

**PEMANFAATAN ALGORITMA EFFICIENTDET PADA DETEKSI
PENYAKIT CACAR MONYET DAN PENYAKIT KULIT
SERUPA MELALUI CITRA LESI KULIT
BERBASIS ANDROID**

SKRIPSI

**AIDA SURYANA RTONGA
171402129**



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

PEMANFAATAN ALGORITMA EFFICIENTDET PADA DETEKSI
PENYAKIT CACAR MONYET DAN PENYAKIT KULIT
SERUPA MELALUI CITRA LESI KULIT
BERBASIS ANDROID

SKRIPSI

Diajukan untuk melengkapi tugas dan memenuhi syarat memperoleh ijazah
Sarjana Komputer Teknologi Informasi

AIDA SURYANA RITONGA
171402129



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

MEDAN

2024

PERSETUJUAN

PERSETUJUAN

Judul : Pemanfaatan Algoritma EfficientDet pada Deteksi
Penyakit Cacar Monyet dan Penyakit Kulit Serupa
Berbasis Android
Kategori : Skripsi
Nama Mahasiswa : Aida Suryana Ritonga
Nomor Induk Mahasiswa : 171402129
Program Studi : Sarjana (S-1) Teknologi Informasi
Fakultas : Ilmu Komputer dan Teknologi Informasi
Universitas Sumatera Utara

Medan, 12 Juli 2024

Komisi Pembimbing:

Pembimbing 2

Fahrurrozi Lubis B.IT., M.Sc.IT
NIP. 198610122018052001

Pembimbing 1

Ade Sarah Huzaifah S.Kom., M.Kom
NIP. 198506302018032001

Diketahui/disetujui oleh

Program Studi S-1 Teknologi Informasi

Ketua,

Dedy Arisandi S.T., M.Kom
NIP. 197908112009121002

PERNYATAAN

**PEMANFAATAN ALGORITMA EFFICIENTDET PADA DETEKSI
PENYAKIT CACAR MONYET DAN PENYAKIT KULIT
SERUPA MELALUI CITRA LESI KULIT
BERBASIS ANDROID**

SKRIPSI

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

Medan , 12 Juli 2024



AIDA SURYANA RITONGA

171402129

UCAPAN TERIMA KASIH

Penulis mengucapkan puji syukur kepada Allah Subhana Wata'ala yang Maha Pengasih dan Penyayang. Berkat rahmat dan kuasa-Nya, penulis berhasil menyelesaikan tugas akhir skripsi ini sebagai syarat untuk meraih gelar Sarjana Komputer di Program Studi S-1 Teknologi Informasi, Fakultas Ilmu Komputer dan Teknologi Informasi, Universitas Sumatera Utara.

Dengan penuh rasa syukur, penulis ingin menyampaikan penghormatan dan ucapan terima kasih yang tulus kepada semua pihak yang telah memberikan bantuan dalam proses penyusunan dan penyelesaian skripsi ini, baik melalui doa, bimbingan, kerjasama, dukungan, maupun kata-kata penyemangat. Penulis menyampaikan rasa terima kasih yang sebesar-besarnya kepada semua pihak yang telah berperan dan membantu dalam penyelesaian skripsi ini.

1. Kedua orang tua penulis, Ayah, Mamak dan juga kepada ketiga saudari kandung yaitu , Rahmika Erida Ritonga, Sri Intan Ritonga, dan Yustika Ayu Ritonga yang selalu mendo'akan dan memberikan perhatian dan kasih sayang kepada penulis serta memberikan motivasi dan dukungan agar penulis dapat menyelesaikan kuliah dengan baik,
2. Bapak Prof.Dr.Muryanto Aminn, S.Sos., M.Si. selaku Rektor Universitas Sumatera Utara.
3. Ibu Dr.Maya Silvi Lydia, B.Sc., M.Sc. selaku Dekan Fakultas Ilmu Komputer dan Teknologi Informasi.
4. Bapak Dedy Arisandi, M.Kom. sebagai Ketua Prodi Departemen Teknologi Informasi Fakultas Ilmu Komputer dan Teknologi Informasi dan Bapak Ivan Jaya., M.Kom selaku Sekretaris Prodi Departemen Teknologi Informasi Universitas Sumatera Utara.
5. Ibu Ade Sarah Huzaifah S.Kom., M.Kom. yang merupakan Dosen Pembimbing Pertama, yang telah sabar membimbing penulis mulai dari awal hingga dapat mengerjakan serta menyelesaikan skripsi ini.

6. Bapak Fahrurrozi Lubis B.IT., M.Sc.IT. yang merupakan Dosen Pembimbing kedua yang telah membimbing penulisan skripsi mulai dari awal sampai selesai.
7. Ibu Marisca Ibu Dosen Penguji, atas masukan dan juga saran yang diberikan kepada penulis dalam menyusun Skripsi ini.
8. Bapak dan Ibu Staff pegawai Prodi Teknologi Informasi USU.
9. Teman seperjuangan penulis yang sudah banyak membantu dalam segala hal, yaitu Aflah Mutsanni Pulungan, Adelia Salmah Siregar, Eka Wulandari, Melati Yulvira Salsabilah, Nuraini, Sintha Sintya Rani, Majidah Atmayana Purba, dan juga teman satu doping Bella Savira.
10. Untuk seluruh Teman-teman Prodi Teknologi informasi angkatan 2017 khususnya teman-teman Kom C yang merupakan teman semasa kuliah penulis.

Semoga dimudahkan urusan dan diberikan rezeki yang berlimpah serta pahala oleh Allah SWT. Dan semoga skripsi ini dapat berguna dan bermanfaat untuk diri saya, untuk keluarga, untuk lingkungan saya, untuk seluruh masyarakat, untuk mahasiswa yang membutuhkan dan untuk negara.

