.
Contoh: Memasak, Mengemudi, Pengenalan Ucapan.
- b. Tugas di luar Kemampuan Manusia: Kategori tugas lain yang dapat diselesaikan dengan pembelajaran mesin secara efektif adalah analisis kumpulan data yang besar dan kompleks seperti Penginderaan Jarak Jauh, Prakiraan cuaca, E-niaga, penelusuran Web, dll. Dengan jumlah data yang besar, hal tersebut menjadi sangat rumit bagi manusia untuk memprediksi data yang bermakna.

Hayashi dan Chikio dalam Jafar Alzubi dkk (2018) mengusulkan untuk mengklasifikasikan *data science* ke dalam salah satu dari lima bidang berbeda, seperti yang diilustrasikan pada gambar di bawah.

30



Gambar 2.5: Tipe Masalah Dalam *Data Science*
Sumber : Jafar Alzubi dkk (2018)

Machine Learning dalam penerapannya sangat tergantung pada jenis masalahnya, berbagai kategori dijelaskan di bawah ini: a. *Classification Problem* / Masalah Klasifikasi

Masalah klasifikasi adalah masalah yang keluarannya hanya dapat berupa salah satu dari sejumlah kelas keluaran apriori yang diketahui, seperti Ya/Tidak atau Benar/Salah. Jenis kategorisasi masalahnya mungkin berkisar dari biner hingga kelas jamak, bergantung pada jumlah kelas keluaran.

b. *Anomaly Detection Problem* / Masalah Deteksi Anomali Masalah yang menganalisis pola tertentu dan mendeteksi perubahan atau

anomali dalam pola tersebut termasuk dalam kategori ini. Misalnya, perusahaan kartu kredit menggunakan algoritma deteksi anomali untuk menemukan penyimpangan dari perilaku transaksi biasa klien mereka dan meningkatkan peringatan setiap kali ada transaksi yang tidak biasa. Masalah seperti itu berkaitan dengan menemukan pencilan.

c. Regression Problem / Masalah Regresi

Algoritma regresi digunakan untuk menangani masalah dengan keluaran kontinu dan numerik. Ini biasanya digunakan untuk masalah

31

yang berhubungan dengan pertanyaan seperti, 'berapa banyak' atau 'berapa banyak'.

d. Clustering Problem / Masalah Pengelompokan

Pengelompokan termasuk dalam kategori algoritma pembelajaran tanpa pengawasan. Algoritma ini mencoba mempelajari struktur di dalam data dan mencoba membuat cluster berdasarkan kemiripan struktur data. Kelas atau cluster yang berbeda kemudian diberi label. Algoritma, saat dilatih, menempatkan data baru yang tidak terlihat di salah satu *cluster*.

e. Reinforcement Problem / Masalah Penguatan

Algoritma penguatan digunakan ketika keputusan harus dibuat berdasarkan pengalaman belajar di masa lalu. Agen mesin mempelajari perilaku menggunakan jenis interaksi coba-coba dengan lingkungan yang terus berubah. Ini menyediakan cara untuk memprogram agen.

2.13. Penelitian Relevan

1. Pada penelitian sebelumnya menjelaskan bahwa penggunaan teknologi dapat membantu proses deteksi dini pada dunia kesehatan. Tujuan dari upaya teknologi ini adalah untuk mengidentifikasi kasus anemia dengan menganalisis gambar sel darah merah. Jaringan saraf tiruan yang disebut *self-organizing mapping* (SOM) mengurutkan pola masukan ke dalam kategori dan kemudian menetapkan keluaran jaringan ke kategori yang memiliki tingkat kemiripan tertinggi dengan masukan. Pada kasus anemia, kami ingin mengetahui seberapa akurat penggunaan SOM untuk menentukan ukuran sel darah merah. Pengelompokan adalah area di mana *self-organizing mapping* (SOM) unggul. Selain itu, dibandingkan cara lain, cara ini lebih mudah diterapkan

2. Pada penelitian yang dilakukan oleh Afrizal, metode pengolahan gambar diciptakan untuk mengotomatisasi diagnosis malaria pada film darah tipis Tujuan dari teknik klasifikasi gambar adalah untuk mengidentifikasi dan membedakan secara positif spesies parasit malaria pada apusan darah tipis. Penulis mempresentasikan hasil penelitiannya tentang sistem *computer vision* yang dapat mendeteksi malaria secara otomatis menggunakan *deep learning* pada dataset Rajaraman sebagai jawaban atas tantangan tersebut di atas. Dengan menggunakan data uji, Analisis Citra Medis mampu mencapai tingkat akurasi deteksi di atas 97%.(Zein, 2019).
3. Pada penelitian yang dilakukan oleh Daniel Martomanggolo (2023), kinerja berbagai desain jaringan dibandingkan dalam tugas klasifikasi gambar digital sel darah putih. Klasifikasi citra sel dilakukan dengan menggunakan metode pembelajaran transfer, yang menggunakan arsitektur jaringan terlatih. Empat arsitektur jaringan yang digunakan adalah *VGG-16*, *ResNet-50*, *AlexNet*, dan *GoogleNet*. Temuan penelitian ini menunjukkan bahwa keempat topologi jaringan yang diuji memiliki kinerja yang mengagumkan ketika mentransfer klasifikasi gambar sel darah putih berbasis pembelajaran. Jika dibandingkan dengan tiga arsitektur lainnya, ResNet-50 menghasilkan jumlah gambar yang dikategorikan dengan benar dalam jumlah terbesar. (Wonohadidjojo, 2021)

BAB 3

ANALISIS DAN PERANCANGAN SISTEM

Dalam melakukan perancangan sistem, ada tahapan-tahapan yang perlu dilakukan untuk memenuhi kebutuhan sistem, mulai dari tahapan analisis sistem sampai perancangan sistem. Analisis dilakukan untuk mengetahui

kebutuhan sistem, sehingga dapat menghasilkan rancangan sistem sesuai yang dibutuhkan.

3.1. Analisis Sistem

Penyelidikan ilmiah seperti ini dikenal dengan penelitian eksperimental. Definisi umum penelitian eksperimental adalah penelitian yang menggunakan sukarelawan manusia untuk menguji kemanjuran pengobatan potensial (Kasi, 2012). Karena sifat eksperimental dari objek yang diteliti dan penggunaan algoritma *Artificial Neural Network* bersama dengan metode *Component Analysis Principal* untuk analisis data di berbagai bidang atau kumpulan data multidimensi, khususnya grafik komputer, maka penelitian ini termasuk dalam kategori ini. kategori penelitian eksperimental. Agar sistem ini berfungsi sebagaimana mestinya, sejumlah persyaratan yang melekat pada desainnya perlu dipenuhi. Berikut adalah persyaratan sistem:

1. Sistem dapat menginputkan atau memasukan gambar
2. Sistem melakukan Analisis, atau proses terhadap gambar yang dimasukan
3. Sistem memberikan *result* berupa informasi yang dibutuhkan oleh pengguna.

34

3.2. Analisis Data

Android Studio dan Python 3.11 adalah *software* utama yang digunakan dalam penelitian ini. Untuk mengatasi masalah seperti ketidakseimbangan kelas data pada kumpulan data atau kekurangan data pelatihan, penelitian ini menerapkan teknik CAPS pada *Artificial Neural Network* (ANN) untuk analisis data. Sun, Shrivastava, Singh, dan Gupta (2011) menemukan bahwa jumlah data pelatihan memiliki hubungan logaritmik dengan kinerja *Artificial Neural Network* (ANN). Konsekuensinya, tingkat akurasi berbanding lurus dengan jumlah data yang diakses.

3.3. Rancangan Arsitektur Sistem

Rancangan arsitektur sistem menggambarkan alur kerja sistem dalam melakukan proses pengolahan citra untuk mendeteksi virus malaria pada darah manusia. Rancangan arsitektur sistem dapat dilihat pada gambar 3.1.