Medan, Juli 2024

Penulis

ABSTRAK

Cacar monyet adalah penyakit menular zoonosis yang disebabkan oleh virus monkeypox, ditularkan dari hewan seperti tikus dan monyet ke manusia, serta antar manusia melalui kontak langsung atau benda terkontaminasi. Gejala termasuk demam, sakit kepala, nyeri otot, kelelahan, dan ruam kulit yang menyerupai cacar biasa. Saat ini sudah dilakukan beberapa penelitian terkait cacar monyet, namun penelitian sebelumnya hanya mengklasifikasikan Cacar Monyet dan bukan Cacar Monyet saja. Penelitian ini memanfaatkan algoritma EfficientDet berbasis Android untuk mendeteksi cacar monyet dengan membandingkan citra lesi kulit manusia yang terpapar penyakit kulit serupa lainnya seperti cacar air, Hand Foot Mouth Disease (HFMD) dan kulit Normal. Deep learning merupakan salah satu metode yang cocok untuk melakukan deteksi citra dengan ukuran data yang relatif besar. Algoritma yang digunakan adalah EfficientDet, karena algoritma ini mampu untuk mendeteksi objek pada gambar atau video dengan akurasi yang tinggi. Data yang digunakan adalah citra lesi kulit penyakit cacar monyet, cacar air, HFMD, dan normal yang diambil dari *open source*, Kaggle dan Dermnet yaitu sebanyak 4.856 data. Data tersebut diberi label Cacar Monyet, Cacar air, HFMD dan Normal. Kemudian dibagi menjadi tiga, yaitu 3.714 data training, 1.062 data validasi dan 80 untuk data testing. Penelitian ini mendapatkan akurasi yang cukup baik dengan rincian sebagai berikut, nilai accuracy 96%, nilai recall 0,96, nilai precision 0,96, dengan model terbaik pada EfficientDet-d1.

Kata kunci : Deteksi, *EfficientDet*, Akurasi, *Training*, *Android*, *Cacar Monyet*, *Cacar Air*, *Hand Foot Mouth Disease*, *Normal*

***UTILIZATION OF EFFICIENTDET ALGORITHM FOR DETECTING
MONKEYPOX AND SIMILAR SKIN DISEASES THROUGH
ANDROID BASED SKIN LESION IMAGES***

ABSTRACT

Monkeypox is a zoonotic infectious disease caused by the monkeypox virus, transmitted from animals such as rodents and monkeys to humans, as well as between humans through direct contact or contaminated objects. Symptoms include fever, headache, muscle aches, fatigue, and skin rashes resembling smallpox. Although several studies on monkeypox have been conducted, previous research has only classified monkeypox rather than focusing solely on it. This study utilizes the EfficientDet algorithm based on Android to detect monkeypox by comparing images of human skin lesions exposed to other similar skin diseases, such as chickenpox, Hand Foot Mouth Disease (HFMD), and normal skin. Deep learning is a suitable method for image detection with relatively large datasets. The chosen algorithm, EfficientDet, is known for its high accuracy in detecting objects in images or videos. The data used includes images of skin lesions caused by monkeypox, chickenpox, HFMD, and normal skin, sourced from open-source platforms like Kaggle and Dermnet, totaling 4,856 images. These images are labeled as Monkeypox, Chickenpox, HFMD, and Normal. The dataset is divided into three parts: 3,714 for training, 1,062 for validation, and 80 for testing. This study achieved a satisfactory level of accuracy with the following metrics: an accuracy rate of 96%, a recall value of 0.96, a precision value of 0.96, with the best model being EfficientDet-d1.

Keywords: *Detection, EfficientDet, Accuracy, Training, Android, Monkeypox, Chickenpox, Hand Foot Mouth Disease, Normal*

DAFTAR ISI

PERSETUJUAN	II
PERNYATAAN	II
UCAPAN TERIMA KASIH	IV
ABSTRAK	VI
ABSTRACT	VII
DAFTAR ISI	VIII
DAFTAR TABEL	X
DAFTAR GAMBAR	XI
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	6
BAB 2 LANDASAN TEORI	7
2.1. Cacar Monyet	7
2.2. Penyakit kulit Serupa	9
2.3. Deteksi Objek	10
2.4. Python	11
2.5. Anaconda Prompt	11
2.6. Node.js	12
2.7. Tensorflow	12
2.8. Expo Go	13
2.9. <i>Deep Learning</i>	13
2.10. <i>Convolutional Neural Network</i>	13
2.11. <i>Image Processing</i>	14
2.12. Algoritma EfficientDet	15
2.13. Penelitian Terdahulu	19

BAB 3 ANALISIS DAN PERANCANGAN	22
3.1 Dataset yang digunakan	22
3.2 Analisis Sistem	23
3.3 Perancangan Sistem	32
BAB 4 IMPLEMENTASI DAN PENGUJIAN	37
4.1 Implementasi Sistem	37
4.2 Implementasi Data	38
4.3 Training dengan Pemanfaatan Model EfficientDet	40
4.4 Implementasi Perancangan Aplikasi	42
4.5 Cara Penggunaan Aplikasi	48
4.6 Pengujian Sistem	49
BAB 5 KESIMPULAN DAN SARAN	54
5.1. Kesimpulan	54
5.2. Saran	54
DAFTAR PUSTAKA	56

DAFTAR TABEL

Tabel 2. 1 Penelitian Terdahulu	19
Tabel 3. 1 Jumlah Data	22
Tabel 4. 1 Spesifikasi kebutuhan Hardware	37
Tabel 4. 2 Pembagian data training berdasarkan label	38
Tabel 4. 3 Pembagian data validasi berdasarkan label	38
Tabel 4. 4 Pembagian data validasi berdasarkan label	38
Tabel 4. 5 Hasil dari Pengujian Sistem	49
Tabel 4. 6 Tabel Confussion Matrix	52
Tabel 4. 7 Tabel Nilai TP, FP, dan FN	52

DAFTAR GAMBAR

Gambar 2. 1 Cacar Monyet	8
Gambar 2. 2 Cacar Air	9
Gambar 2. 3 HFMD	10
Gambar 2. 4 Gambaran sederhana algoritma EfficientDet	15
Gambar 2. 5 Feature Network Design	16
Gambar 3. 1 Arsitektur Umum	23
Gambar 3. 2 Label Cacar Monyet	24
Gambar 3. 3 Label HFMD	25
Gambar 3. 4 Label Cacar Air	25
Gambar 3. 5 Label Normal	25
Gambar 3. 6 Grayscale	26
Gambar 3. 7 Proses Flip	26
Gambar 3. 8 Rotasi	27
Gambar 3. 9 Data Setelah Augmentasi	27
Gambar 3. 10 Arsitektur Backbone EfficientDet	28
Gambar 3. 11 Environment	30
Gambar 3. 12 Install Requirements	30
Gambar 3. 13 Roboflow API	31
Gambar 3. 14 Proses Training	32
Gambar 3. 15 Tampilan Home	34
Gambar 3. 16 Tampilan Scan	35
Gambar 3. 17 Tampilan Informasi	36
Gambar 4. 1 Data penyakit Cacar Monyet	39
Gambar 4. 2 Data penyakit Cacar Air	39
Gambar 4. 3 Data penyakit Hand foot mouth disease	39
Gambar 4. 4 Data kulit Normal	40
Gambar 4. 5 Konfigurasi pipeline (Train Config)	40
Gambar 4. 6 Grafik Hasil Training	42
Gambar 4. 7 Nilai mAP	42

Gambar 4. 8 Tampilan Halaman Splash Screen	43
Gambar 4. 9 Tampilan Home	44
Gambar 4. 10 Tampilan Scan Aplikasi	45
Gambar 4. 11 Hasil Proses Gambar	46
Gambar 4. 12 Tampilan Informasi	47

BAB 1

PENDAHULUAN

1.1. Latar Belakang

Cacar monyet, juga dikenal sebagai *monkeypox*, yaitu penyakit menular zoonosis yang disebabkan oleh virus monkeypox yang termasuk dalam genus Orthopoxvirus, famili Poxviridae, dan famili Chordopoxvirinae (Kaler et al., 2022). Virus ini awalnya berasal dari hewan, terutama hewan pengerat dan primata seperti tikus, monyet, atau tupai yang terinfeksi. Penyakit ini dapat menular baik dari hewan ke manusia maupun dari manusia ke manusia. Penularan *Monkeypox* pada manusia terjadi melalui kontak langsung dengan orang ataupun hewan yang terinfeksi, selain itu bisa juga melalui benda yang terkontaminasi oleh virus tersebut (Sinto et al., 2022). Infeksi pertama diidentifikasi pada manusia sejak awal tahun 1970 Di Republik Demokratik Kongo.

Cacar Monyet bisa sembuh sendiri dalam beberapa waktu namun lain halnya jika virus menyerang bayi baru lahir, anak-anak, dan orang-orang dengan kekebalan tubuh yang lemah mungkin mengalami gejala yang lebih parah dan dapat menyebabkan kematian (Ludji & Buan, 2023). Berdasarkan data dari World Health Organization (WHO), pada 28 September 2022 terkonfirmasi kasus cacar monyet mencapai 67.539 kasus dan angka kematian mencapai 27 jiwa di 105 negara (Marisah et al., 2022).

Gejala-gejala cacar monyet mencakup demam, sakit kepala, nyeri otot, nyeri punggung, kelelahan, pembengkakan kelenjar getah bening, dan yang lebih penting, ruam yang ditemukan di wajah, telapak tangan, telapak kaki, mata, mulut, tenggorokan, selangkangan, dan daerah genital pada tubuh bisa menjadi contoh dari gejala cacar monyet (World Health Organisation, 2022). Gejala ruam ini serupa dengan penyakit cacar pada seperti cacar air memiliki gejala yang mirip, yaitu ruam pada kulit mulai lesi datar berwarna merah, benjolan kecil kemerahan seperti jerawat, lenting berisi cairan yang berubah warna menjadi keruh, hingga mengering dan membentuk keropeng (Ng, 2021). Kemiripan gejala ini merupakan tantangan bagi

para tenaga kesehatan dalam membedakan Cacar Monyet dari penyakit kulit serupa, seperti Cacar air, dan *Hand Foot Mouth Disease*/Flu Singapura.

Penggunaan Kecerdasan Buatan (KB), dalam pendekatan *deep learning*, telah banyak digunakan dalam hal analisis citra medis. *Deep learning* adalah metode pembelajaran mesin yang menggunakan jaringan saraf tiruan dengan banyak lapisan untuk memahami dan memproses data. Penelitian terkait yang membahas mengenai deteksi citra penyakit cacar monyet dengan mengimplementasikan deep learning adalah penelitian yang dilakukan oleh (Susanto et al., 2023) melakukan penelitian tentang virus cacar monyet (*Monkeypox*) dengan menggunakan VGG-19 pada arsitektur CNN. Dataset yang digunakan dalam penelitian ini berjumlah 1.000 citra lesi kulit serupa virus monkeypox untuk diukur tingkat persentase akurasi, sensitivitas, dan spesifitas. Hasil yang diperoleh diperoleh tingkat akurasi sebesar 76%.

Victory (2023) dalam Penelitiannya yang membahas tentang klasifikasi *monkeypox* dan *non monkeypox* menggunakan Ekstraksi LBP dan algoritma Random Forest. Dengan jumlah dataset keseluruhan *monkeypox* dan *non monkeypox* berjumlah 3.192 gambar dengan hasil akurasi 85%. Juga ada penelitian dari (Wijaya et al., 2023) Dalam penelitian ini Mengklasifikasikan Monkeypox menggunakan ekstraksi fitur GLCM dan juga menggunakan algoritma Random Forest. Dataset yang digunakan berjumlah 3.192 gambar, yang menghasilkan akurasi terbaik sebesar 77%.

Juga ada (Nayak et al., 2023) melakukan penelitian Deteksi virus Monkeypox berdasarkan citra gambar lesi kulit menggunakan *Deep learning*. Dengan metode penelitian menggunakan *Ensemble Learning* dengan tiga model dasar yang telah dilatih sebelumnya, yaitu *Inception V3*, *Xception*, dan *DenseNet169*. Hasil penelitian pada dataset gambar lesi kulit monkeypox mencapai rata-rata akurasi 93,39%.