35



Gambar 3.1 Proses Klasifikasi Sel Darah

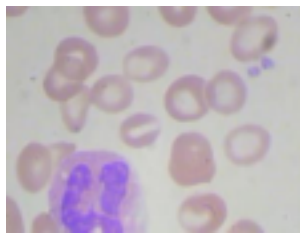
Pada saat aplikasi dimulai, aplikasi akan mendeteksi data *sample* yang disimpan, kemudian dari data tersebut sistem melakukan proses *machine learning*, pada proses sistem melakukan pendeteksian dan pencocokan gambar yang teridentifikasi terdapat virus malaria. Setelah proses tersebut dilakukan sistem akan memunculkan hasil *result* dari proses tersebut.

36

3.4. Data Penelitian

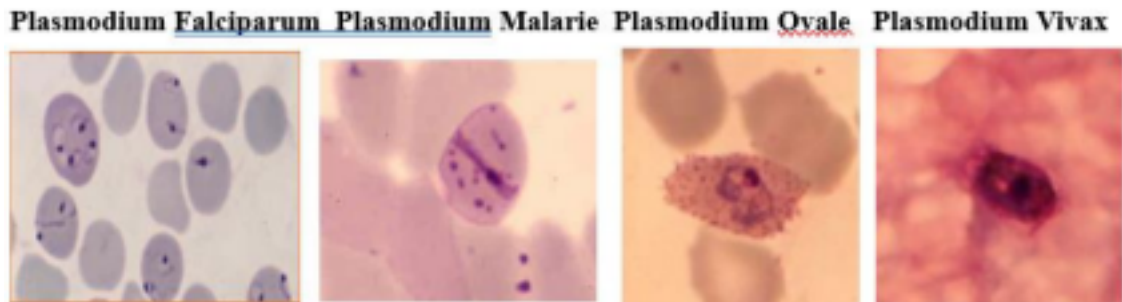
Data penelitian ini berasal dari *database* malaria di *National Institutes of Health* (NIH). Gambar di bawah menjelaskan koleksi malaria yang disediakan oleh *National Institutes of Health* (NIH).

Tidak terinfeksi



Gambar 3.2 Sebuah *Subset* dari Sample darah yang tidak terinfeksi Malaria yang Disediakan oleh *National Institute of Health* (NIH).

Terinfeksi



Gambar 3.3 Sebuah *Subset* dari Sample darah yang terinfeksi Malaria berdasarkan Plasmodium yang Disediakan oleh *National Institute of Health* (NIH).

Dataset yang penulis dapatkan dari *National Institute of Health* (NIH) kurang lebih ada sekitar dua puluh tujuh ribu data, yang didalamnya terdapat dataset yang terkena virus malaria, dan *dataset* yang tidak terkena virus malaria.

3.4.1. Data Sample

Kumpulan data yang disediakan diambil sampelnya, lalu datanya dipartisi. Kumpulan data dibagi menjadi dua bagian: 90% untuk *training* dan 10% untuk *test*. Pengambilan data *sample* dilakukan adalah sebagai salah satu metode yang dilakukan oleh laboratorium untuk mengetahui atau mendeteksi penyakit.

37

3.4.2. Data Augmentation

Data pelatihan disempurnakan menggunakan strategi *Augmentation* dalam penelitian ini dengan memodifikasi rentang *rotation range*, *zoom range* *width shift range*, *height shift range*. Kompilasi dan pelatihan model difasilitasi oleh beberapa teknik ini. Gambar 3.4 di bawah menunjukkan alur Data *Augmentation*.



Gambar 3.4 Proses Data *Augmentation*

3.5. Processing Data

Langkah *processing* data pada penelitian ini akan membagi kumpulan data pelatihan awal menjadi dua bagian yaitu 80% untuk pelatihan dan 20% untuk pengujian. Setelah itu, ia menskalakan data sedemikian rupa sehingga semua nilai berada dalam interval $[0,1]$ dan memformat ulang data tersebut sesuai dengan ekspektasi jaringan. Misalnya, data pelatihan awalnya disimpan sebagai array (60000, 28, 28) bertipe *uint8* dengan nilai berkisar antara 0 hingga 255. Selanjutnya diubah menjadi *array float32* (60000, 28*28) dengan nilai berkisar antara 0 hingga 1 Akhirnya, CNN memiliki celah dan diselesaikan dengan jaringan kapsul (*Capsnet*) yang baru dikembangkan.

Penelitian ini mewakili model diagnostik baru yang mengklasifikasikan citra darah malaria yang mengandung parasit/ terinfeksi dan citra darah yang sehat. Aplikasi dikembangkan untuk dapat mendeteksi dan mengklasifikasikan citra darah yang terinfeksi malaria dan yang tidak terinfeksi. Enam lapis tentang penumpukan mesin *Boltzmann* yang dibatasi (pra- pelatihan). Pengklasifikasian keberadaan malaria dengan gambar apusan darah dengan mencapai sensitivitas 0,976 dan spesifisitas 0,952. (Ross dkk., 2006; Madhu G., 2020)) mengidentifikasi keberadaan parasit dalam *eritrosit* dengan transformasi morfologi dan metode *thresholding*. Mereka mengekstrak fitur visual parasit

38

dengan mengklasifikasikan sel yang terinfeksi dan tidak terinfeksi melalui warna, tekstur, dll menggunakan jaringan saraf *feed-forward* dan mencapai spesifisitas 0,85 dan PPR 0,81. Selain itu, mereka mampu mengklasifikasikan dengan skor akurasi 0,733.



Gambar 3.5. Arsitektur *Capsule Network*
Sumber: Madhu G. (2020)

Gambar diatas mengilustrasikan arsitektur lengkap dari jaringan kapsul yang diusulkan. Awalnya data diproses sebelumnya dengan membentuk kembali gambar individual menjadi 3 resolusi 128x128 pixel yang disalurkan. Data ini dimasukkan ke dalam rangkaian jaringan konvolusi untuk ekstraksi fitur. Vektor fitur ini selanjutnya diteruskan ke lapisan Kapsul ke-1 (yaitu kapsul primer) dan kemudian melakukan perutean dinamis dengan kapsul sekunder untuk 2 iterasi. Vektor aktivasi yang dirutekan dengan hati-hati ini menghitung klasifikasi Norma L2 ke dalam kelas yang terinfeksi dan tidak terinfeksi.

CNN dapat mengklasifikasikan gambar noda darah yang mengandung malaria. Menurut hasil penelitian (Liang dkk., 2016) Inputnya adalah (50x50x3) gambar saluran RGB dan mencapai akurasi pelatihan 0,9971 dan akurasi pengujian 0,9823 dalam memeriksa gambar noda darah yang mengandung malaria. Gambar-gambar yang mengandung malaria ini dimasukkan ke dalam CNN berukuran 44x44x3. Akurasi rata-rata: 0,9731 paling besar dibandingkan model pembelajaran transfer lainnya.

Sivaramakrishnan dkk. (2018) menunjukkan pentingnya memvisualisasikan CNN, dilatih dengan data (ukuran: 44x44x3) yang berisi gambar noda darah tipis. Mereka mengembangkan model dengan kompleksitas

39

dan waktu komputasi yang lebih rendah dengan mencapai akurasi 0,9861. Sementara itu Gopakumar dkk. (2018) berfokus pada masalah penghitungan porsi parasit pada gambar (ukuran: 32x32x3) dengan segmentasi dengan bantuan CNN's. Selama prosedur segmentasi dua tingkat ini mencapai sensitivitas: 0,9706 dan spesifisitas: 0,9850. Dalam sebagian besar skenario, kinerja untuk model individual sangat mencengangkan tetapi resolusi pixel rendah tidak dapat digunakan dalam situasi dunia nyata.

3.6. Rancang Desain Aplikasi

Sebelum mengimplementasikan aplikasi ke tahap perancangan, maka diperlukan rancang desain aplikasi untuk membuat tampilan saat pembangunan aplikasi, berikut ini tampilan rancang desain aplikasi:

3.6.1. Tampilan Rancangan Dashboard

Gambar 3.6. Tampilan Rancangan *Dashboard*

Berikut ini keterangan mengenai tampilan rancangan desain pada *dashboard* dapat dijelaskan pada tabel dibawah ini :

40

Tabel 3.1. Keterangan Tampilan Rancangan *Dashboard*

| No | Nama | Keterangan |
|----|---------------|---|
| 1 | <i>Logo</i> | Untuk menampilkan logo aplikasi |
| 2 | <i>Output</i> | Untuk tempat menampilkan dataset citra darah |
| 3 | <i>Button</i> | Untuk menuju ke media penyimpanan media galeri pada android |

3.6.2. Tampilan Rancangan Galeri

Gambar 3.7. Tampilan Rancangan Galeri

Berikut ini keterangan mengenai tampilan rancangan desain pada galeri dapat dijelaskan pada tabel dibawah ini :

Tabel 3.2. Keterangan Tampilan Rancangan Galeri

| No | Nama | Keterangan |
|----|---------------|---|
| 1 | <i>Logo</i> | Untuk menampilkan logo aplikasi |
| 2 | <i>Tabel</i> | Untuk memilih salah satu dataset citra darah |
| 3 | <i>Button</i> | Untuk mengkonfirmasi setelah memilih salah satu dataset |

3.6.3. Tampilan Rancangan Hasil

Gambar 3.8. Tampilan Rancangan Hasil

Berikut ini keterangan mengenai tampilan rancangan desain pada hasil dapat dijelaskan pada tabel dibawah ini :

Tabel 3.3. Keterangan Tampilan Rancangan Hasil

| No | Nama | Keterangan |
|----|------|------------|
|----|------|------------|

| | | |
|---|---------------|--|
| 1 | <i>Logo</i> | Untuk menampilkan logo aplikasi |
| 2 | <i>Output</i> | Untuk melambangkan dataset yang telah dipilih |
| 3 | <i>Output</i> | Untuk menampilkan berapa persentase dataset citra terjangkit malaria |
| 4 | <i>Button</i> | Untuk menuju ke media penyimpanan media galeri pada android |

BAB 4

IMPLEMENTASI DAN PENGUJIAN

Pada bab ini akan dijelaskan mengenai Analisis dan perancangan klasifikasi citra darah menggunakan *Component Analysis Principal* dengan algoritma *neural network*. Tujuan utama analisis ini adalah untuk mengembangkan sistem untuk mengkategorikan gambar darah yang terinfeksi malaria. Secara bersamaan, desain akan merinci cetak biru aplikasi; *Android Studio* digunakan untuk penelitian ini.

4.1. Implementasi

Hasil dari penelitian tugas akhir ini berupa aplikasi deteksi citra darah menggunakan metode *caps neural network*. Aplikasi ini memiliki fungsi untuk mendeteksi citra darah pada sel darah untuk penyakit malaria. Adapun hasil dari aplikasi ini sebagai berikut:

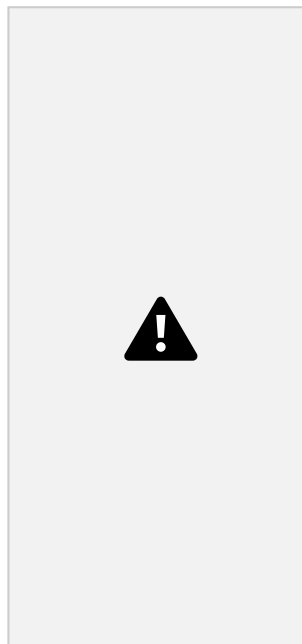
4.1.1. Tampilan Aplikasi

Pada Bagian ini merupakan menu awal yang ditunjukan untuk pengguna yang akan melakukan deteksi citra darah untuk penyakit malaria. Terdapat menu Pilih Foto/Gambar yang berfungsi untuk mengunggah gambar/citra yang akan dideteksi. Tampilan Menu Utama Aplikasi dapat dilihat pada Gambar 4.1.



Gambar 4.1. Tampilan Menu Utama Aplikasi

Pada Gambar 4.2 ditunjukkan ketika proses mengunggah gambar. Pengguna dapat mengambil gambar dari galeri *smartphone* mereka.



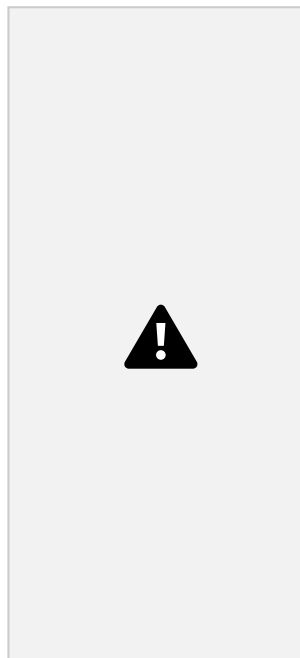
Gambar 4.2. Tampilan Unggah Citra

Kemudian sistem akan langsung mengarahkan pengguna ke galeri yang dipilih, dan menampilkan gambar-gambar yang tersimpan di galeri tersebut. Pada Gambar 4.3 pengguna dapat memilih citra mana yang akan dideteksi di tahap berikutnya.



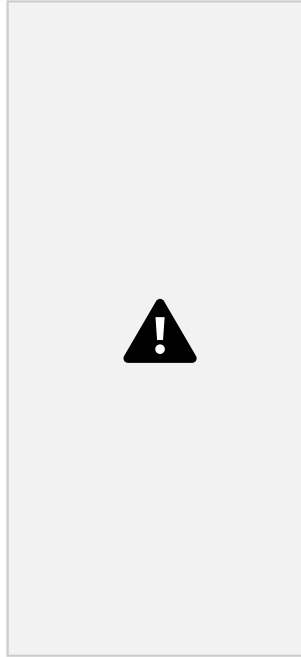
Gambar 4.3. Tampilan Menu Galeri

Berikut ini merupakan tampilan hasil Plasmodium Falciparum seperti gambar 4.4 yang menampilkan persentase plasmodium vivax, plasmodium ovale, plasmodium malariae, plasmodium falciparum dan tidak terinfeksi.



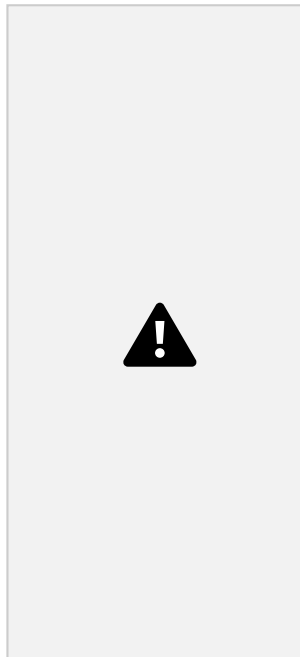
Gambar 4.4. Tampilan Hasil Plasmodium Falciparum

Berikut ini merupakan tampilan hasil Plasmodium Malariae seperti gambar 4.5 yang menampilkan persentase plasmodium vivax, plasmodium ovale, plasmodium malariae, plasmodium falciparum dan tidak terinfeksi.



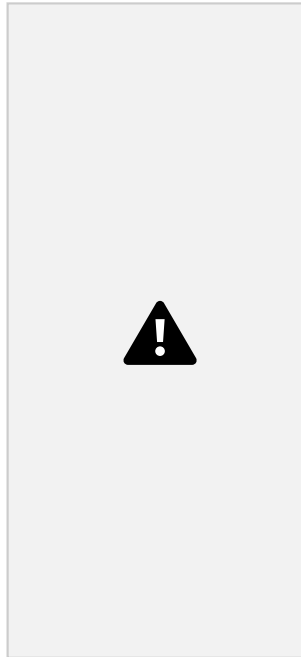
Gambar 4.5. Tampilan Hasil Plasmodium Malariae

Berikut ini merupakan tampilan hasil Plasmodium Ovale seperti gambar 4.6 yang menampilkan persentase plasmodium vivax, plasmodium ovale, plasmodium malariae, plasmodium falciparum dan tidak terinfeksi.