Sahin et al., (2022) yang mengimplementasikan Tensorflow Lite dalam Klasifikasi Cacar Monyet pada Manusia dari Gambar Lesi Kulit dengan Deep Pre-Trained. Hasilnya sistem dapat mengklasifikasikan gambar penyakit monkeypox dengan akurasi 91,11%.

Penelitian terkait arsitektur dalam penerapan algoritma CNN dengan memanfaatkan EfficientDet yaitu ada, (Tan et al., 2020) dalam penelitiannya

menciptakan EfficientDet yang menggunakan bi-directional feature pyramid network (BiFPN) dan metode penskalaan gabungan untuk meningkatkan efisiensi model. Penelitian dari Nursulistio et al., (2022) menggunakan EfficientDet untuk deteksi objek masker dengan akurasi 75,84%. Kemudian ada Ferian et al., (2023) menerapkan *Deep Learning* menggunakan EfficientDet untuk deteksi kelayakan penerima bantuan langsung tunai berdasarkan citra rumah dengan akurasi 93%. Dan terakhir ada dari (Nainggolan, 2023) melakukan klasifikasi penyakit Alzheimer menggunakan EfficientDet menghasilkan akurasi 97,4%

Algoritma ini dirancang untuk mengoptimalkan proses deteksi objek dalam gambar yang memiliki variasi skala yang luas, dari objek berukuran kecil hingga besar. Algoritma ini juga memiliki kemampuan untuk dilatih menggunakan dataset yang relatif kecil dan kemudian diterapkan pada dataset yang lebih besar. Mengingat variasi ukuran gambar dan jumlah yang cukup banyak, EfficientDet menjadi salah satu algoritma yang handal untuk melakukan deteksi objek pada gambar lesi kulit, khususnya untuk deteksi penyakit cacar monyet dan penyakit kulit serupa. Dengan pendekatan ini, diharapkan dapat mengatasi tantangan dalam mengenali penyakit cacar monyet secara lebih akurat.

Adapun tujuan dari Pemanfaatan algoritma Efficientdet untuk mendeteksi penyakit cacar monyet dengan membandingkan citra lesi kulit manusia yang terpapar Monkeypox dengan penyakit kulit serupa seperti cacar air, Hand foot mouth disease/Flu Singapura dan normal. Berdasarkan pada latar belakang diatas, penulis melakukan penelitian dengan judul **“Pemanfaatan Algoritma EfficientDet pada Deteksi Penyakit Cacar Monyet dan Penyakit Kulit Serupa Berbasis Android ”**

1.2. Rumusan Masalah

Cacar monyet merupakan penyakit menular zoonosis yang disebabkan oleh virus monkeypox yang termasuk dalam genus Orthopoxvirus (Kaler et al., 2022). Penularan Monkeypox pada manusia terjadi melalui kontak langsung dengan orang ataupun hewan yang terinfeksi. Cacar Monyet bisa sembuh sendiri namun, komplikasi dari cacar monyet dapat menyebabkan kematian. Data dari World Health Organization (WHO, 2022), terkonfirmasi kasus cacar monyet mencapai 67.539 kasus dan angka kematian mencapai 27 jiwa. Gejala yang muncul ketika seseorang terinfeksi cacar monyet adalah munculnya ruam pada berbagai bagian tubuh seperti tangan, kaki, dada, wajah, dan

anggota tubuh lainnya. Gejala-gejala ini serupa dengan penyakit kulit pada umumnya karena berasal dari famili yang sama. Untuk itulah perlu dilakukan penelitian untuk mendeteksi penyakit cacar monyet dengan penyakit kulit serupa.

1.3. Batasan Masalah

Adapun batasan masalah dalam penelitian ini yaitu:

1. Data diambil dari open source yang ada di Kaggle, dan Dermnet.
2. Deteksi dilakukan berdasarkan bentuk dari citra kulit manusia.
3. Hasil Deteksi dari penelitian ini adalah cacar monyet, dan penyakit kulit serupa, yaitu Cacar air, Hand Foot Mouth Disease/Flu Singapura dan Normal.
4. Jarak pengambilan gambar untuk capture dari android harus dilakukan dengan jarak 3 cm, dan pengambilan gambar untuk seluruh badan paling jauh 1 meter.
5. Pengambilan gambar harus berada di tempat yang memiliki cahaya yang cukup baik.

1.4. Tujuan Penelitian

Penelitian ini bertujuan untuk mengimplementasikan algoritma EfficientDet pada proses deteksi penyakit kulit Cacar Monyet, dan penyakit kulit serupa, seperti Cacar air, Hand Foot Mouth Disease/Flu Singapura dan Normal.

1.5. Manfaat Penelitian

Manfaat pada penelitian ini yaitu:

1. Penelitian ini diharapkan dapat membantu masyarakat agar bisa mengidentifikasi penyakit cacar monyet dan mampu membedakan dengan penyakit kulit sejenis.
2. Mengetahui kemampuan metode EfficientDet dalam deteksi penyakit cacar monyet dan penyakit kulit serupa seperti, Cacar air, Hand Foot and Mouth Disease/flu singapura, dan normal.

1.6. Metodologi Penelitian

Berikut ini merupakan beberapa tahap yang dikerjakan penulis dalam perancangan penelitian yaitu:

1. Studi Literatur

Ditahap ini mempelajari dan mengumpulkan informasi yang dibutuhkan terkait penelitian yang nantinya akan dilakukan. Referensi ini beberapa skripsi, makalah, jurnal, buku, internet, dan sumber lain yang mendukung penelitian ini dan juga berhubungan dengan algoritma efficientdet.

2. Analisis Permasalahan

Permasalahan yang telah ditentukan untuk diteliti kemudian dianalisis guna memperoleh pengetahuan terkait metode yang akan diterapkan dalam melakukan prediksi yaitu menggunakan metode efficientdet.