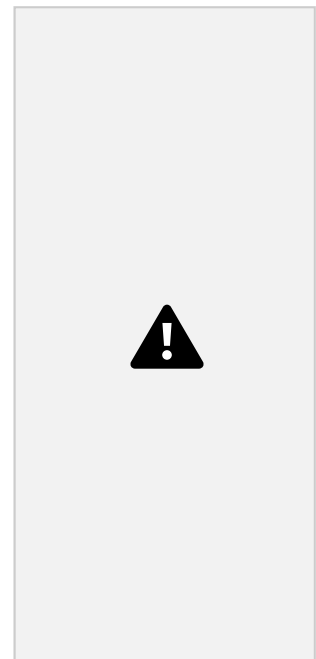
Gambar 4.6. Tampilan Hasil Plasmodium Ovale

Berikut ini merupakan tampilan hasil Plasmodium Vivax seperti gambar 4.7 yang menampilkan persentase plasmodium vivax, plasmodium ovale, plasmodium malariae, plasmodium falciparum dan tidak terinfeksi.



Gambar 4.7. Tampilan Hasil Plasmodium Vivax

Berikut ini merupakan tampilan hasil Tidak Terinfeksi seperti gambar 4.8 yang menampilkan persentase plasmodium vivax, plasmodium ovale, plasmodium

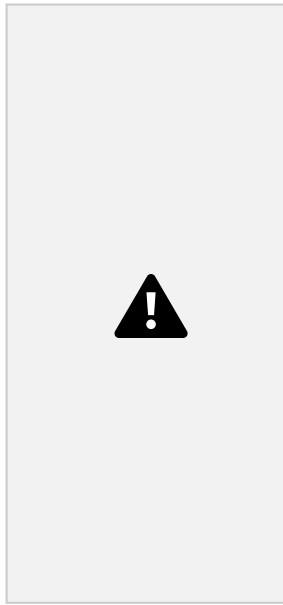


malariae, plasmodium falciparum dan tidak terinfeksi.

Gambar 4.8. Tampilan Hasil Tidak Terinfeksi

4.2. Pengujian

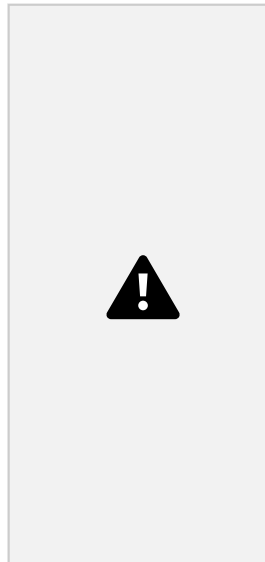
Pada penelitian ini, dilakukan 5 kali pengujian menggunakan 5 gambar/citra sel darah yang berbeda yang mewakili 4 kategori jenis infeksi dan 1 citra mewakili kategori sel darah yang tidak terinfeksi. Hasil pengujian ini memperlihatkan deteksi sel darah yang terinfeksi atau tidak dan jenis infeksi berdasarkan 4 kategori, yaitu *Plasmodium Vivax*, *Plasmodium*



Gambar 4.9. Hasil Pengujian Pertama

Hasil pengujian pertama ditampilkan pada Gambar 4.9, sel darah menunjukkan kecenderungan pada kategori *Plasmodium Falciparum* dengan hasil kemiripan 98,2% dan 1,8% pada *Plasmodium Malariae*. Sedangkan hasil untuk kemungkinan tidak terinfeksi mendapat nilai 0,0%. Oleh karena itu, dapat disimpulkan bahwa sel darah tersebut telah terinfeksi oleh *Plasmodium Falciparum* yang mendominasi jenis infeksi.

48



Gambar 4.10. Hasil Pengujian Kedua

Hasil pengujian kedua ditampilkan pada Gambar 4.10, sel darah menunjukkan kecenderungan pada kategori *Plasmodium Malariae* dengan hasil kemiripan 89,4%. Sedangkan pada kategori yang lain menunjukkan kemiripan sebesar 5,2% pada *Plasmodium Vivax*, 4,9% pada *Plasmodium Ovale* dan 0,1% pada *Plasmodium Falciparum*. Selain itu, hasil juga menunjukkan kemungkinan 0,4% sel darah tersebut tidak terinfeksi, namun hasil ini terlalu kecil untuk

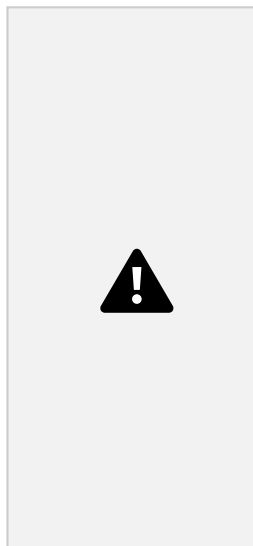
dikatakan tidak terinfeksi. Oleh karena itu, dapat disimpulkan bahwa sel darah tersebut telah terinfeksi oleh *Plasmodium Malariae* yang mendominasi jenis infeksi.



Gambar 4.11. Hasil Pengujian Ketiga

49

Hasil pengujian ketiga ditampilkan pada Gambar 4.11, sel darah menunjukkan kecenderungan pada kategori *Plasmodium Ovale* dengan hasil kemiripan 99,8%. Sedangkan pada kategori lain menunjukkan kemiripan sebesar 0,1% pada *Plasmodium Vivax* dan 0,1% pada *Plasmodium Malariae*. Selain itu, hasil yang menunjukkan kemungkinan sel darah tersebut tidak terinfeksi adalah 0,0%. Oleh karena itu, dapat disimpulkan bahwa sel darah tersebut telah terinfeksi oleh *Plasmodium Ovale* yang mendominasi jenis infeksi.

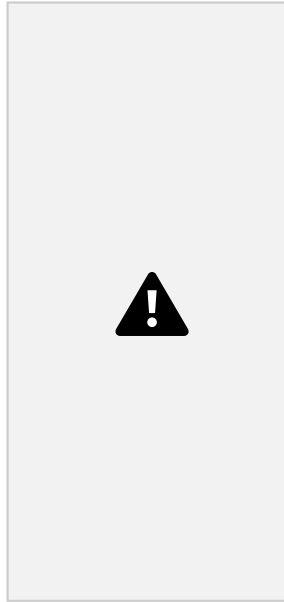


Gambar 4.12. Hasil Pengujian Keempat

Hasil pengujian keempat ditampilkan pada Gambar 4.12, sel darah menunjukkan kecenderungan pada kategori *Plasmodium Vivax* dengan hasil kemiripan 88,4%. Sedangkan pada kategori lain menunjukkan kemiripan sebesar 8,0% pada *Plasmodium Malariae*, 2,6% pada *Plasmodium Falciparum*

dan 0,9% pada *Plasmodium Ovale*. Selain itu, hasil juga menunjukkan kemungkinan 0,1% sel darah tersebut tidak terinfeksi, namun hasil ini terlalu kecil untuk dikatakan tidak terinfeksi. Oleh karena itu, dapat disimpulkan bahwa sel darah tersebut terinfeksi oleh *Plasmodium Vivax* yang mendominasi jenis infeksi.