3. Perancangan Sistem

Setelah menganalisis penulis merancang sistem dan melakukan *training* data yang nantinya berguna untuk mendapatkan bobot nilai dari data yang diteliti.

4. Implementasi Sistem

Setelah itu dilakukan *testing* terhadap model efficientdet dengan model yang nantinya akan dibangun, dengan menggunakan *testing data* untuk mendapatkan *output* didalam penelitian ini.

5. Pengujian Sistem

Langkah selanjutnya adalah mendeteksi gambar lesi kulit penyakit cacar monyet, cacar air, Hand foot mouth disesase, dan normal, serta menguji system yang dibangun untuk mengetahui bahwa hasil dari penelitian ini sesuai dengan target.

6. Penyusunan Laporan

Dalam tahap ini merupakan tahapan terakhir, penulis akan membuat dokumentasi dan melaporkan setiap kegiatan yang dilakukan berupa laporan penelitian.

1.7. Sistematika Penulisan

Berikut sistematika penulisan yang akan menjabarkan lima bab dengan penjelasan tersendiri, dari setiap bab akan dijelaskan secara singkat sebagai berikut:

Bab 1: Pendahuluan

Bab ini memaparkan latar belakang, lalu batasan masalah, tujuan penelitian, manfaat penelitian, kemudian yang terakhir adalah metodologi penelitian dan juga sistematika penulisan.

Bab 2: Landasan Teori

Dalam bab ini berisikan tentang landasan teori untuk bisa memahami setiap permasalahan yang dibahas dan berkaitan tentang penelitian ini, seperti Cacar Monyet dan algoritma Efficientdet.

Bab 3: Analisis dan Perancangan Sistem

Dalam bab ini menjelaskan tentang arsitektur umum dan analisis pada tahapan tentang *pre-processing*, proses *training*, *validasi* dan *testing* serta sampai penelitian ini mendapatkan hasil deteksi dengan kecepatan akurasi dari citra lesi kulit dengan metode Efficientdet supaya dapat digunakan untuk menguji sistem.

Bab 4: Implementasi dan Pengujian Sistem

Dalam bab 4 ini menjelaskan peng-aplikasian suatu program terhadap sistem dari suatu hasil menganalisis serta suatu perancangan yang sebelumnya telah dijelaskan pada Bab terdahulu, serta pembahasan hasil yang sudah diperoleh dari hasil pengujian sistem.

Bab 5: Kesimpulan dan Saran

Dalam bab ini berisikan tentang kesimpulan penelitian yang telah dibahas pada bab sebelumnya dan saran dari penelitian ini untuk perbaikan serta pengembangan pada penelitian selanjutnya.

BAB 2

LANDASAN TEORI

2.1. Cacar Monyet

Cacar Monyet atau yang dikenal dengan *monkeypox* ialah penyakit yang disebabkan virus *monkeypox* dari *family Orthopoxvirus, Poxviridae* (Kaler et al., 2022) yang ditemukan ketika penyakit yang terlihat seperti cacar ini terjadi pada sebuah koloni monyet yang dirawat untuk kepentingan sebuah penelitian. Karena itu, infeksi dari penyakit ini dinamakan dengan *monkeypox*. Pada tahun 1970 di Negara Republik Demokratik Kongo, untuk pertama kalinya kasus cacar monyet menginfeksi manusia dan itu terjadi di wilayah *endemis* pedesaan, hutan tropis Congo Basin, dan Afrika Barat (Bunge et al., 2022). Penyakit ini dapat ditularkan melalui kontak langsung, terhadap hewan pengerat, serta daging yang dikonsumsi tidak dimasak sampai matang. Infeksi *inokulasi* secara kontak langsung dengan lesi pada kulit maupun mukosa hewan yang terinfeksi (Petersen et al., 2019)

Tanda dan gejala *monkeypox* pada manusia mirip dengan cacar tetapi lebih ringan. Masa inkubasi dari infeksi hingga gejala awal cacar monyet adalah 12 hari. Selama 5 hari pertama setelah terinfeksi, pasien mengalami gejala mirip flu antara lain, Demam, Sakit kepala Nyeri otot Sakit punggung Kelelahan (kurang tenaga) Limfadenopati (pembengkakan kelenjar getah bening) Limfadenopati merupakan gejala penting yang berbeda dengan cacar air.

Setelah periode ini, timbul ruam yang mirip dengan cacar air. Lesi terutama muncul di wajah, namun bisa juga terjadi di telapak tangan, telapak kaki, serta tangan dan kaki. Pada tahun 2022, wabah terbaru ditandai dengan lesi genital dan perigenital. Lesi orofaring, eritema, edema dan ulserasi (Gilbourne, 2022).

Ruam dimulai dengan ruam makulopapular dengan diameter 2 sampai 5 mm dan berkembang menjadi *Bullae* (lepuh kecil berisi cairan) Pseudopustula (lesi padat menyerupai pustula asli) *Eschar* pada bekas luka atrofi. Gejala biasanya berlangsung 14 hingga 21 hari. Pada Agustus 2022, sejumlah besar orang dilaporkan terkena dampak wabah tahun 2022 di Spanyol.

Sebagian besar individu yang terkena dampak adalah laki-laki dan hanya memiliki sedikit lesi (total kurang dari 20), seringkali hanya di satu lokasi. Lesi sering dimulai pada alat kelamin, wajah, lengan, tangan, dan daerah perianal. Di tempat pustula yang sebenarnya, dilaporkan ada pseudopustula yang keras, berwarna putih, menonjol, dan tidak memiliki lapisan atas yang dapat dikikis untuk mengeluarkan nanah.

Dalam kasus yang parah, monkeypox dapat menyebabkan komplikasi seperti pneumonia. Monkeypox secara utama ditemukan di Afrika Tengah dan Barat, terutama di daerah hutan hujan tropis. Namun, kasus monkeypox di luar Afrika telah dilaporkan, termasuk di Amerika Serikat dan negara-negara lain. Kasus manusia di luar Afrika sering terkait dengan impor hewan yang terinfeksi atau kontak dengan wisatawan yang terinfeksi.