50



Gambar 4.13. Hasil Pengujian Kelima

Hasil pengujian kelima ditampilkan pada Gambar 4.13, sel darah menunjukkan kemungkinan tidak terinfeksi sebesar 97,1%. Selain itu, pada kategori lain menunjukkan kemiripan sebesar 2,6% pada *Plasmodium Vivax* dan 0,3% pada *Plasmodium Ovale*. Karena hasil nilai kategori infeksi terlalu kecil, dapat disimpulkan bahwa sel darah tersebut tidak terinfeksi. Berdasarkan pengujian menggunakan kelima citra tersebut, dapat disimpulkan bahwa aplikasi deteksi citra penyakit malaria bekerja dengan baik. Ketepatan hasil menunjukkan, aplikasi bisa mengklasifikasikan dengan benar sesuai kategori citra yang dimasukkan.

4.3. Analisis

Analisis dalam penelitian ini diawali dengan analisis pengumpulan dan *preprocessing*.

4.3.1 Perolehan Data

Dalam penelitian ini, kebutuhan data untuk memperoleh citra *digital* yang sesuai untuk klasifikasi menggunakan *Convolutional neural network* akan dibahas pada langkah analisis kebutuhan data.

Data-data yang dibutuhkan dalam penelitian ini adalah *dataset* sel

Institut Kesehatan Nasional AS untuk 27.558 gambar sel berbeda dari 150 pasien yang telah terinfeksi oleh parasit penyebab malaria yang disebut *Plasmodium Falciparum* dan dicampur dengan gambarsel dari 50 pasien sehat.

4.3.2 Analisis *Processing*

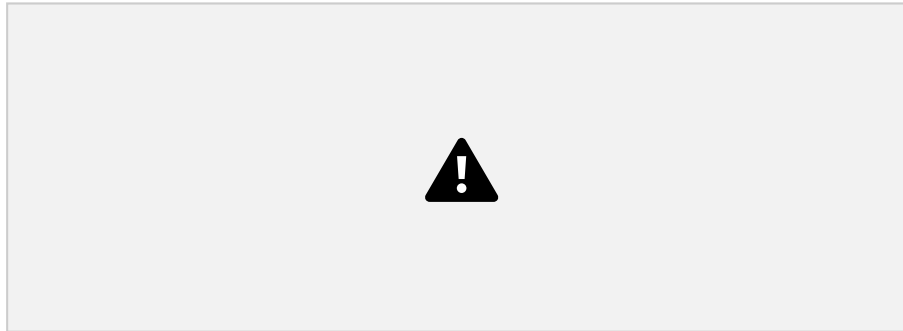
Data gambar mengalami pemrosesan pada tahap *preprocessing* sebagai persiapan klasifikasi menggunakan *Multilayer Perceptron Neural Network*. Tahapan *preprocessing* penelitian ini digambarkan pada diagram alir berikut:



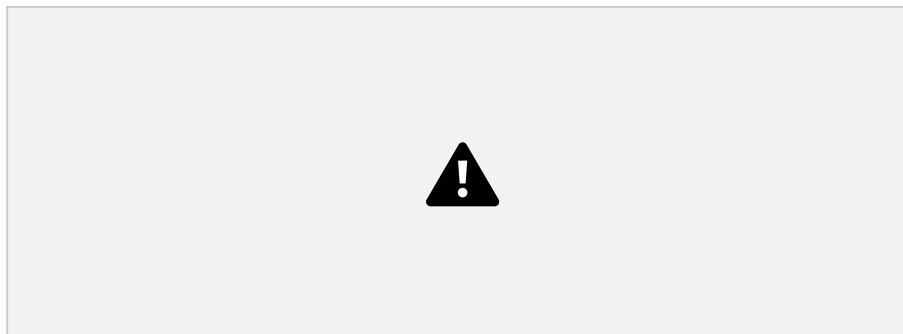
Gambar 4.14. *Processing* CNN

4.3.3 Image Preprocessing and Data Augmentation

Dalam pengujian ini, penulis menggunakan *preprocessing* data citra keras untuk membuat data lebih beragam dan beragam. Kami menggunakan parameter seperti rentang rotasi, geser dan *zoom*, pergeseran lebar, pergeseran tinggi, dan *flip horizontal*. Untuk mengatasi tantangan yang ditimbulkan oleh berbagai ukuran gambar *input*, kami menggunakan parameter penskalaan ulang untuk mengubah ukuran semua gambar ke dimensi 134×134 . Untuk menormalkan nilai tingkat keabuan gambar, kami menggunakan parameter skala ulang, dan kami menggunakan pemisahan data *generator* untuk membagi data asli menjadi pengujian dan pelatihan dengan ukuran berbeda dalam berbagai proporsi. Gambar 4.15 menggambarkan sel darah merah tanpa augmentasi, dan Gambar 4.16 menggambarkan sel darah merah setelah augmentasi.



Gambar 4.15. Sel Darah Merah Tanpa Augmentasi



Gambar 4.16. Sel Darah Merah dengan Augmentasi

4.3.4 Training dan Result

Setelah melakukan *Augmentasi* Data, model CNN yang telah dilatih sebelumnya dilengkapi dengan dataset untuk melakukan *training transfer*. *Layer frozen*, dan *fine-tuning* diterapkan. Hasil akurasi sebelum dan sesudah

menerapkan fine-tuning telah dicatat. Matriks konfusi untuk masing-masing model ini telah diplot untuk mengevaluasi metrik kinerja.

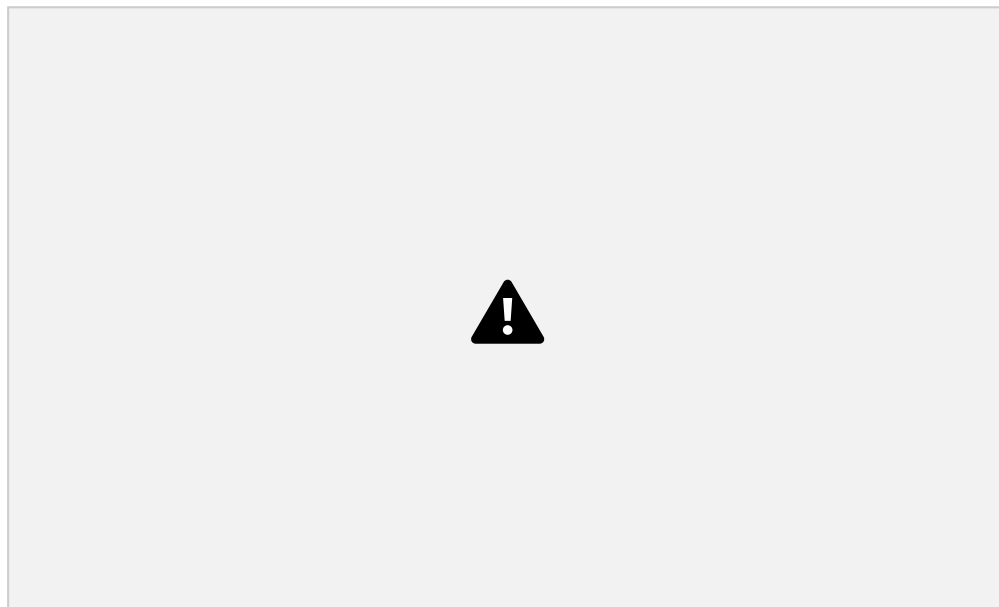
Performa model setelah *fine-tuning* diamati lebih baik dibandingkan dengan *transfer training* saja. Tabel 4.1 menunjukkan hasil akurasi yang diperoleh untuk model *transfer training* sebelum dan sesudah *fine-tuning*. **Tabel 4.1**

Training Model

Model Accuracy when fine tuning is

| | Not applied | Applied |
|-------------------------|----------------------|----------------------|
| | | |
| ResNet-50 34 | 0.957354 0.957358 | 0.958265 0.963527 |
| VGG-16 | 0.960624 | 0.960806 |
| VGG-19 | 0.960987 | 0.972055 |

Akurasi model *transfer training* sebelum dan sesudah *fine-tuning* sama dalam bentuk grafik untuk kedua kasus. Kita dapat menggunakan jaringan yang telah dilatih sebelumnya untuk mengenali kelas yang awalnya tidak deprogram untuk dikenali dengan menyempurnakan tanggapan mereka terhadapnya. Di sisi lain, tingkat akurasi yang lebih rendah dapat dicapai dengan menggunakan pembelajaran transfer melalui ekstraksi fitur.