Tidak ada pengobatan khusus untuk monkeypox, tetapi vaksinasi terhadap cacar telah terbukti memberikan perlindungan. Langkah-langkah pencegahan lainnya termasuk menghindari kontak dengan hewan liar, terutama yang sakit atau mati, dan menjaga kebersihan yang baik, seperti mencuci tangan secara teratur.

Mengingat potensi monkeypox untuk menyebabkan wabah pada populasi manusia, penelitian dan pengawasan yang berkelanjutan sangat penting untuk memantau penyebaran virus dan mengembangkan strategi pencegahan dan pengendalian. Untuk mengetahui bagaimana gambar lesi kulit cacar monyet dapat dilihat pada Gambar 2.1.



Gambar 2. 1 Cacar Monyet

(Sumber: (Bala et al., 2023))

2.2. Penyakit kulit Serupa

Penyakit kulit yang memiliki tampilan serupa dengan cacar monyet yaitu termasuk penyakit cacar air (chickenpox) dan Hand Foot Mouth Disease, yang juga dikenal sebagai penyakit flu Singapura.

2.2.1. *Cacar Air*

Penyakit cacar air, atau chickenpox, disebabkan oleh virus varicella zoster, yang termasuk dalam kelompok virus alphaherpes dan bersifat imunogenik. Penyakit ini dikenal dengan penularan yang sangat cepat. Seseorang yang terinfeksi dapat menularkan virus melalui cairan tubuh, seperti saat bersin atau batuk, sehingga orang sehat yang terkena cipratan tersebut berisiko terjangkit virus ini. (Anggeriyane et al., 2022). Penyakit cacar air memiliki ciri khas berupa benjolan kecil mirip bisul dengan lapisan tipis berisi cairan, yang sering disebut plentingan. Awalnya, muncul kemerahan pada kulit yang kemudian berubah menjadi bintik-bintik berisi cairan dan menyebar ke seluruh tubuh, bahkan hingga ke bagian mata. (Susanto, 2022). Untuk mengetahui bagaimana gambar lesi kulit cacar air dapat dilihat pada Gambar 2.2.



Gambar 2. 2 Cacar Air

(Sumber : Kaggle, 2022)

2.2.2. *Penyakit Hand Foot Mouth Disease*

Hand-foot-and-mouth disease (HFMD) adalah penyakit infeksi sistemik akut yang disebabkan oleh enterovirus. Penyakit ini ditandai dengan lesi berbentuk ulkus di mulut dan eksantema berbentuk vesikel pada ekstremitas bagian distal, disertai gejala konstitusi yang ringan dan biasanya sembuh dengan sendirinya (Belazarian et al., 2011). Anak-anak di bawah usia 10 tahun adalah kelompok yang paling rentan terkena

penyakit ini, dan wabah sering kali terjadi di antara anggota keluarga. Penyakit ini sangat menular dan ditandai dengan adanya lesi di mulut serta lesi kulit pada ekstremitas bagian distal. Penularan dapat terjadi melalui kontak langsung dengan cairan dari lesi, serta melalui droplet, Droplet adalah partikel kecil dari cairan yang dilepaskan ke udara ketika seseorang batuk, bersin, berbicara, atau bernapas. Penyakit ini menjadikannya mudah menyebar di lingkungan keluarga atau tempat bermain anak-anak. (Wolf, K et al., 2005). Untuk mengetahui bagaimana gambar lesi kulit hand foot mouth disease dapat dilihat pada Gambar 2.3.



Gambar 2. 3 HFMD

(Sumber : Kaggle, 2022)

2.3. Deteksi Objek

Deteksi objek adalah teknologi komputer yang memungkinkan mesin untuk mengidentifikasi dan melacak berbagai objek dalam gambar atau video. Dengan menggunakan algoritma dan kecerdasan buatan, deteksi objek memungkinkan komputer untuk mengenali objek-objek tertentu dan menentukan posisi serta batas-batasnya dalam citra visual. Ini membantu dalam otomatisasi proses analisis visual yang sebelumnya hanya dapat dilakukan oleh manusia (Brownlee, 2021).

Deteksi objek merupakan teknologi yang mampu memberikan informasi penting dalam hal pemahaman semantik gambar dan video. Teknologi ini berperan krusial dalam berbagai aplikasi, seperti klasifikasi gambar, analisis perilaku manusia, pengenalan wajah, serta pengembangan kendaraan otonom. Dengan deteksi objek, sistem dapat mengenali dan mengkategorikan berbagai elemen dalam gambar atau video, memungkinkan interpretasi yang lebih mendalam dan kontekstual. Misalnya,

dalam klasifikasi gambar, teknologi ini dapat mengidentifikasi objek-objek spesifik dalam sebuah gambar dan mengelompokkannya ke dalam kategori yang relevan. Dalam analisis perilaku manusia, deteksi objek dapat digunakan untuk memantau dan menganalisis gerakan atau aktivitas individu, yang bermanfaat dalam bidang keamanan atau penelitian ilmiah. Pengenalan wajah, yang merupakan salah satu aplikasi deteksi objek, memungkinkan identifikasi individu berdasarkan fitur wajah mereka, yang digunakan dalam berbagai konteks mulai dari keamanan hingga interaksi pengguna dengan perangkat teknologi. Dalam mengemudi otonom, deteksi objek menjadi komponen penting untuk mengenali dan bereaksi terhadap lingkungan sekitar, seperti kendaraan lain, pejalan kaki, dan rambu lalu lintas, sehingga mendukung navigasi yang aman dan efisien. (Zhao et al., 2019).

2.4. Python

Python adalah bahasa pemrograman yang digunakan untuk berbagai tujuan umum seperti membangun situs web, software, atau aplikasi, mengotomatiskan tugas, dan melakukan analisis data. Python dapat digunakan untuk membuat berbagai program, bukan hanya untuk menyelesaikan masalah tertentu. Python menjadi bahasa pemrograman yang paling banyak digunakan karena mudah digunakan, terutama untuk pemula.

Di dalam Python Ada lebih dari 125.000 library pihak ketiga yang dapat digunakan untuk berbagai tugas seperti, machine learning, pemrosesan website, dan bahkan biologi. Library data yang berfokus pada data seperti pandas, NumPy, Pytorch dan matplotlib karena memproses, mengubah, dan menampilkan data dengan mudah.

2.5. Anaconda Prompt

Anaconda Prompt adalah terminal perintah baris yang disediakan oleh distribusi Anaconda untuk membantu pengguna mengelola lingkungan dan paket Python. Fungsinya adalah agar pengguna dapat membuat, mengaktifkan, menonaktifkan, dan menghapus lingkungan virtual Python dengan Anaconda Prompt.

Dalam Anaconda, memungkinkan pengguna menginstal, memperbarui, dan menghapus paket Python. Ini membuat pengelolaan pustaka dan modul yang diperlukan untuk berbagai proyek lebih mudah. Juga dapat menggunakan Anaconda Prompt untuk

menjalankan skrip Python dan aplikasi lainnya yang terintegrasi dengan lingkungan Anaconda.

2.6. Node.js

Node.js adalah lingkungan runtime open-source dan cross-platform yang memungkinkan eksekusi kode JavaScript di luar peramban web. Dikembangkan oleh Ryan Dahl pada tahun 2009, Node.js menggunakan mesin JavaScript V8 dari Google Chrome. Fungsi utama dari Node.js yaitu memungkinkan pengembang untuk menulis kode server-side menggunakan JavaScript, yang biasanya digunakan untuk client-side scripting.

Juga Node.js banyak digunakan untuk membuat aplikasi web, terutama yang memerlukan komunikasi real-time seperti aplikasi chat dan permainan multiplayer. Dan juga Node.js sering digunakan untuk membuat API dan layanan mikro karena kecepatan dan skalabilitasnya.

2.7. Tensorflow

TensorFlow adalah library perangkat lunak yang fleksibel dan skalabel yang dirancang untuk komputasi numerik menggunakan grafik aliran data. Library ini, bersama dengan alat-alat terkaitnya, memungkinkan pengguna untuk memprogram, melatih, dan menerapkan jaringan saraf serta algoritma pembelajaran mesin secara efisien dalam berbagai lingkungan produksi. Dengan menggunakan grafik aliran data, TensorFlow memungkinkan pengembang untuk membangun model komputasi kompleks dengan menyusun operasi sebagai grafik yang terdiri dari node (representasi operasi matematis) dan edge (representasi aliran data).

TensorFlow menawarkan kemampuan untuk melakukan komputasi secara efisien pada berbagai perangkat keras, termasuk CPU, GPU, dan TPU, yang membuatnya sangat berguna dalam pengembangan dan pelatihan model pembelajaran mesin yang membutuhkan daya komputasi tinggi. Selain itu, library ini menyediakan berbagai API tingkat tinggi dan rendah yang memudahkan pengembang dengan berbagai tingkat keahlian untuk membuat, melatih, dan menyebarkan model pembelajaran mesin.

Dengan demikian, TensorFlow tidak hanya menawarkan fleksibilitas dan skalabilitas, tetapi juga menjadikannya alat yang sangat berguna bagi siapa pun yang bekerja di bidang komputasi numerik dan pembelajaran mesin. (Pang et al., 2020)

2.8. Expo Go

Expo Go adalah aplikasi mobile pada sebuah framework untuk membangun aplikasi React Native dengan cepat dan mudah. Expo menyediakan berbagai alat dan layanan yang mempermudah pengembangan, pengujian, dan penyebaran aplikasi mobile. Adapun fungsi utama Expo Go yaitu, aplikasi ini memungkinkan pengembang untuk melihat perubahan dalam aplikasi secara real-time saat menulis kode. Ini sangat berguna untuk proses pengembangan dan debugging. Juga manfaat yang lain pengembang dapat menguji aplikasi mereka di berbagai perangkat tanpa perlu mengkompilasi ulang atau menginstal APK/IPA baru setiap kali ada perubahan (Wern, 2018)

2.9. Deep Learning

Algoritmanya memiliki ratusan lapisan jaringan saraf (Wulandari, et al., 2020). Salah satu kelebihan deep learning dapat membantu menyelesaikan menyelesaikan masalah yang cukup kompleks, seperti kemampuan untuk mengenali sebuah citra, mengenali suara, hingga dapat meniru cara kerja otak manusia melalui saraf tiruan yang ada pada algoritmanya (Purmana, 2021). Deep learning memiliki fitur utama untuk melakukan ekstraksi pola yang berguna dari data untuk memudahkan model untuk membedakan kelas, fitur ini disebut dengan Feature Engineering (Purmana, 2021).

2.10. Convolutional Neural Network

Convolutional Neural Network (CNN) adalah jenis khusus dari jaringan saraf yang dirancang untuk memproses data dengan topologi jala atau *grid-like topology*. Nama "*Convolutional Neural Network*" menunjukkan bahwa jaringan ini memanfaatkan operasi matematika yang disebut konvolusi. Konvolusi adalah sebuah operasi linear yang digunakan untuk menggabungkan dua fungsi. Dengan demikian, *Convolutional Network* adalah jaringan saraf yang menerapkan konvolusi setidaknya pada satu lapisannya. CNN sangat efektif dalam tugas-tugas seperti pengenalan gambar dan analisis video karena kemampuannya untuk menangkap pola spasial dan temporal dalam data (Lecun et al., 2015).

Algoritma ini telah terbukti sangat efektif dan telah diadopsi secara luas dalam berbagai bidang, termasuk pengenalan suara (*speech recognition*), pengenalan wajah (*face recognition*), pemrosesan bahasa alami (*natural language processing*), analisis gelombang otak (*brain wave analysis*), dan masih banyak lagi.