Gambar 4.17. Akurasi Model *Transfer Training* Sebelum dan Sesudah *Fine-Tuning*

4.3.5 Performance Matrics

Performance Matrics digunakan untuk mengevaluasi kinerja keseluruhan model. Saat menentukan nilainya, matriks kebingungan digunakan untuk

menentukan nilainya. Dalam masalah klasifikasi *machine learning* yang melibatkan dua atau lebih keluaran alternatif, matriks konfusi dapat digunakan untuk mengevaluasi masalah. Ada empat kombinasi berbeda dari data antisipasi dan data aktual yang perlu dipertimbangkan. Untuk perbandingan, *Confusion matrix* ditampilkan dalam kaitannya dengan dataset validasi.

Tabel 4.2 *Confusion Matrix*

| | Predicted class | |
|--------------|--|---|
| | Class = yes | Class = no |
| Actual class | Class = yes True positive (TP) False negative (FN) | Class = no False positive (FP) True negative (TN) |

4.3.6 Result

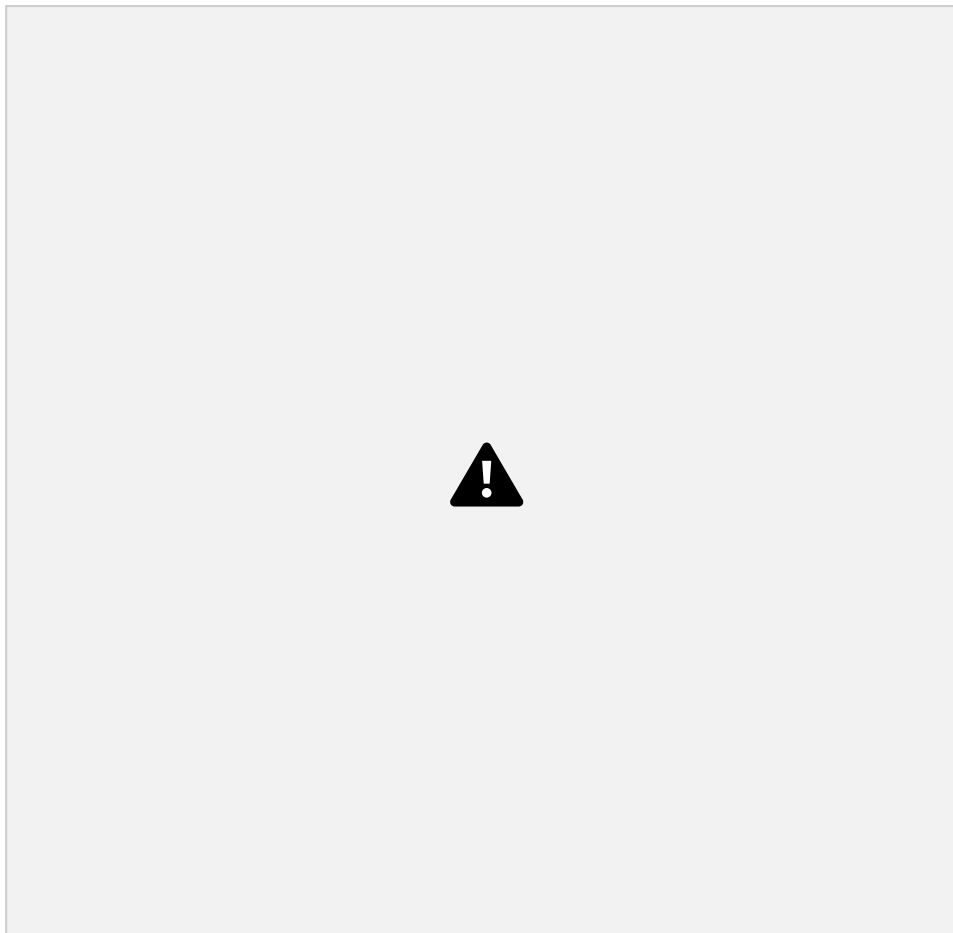
Setelah melakukan *training model* pada data yang dipartisi dalam berbagai rasio, masing-masing model mendapatkan akurasi *training* yang berbeda dan akurasi validasi yang berjalan untuk jumlah waktu yang berbeda karena titik pemeriksaan “*Early Stopping*”.

Tabel 4.3 Perbandingan Akurasi Pada Semua Model

| Data variation (%) | Accuracy (%) | | |
|--------------------|-----------------------------|------------------|------------------|
| | Proposed model (custom CNN) | VGG-19 | Inception v3 |
| Training: 30 | 95.93 (training) | 91.98 (training) | 91.29 (training) |
| Validation: 70 | 95.20 (testing) | 89.99 (testing) | 88.83 (testing) |
| Training: 50 | 95.73 (training) | 93.05 (training) | 91.66 (training) |
| Validation: 50 | 94.8 (testing) | 91.99 (testing) | 89.83 (testing) |
| Training: 70 | 96.36 (training) | 93.34 (training) | 91.06 (training) |
| Validation: 30 | 96.23 (testing) | 91.64 (testing) | 91.03 (testing) |

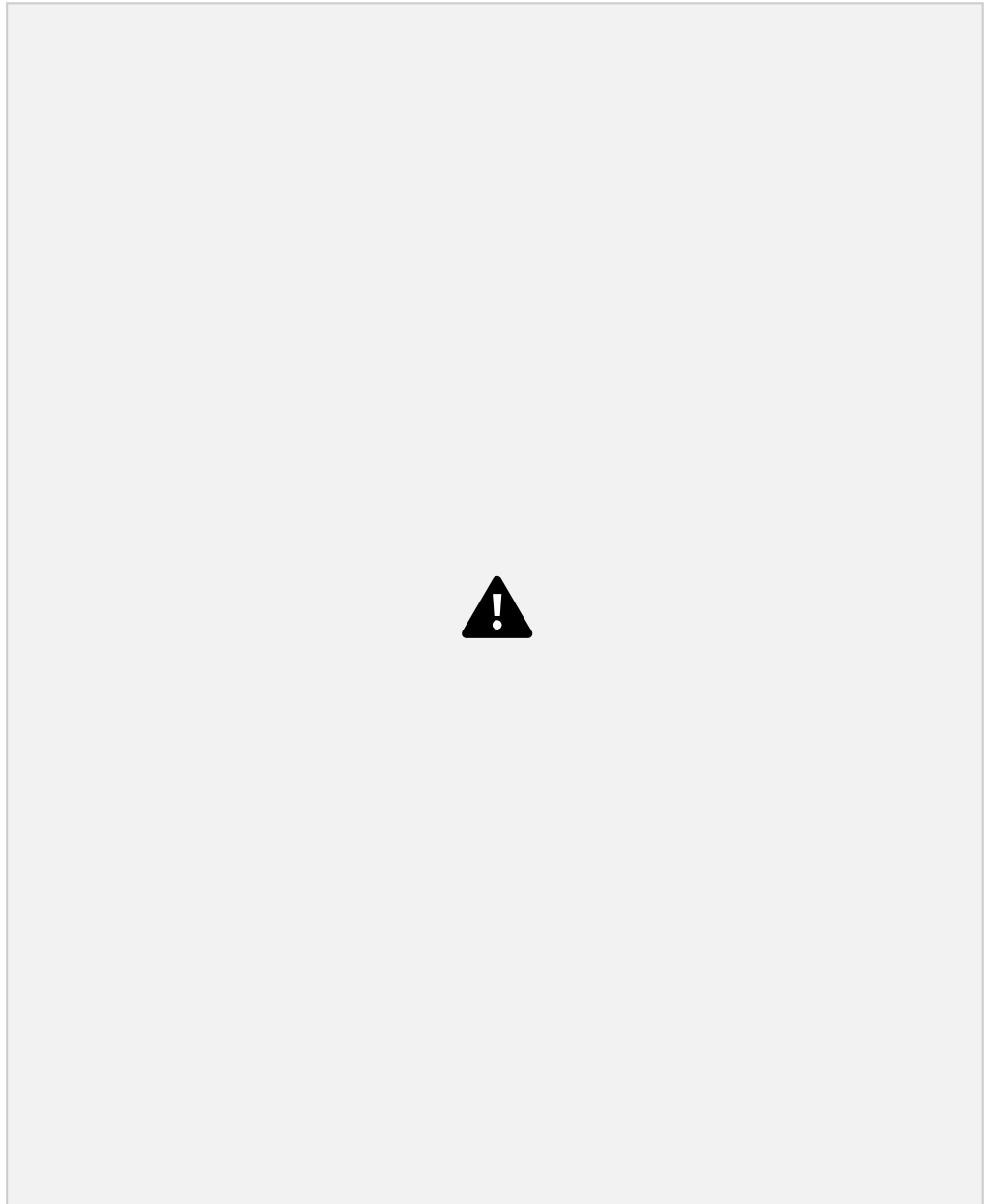
Gambar 4.18 menunjukkan akurasi klasifikasi yang diperoleh dari yang diusulkan model saat dataset dipartisi dalam rasio 30 : 70 masing-masing

menjadi sampel *training* dan sampel validasi.



Gambar 4.18. Akurasi Klasifikasi Dataset Rasio 30 : 70

Gambar 4.19. menunjukkan akurasi klasifikasi yang diperoleh oleh model yang diusulkan ketika dataset dipartisi dalam rasio 50 : 50 masing masing menjadi sampel *training* dan sampel validasi.



Gambar 4.19 Akurasi Klasifikasi Dataset Rasio 50 : 50

Gambar 4.20 menunjukkan akurasi klasifikasi yang diperoleh oleh model yang diusulkan ketika dataset dipartisi dalam rasio 70 : 30 masing-masing menjadi sampel *training* dan sampel validasi.



Gambar 4.20. Akurasi Klasifikasi Dataset Rasio 70 : 30

Dari semua model, CNN yang kami usulkan memberikan *training* terbaik serta akurasi validasi pada semua variasi data. Perbedaan rendah dalam akurasi *training* dan validasi menunjukkan bahwa model kami belum di-*overfit*. Model dilatih selama 15 *epoch* untuk setiap variasi data dengan penerapan *callback* yang sesuai seperti yang dijelaskan di bagian sebelumnya. Model VGG-19 memberikan akurasi *training* maksimum 93,34 dan akurasi validasi 91,64 ketika variasi data 70 : 30. Perbedaan antara akurasi *training* dan validasi lebih kecil

yang

menunjukkan bahwa model ini tidak *overfit* pada data. VGG-19 dilatih selama 15 *epoch* sebagai CNN khusus kami, untuk semua variasi data dengan semua panggilan balik diterapkan. *Inception v3* melakukan yang terburuk pada data yang diberikan, memberikan kinerja terburuk pada semua variasi data. Model ini dilatih untuk *epoch* mulai dari 5 hingga 14 variasi data yang berbeda dengan penerapan callback. Ini dilatih dalam *epoch* yang lebih sedikit saat *callback* diterapkan, dan akurasi model tidak meningkat pada *epoch* yang berurutan, sehingga menghentikan model dari *training* lebih lanjut untuk mencegah *overfitting*.

Untuk evaluasi lebih lanjut dari model kami, kami mengevaluasi model pada lima parameter, yaitu, *Precision*, *recall*, *epoch*, *Auc*, and *F1-score* seperti yang ditunjukkan pada Tabel berikut.

Tabel 4.4 Perbandingan Akurasi Pada Semua Model

| Data Variation % | Model | Class | Precision | Recall | F1 - Score | Epoch | Auc |
|----------------------------------|------------------------------|------------|-----------|--------|------------|-------|------|
| Training : 30 Validation : 70 | Proposed Custom (Custom CNN) | Infected | 0.96 | 0.95 | 0.95 | 15 | 0.98 |
| | | Uninfected | 0.95 | 0.96 | 0.95 | 15 | 0.98 |
| | VGG 19 | Infected | 0.87 | 0.93 | 0.90 | 15 | 0.95 |
| | | Uninfected | 0.92 | 0.87 | 0.89 | 15 | 0.95 |
| | Interception V3 | Infected | 0.86 | 0.91 | 0.89 | 14 | 0.92 |
| | | Uninfected | 0.91 | 0.85 | 0.88 | 14 | 0.92 |
| Training : 50 Validation : 50 | Proposed Custom (Custom CNN) | Infected | 0.93 | 0.95 | 0.94 | 15 | 0.98 |
| | | Uninfected | 0.95 | 0.93 | 0.94 | 15 | 0.98 |
| | VGG 19 | Infected | 0.85 | 0.96 | 0.90 | 15 | 0.95 |
| | | Uninfected | 0.95 | 0.83 | 0.88 | 15 | 0.95 |
| | Interception V3 | Infected | 0.81 | 0.94 | 0.87 | 14 | 0.92 |
| | | Uninfected | 0.93 | 0.78 | 0.85 | 14 | 0.92 |
| Training : 70 Validation : 30 | Proposed Custom (Custom CNN) | Infected | 0.92 | 0.97 | 0.94 | 15 | 0.98 |
| | | Uninfected | 0.97 | 0.92 | 0.95 | 15 | 0.98 |
| | VGG 19 | Infected | 0.87 | 0.94 | 0.90 | 15 | 0.95 |
| | | Uninfected | 0.93 | 0.86 | 0.90 | 15 | 0.95 |
| | Interception V3 | Infected | 0.80 | 0.96 | 0.87 | 14 | 0.92 |
| | | Uninfected | 0.95 | 0.76 | 0.84 | 14 | 0.92 |

BAB 5

KESIMPULAN DAN SARAN

5.1 Kesimpulan

Dengan melatih *Convolutional Neural Network* pada 260 contoh data uji, penelitian ini mengembangkan sistem untuk mengenali pola gambar kode rumput. Temuan berikut berasal dari penelitian komprehensif.

1. Sistem pengenalan deteksi sampel citra darah dengan menggunakan metode *caps* dan algoritma *Convolutional Neural Network* diperoleh hasil akurasi terbaik sebesar 96.92%.
2. Tingkat akurasi ini dipengaruhi oleh nilai *learning rate* pada proses pelatihan, selain itu tingkat resolusi gambar citra darah, jumlah data latih dan data uji serta jumlah lapisan *layer* pada arsitektur CNN pun berpengaruh.

5.2. Saran

Terdapat ruang untuk perbaikan pada sistem yang sedang dibangun yang dapat menghasilkan nilai akurasi yang lebih besar. Berikut beberapa saran yang dapat dimanfaatkan untuk pengembangan di masa depan.

1. Meningkatkan proses pengenalan dengan menggunakan metode ekstraksi fitur dan memanfaatkan berbagai pendekatan *preprocessing*.
2. Klasifikasi menggunakan metode *Deep Learning* lainnya seperti *R CNN*, *Faster R-CNN*, dan *GAN (Generative Adversary Network)*.
3. Meningkatkan pengenalan dengan memperkaya dan memvariasikan data pelatihan.
4. Memastikan untuk menggunakan perangkat keras berspesifikasi tinggi untuk kinerja optimal selama pelatihan dan pengujian.

- Afriyie, Y., A.Weyori, B. & A.Opoku, "A. *Classification of Blood Cells Using Optimized Capsule Networks. Neural Process Lett*" 54, 4809–4828 (2022). <https://doi.org/10.1007/s11063-022-10833-6>.
- Afrizal, Zein. (2019). "Pendeteksian Penyakit Malaria Menggunakan *Medical Images Analisis dengan Deep Learning Python*". *Jurnal Sainstech*. Vol 29, No 1.
- Aisara, S., Azmi, S., & Yanni, M. (2018). "Gambaran Klinis Penderita Penyakit Ginjal Kronik yang Menjalani Hemodialisis di RSUP Dr. M. Djamil Padang". *Jurnal Kesehatan Andalas*. Vol 7, No 1, hal 42. <https://doi.org/10.25077/jka.v7i1.778>.
- Daniel Martomanggolo Wonohadidjojo dan Jonathan Andrei Wibawa. (2023). "Sistem Pemantauan dan Pengendalian Elektronik dengan Stop Kontak Pintar Menggunakan Aplikasi Smartphone". *Jurnal Sistem dan Teknologi Informasi*. Vol 11, No 2.
- DeCoster. (2012). Pengertian *Java Development Kit*. <http://library.binus.ac.id>.
- Departemen Kesehatan RI. (2001). "*Survei Kesehatan Rumah Tangga (SKRT) 2001*". Badan Penelitian dan Pengembangan Kesehatan: Jakarta.
- Effendi, M., Fitriyah, dan Effendi, U. (2017). "Identifikasi Jenis dan Mutu Teh Menggunakan Pengolahan Citra Digital dengan Metode Jaringan Syaraf Tiruan". *Jurnal Teknotan*. Vol 11, No 2, hal 67-76.
- Gopakumar, G. P. et al. "*Convolutional neural network-based malaria diagnosis from focus stack of blood smear images acquired using custom-built slide scanner*". *J. Biophoton*. 11(3), e201700003 (2018).).
- Hutahaean, Harvei Desmon, Bakti Dwi Waluyo, dan Muhammad Amin Rais. (2019). "Teknologi Identifikasi Objek Berbasis Drone Menggunakan Algoritma *Sift Citra Digital*". *Jurnal Teknik Informatika UNIKA Santo Thomas (JTIUST)*, Vol 4, No 2, hal 193–98.
- Jafar Alzubi, Anand Nayyar, Akshi Kumar (2018). "*Machine Learning from Theory to Algorithms: An Overview*". Published under licence by IOP Publishing Ltd. DOI 10.1088/1742-6596/1142/1/012012.
- Julia Fitriany, Ahmad Sabiq (2018). "MALARIA". *Jurnal Kedokteran dan Kesehatan Malikussaleh*. Vol. IV No.2 DOI: <https://doi.org/10.29103/averrous.v4i2.1039>.
- Lesnussa, Yopi Andry, Lexy Jansen Sinay, dan Mus Rika Idah. (2017). "Aplikasi Jaringan Saraf Tiruan *Backpropagation* Untuk Penyebaran Penyakit Demam Berdarah Dengue (DBD) Di Kota Ambon". *Jurnal Matematika Integratif*, Vol 13, No 2, hal.63-72.
- Liang, Z. et al. "*CNN-based image analysis for malaria diagnosis. in IEEE International Conference on Bioinformatics and Biomedicine*", IEEE, 493–496 (2016).

Madhu, G., Mohamed, A.W., Kautish, S. et al. "*Intelligent diagnostic model for malaria parasite detection and classification using imperative inception based capsule neural networks*". *Sci Rep* 13, 13377 (2023). <https://doi.org/10.1038/s41598-023-40317-z>.

Max Pechyonkin, 2017. "Understanding Hinton's *Capsule Networks*. Part I: Intuition" *AI³ Theory, Practice*. Nov 3, 2017.

- P. Lalit Shah et al, 2018. "A Review Paper on Understanding Capsule Networks". *International Journal Of Engineering Development And Research*, 2018 | Volume 6, Issue 4 | ISSN: 2321-9939 .
- Prakarsya, A. (2019). "Perangkat Lunak Permainan Untuk Mendeteksi Dominasi Perkembangan Otak Kanan dan Otak Kiri pada Anak Usia 4-5 Tahun Berbasis Android". In *Prosiding Seminar Nasional Darmajaya*, Vol.1, pp.127 -134.
- Putri, A. R. (2016). "Pengolahan Citra dengan Menggunakan Web Cam Pada Kendaraan Bergerak Di Jalan Raya". *JUPI (Jurnal Ilmiah Pendidikan Informatika)*, Vol 1, No 1, hal 1-6.
- Riti, Yosefina Finsensia. Yulia Wahyuningsih. Josephine. Paulus Wiliam. (2023). Perbandingan Algoritma *Convolutional Neural Network* dan *Capsule Network* Dalam Klasifikasi Jenis Rumah Adat. Surabaya : Jurnal Teknik. Vol.12 No.3.
- Sabour, Sara; Frost, Nicholas; Hinton, Geoffrey E. (26-10-2017). "Dynamic Routing Between Capsules". arXiv: 1710.09829 [cs.CV].
- Sivaramakrishnan, R., Antani, S. & Jaeger, S. "Visualizing Deep Learning Activations For Improved Malaria Cell Classification". *Med. Inf. Healthc.* 1, 40–47 (2018)
- Srihari, Sargur (2017). "Capsule Nets". *University of Buffalo*. Retrieved 2017-12-07
- Sun, C., Shrivastava, A., Singh, S., & Gupta, A. (2011). "Revisiting Unreasonable Effectiveness of Data in Deep Learning Era".
- Widyaningsih, Maura. (2017). "Identifikasi Kematangan Buah Apel dengan Gray Level Co-Occurrence Matrix (GLCM)". *Jurnal SAINTEKOM*. Vol 6, No 1, hal 71. <https://doi.org/10.33020/saintekom.v6i1.7>.
- Arsyad, Azhar. (2019). Media Pembelajaran. PT. Raja Grafindo Persada. Depok.
- Firly, N. *Create Your Own Android Application*. Jakarta: PT Elex Media Komputindo, 2018
- Laporan Kinerja Kesehatan Semester 1 Tahun 2023. Pusat Kebijakan Upaya Kesehatan Badan Kebijakan Pembangunan Kesehatan, Kementerian Kesehatan Republik Indonesia.

- Muhammad Yusril Helmi Setyawan, Aip Suprpto Munari (2020). "Panduan Lengkap Membangun Sistem Monitoring Kinerja Mahasiswa Internship Berbasis Web dan Global Positioning System" Kreatif Industri Nusantara, Mar 31, 2020.
- Nadia, Firly. (2018). "Create Your Own Android Application". Jakarta: PT Elex Media Komputindo.
- Rachmad Hakim.S (2018). "Aplikasi Web/asp.net + cd". Jakarta : Elex Media Komputindo.
- Rachmad, Hakim S. (2018). "Pengertian Aplikasi". Retrieved from <https://lesmardin1988.wordpress.com>
- Rizky, Dhanta. (2009). "Pengantar Ilmu Komputer". Surabaya: Indah. Santoso, Harip. (2010). *Aplikasi Web/asp.net + cd*. Jakarta: Elex Media Komputindo.
- Setyawan dan Munari. (2020). "Panduan Lengkap Membangaun Sistem Monitoring Kinerja Mhasiswa Intership Berbasis Web dan Global

Yuniar, Supardi. (2015). "*Belajar Coding Android Bagi Pemula*". Yogyakarta. PT. Elex Media Komputindo.

<https://sehatnegeriku.kemkes.go.id/baca/rilis-media/20230503/3842868/tahun-ini-5-provinsi-dan-9-kabupaten-kota-berhasil-eliminasi-malaria> (diakses 1 Agustus 2023).

https://en.wikipedia.org/wiki/Capsule_neural_network (diakses 1 Agustus 2023)

LAMPIRAN