Di bidang pengenalan suara, CNN digunakan untuk meningkatkan akurasi transkripsi audio ke teks, memungkinkan sistem asisten virtual dan perangkat lunak pengenalan suara menjadi lebih canggih dan responsif. Dalam pengenalan wajah, algoritma ini digunakan untuk mengidentifikasi dan memverifikasi identitas individu dengan tingkat akurasi yang sangat tinggi, mendukung aplikasi keamanan dan pengawasan.

Selain itu, algoritma CNN juga telah menemukan aplikasi dalam bidang-bidang lain seperti pengenalan objek dalam gambar, analisis video, deteksi anomali, dan bahkan dalam seni dan musik untuk pengenalan dan generasi pola. Perkembangan dan penerapan CNN yang begitu luas menunjukkan potensi besar algoritma ini dalam memecahkan berbagai masalah kompleks dan meningkatkan berbagai teknologi yang kita gunakan sehari-hari. *Convolutional Neural Network* merupakan kombinasi dari *Deep Learning* dan *Artificial Neural Network*, dan secara umum digunakan dalam pengolahan gambar/Image processing (Tian, 2020).

2.11. Image Processing

Image Processing atau Pengolahan citra adalah cabang ilmu yang berfokus pada manipulasi dan analisis citra digital. Tujuannya adalah untuk meningkatkan kualitas citra, menghilangkan noise, memperbaiki kontras, dan mengubah citra agar lebih mudah diinterpretasi oleh manusia atau komputer. Dalam pengolahan citra, masukan berupa citra (gambar) dan keluaran juga berupa citra yang telah mengalami transformasi atau perbaikan (Hidayatullah, 2017).

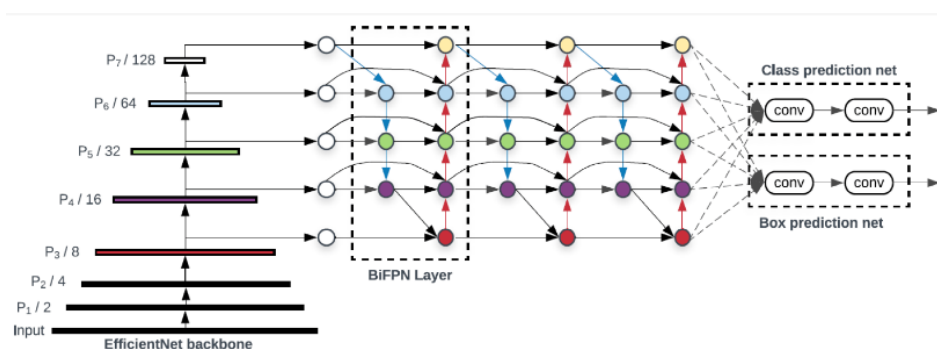
Citra atau gambar merupakan representasi visual yang tersusun dari angka-angka. Dari segi estetika, citra atau gambar dapat dianggap indah karena mengandung kumpulan warna, pola yang menarik, bentuk-bentuk abstrak, dan elemen visual lainnya. Contoh dari citra bisa berupa foto udara yang menampilkan pemandangan dari atas, penampang lintang (*cross section*) dari suatu objek, gambar wajah yang merekam ekspresi seseorang, hasil tomografi otak yang menggambarkan struktur otak, serta

banyak lagi jenis citra lainnya. Dalam konteks yang lebih luas, citra atau gambar memainkan peran penting dalam berbagai bidang seperti ilmu pengetahuan, seni, kedokteran, dan teknologi informasi, menjadi sarana untuk menyampaikan informasi, mengekspresikan kreativitas, serta memperluas pemahaman visual tentang dunia di sekitar kita.

2.12. Algoritma EfficientDet

EfficientDet adalah sebuah algoritma deteksi objek yang efisien yang dikembangkan oleh para peneliti di Google. EfficientDet menggunakan arsitektur backbone yang didasarkan pada *EfficientNet* seperti terlihat pada gambar 2.4, yaitu sebuah arsitektur jaringan saraf konvolusi yang juga dikembangkan oleh Google. Arsitektur *EfficientNet* mampu mencapai keseimbangan antara akurasi dan efisiensi komputasional.

Untuk meningkatkan kualitas deteksi, EfficientDet menggunakan teknik multi-level feature fusion. Hal ini memungkinkan model untuk menggabungkan informasi dari berbagai tingkat fitur dalam gambar untuk mengenali objek dengan lebih baik. EfficientDet telah terbukti memberikan performa yang sangat baik dalam berbagai tugas deteksi objek, termasuk dalam kompetisi COCO (Common Objects in Context). Model EfficientDet mampu mencapai tingkat akurasi yang sangat tinggi dan menjadi salah satu yang terbaik dalam kelasnya (Tan et al., 2020).



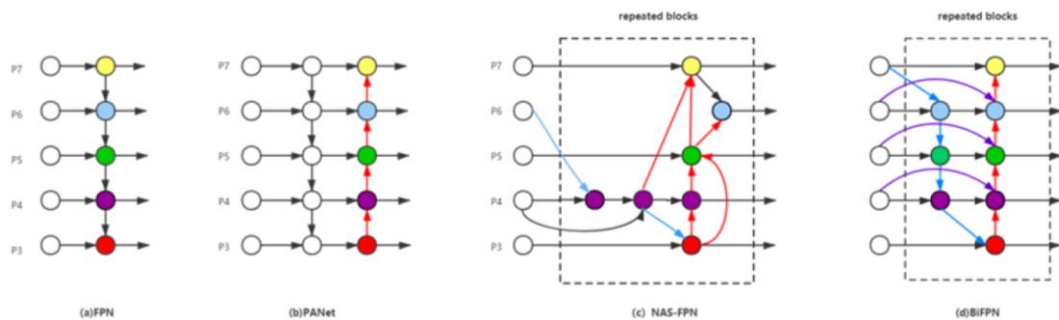
Gambar 2. 4 Gambaran sederhana algoritma EfficientDet

(Tan et al., 2020)

Algoritma EfficientDet dibagi menjadi tiga bagian yaitu bagian backbone, neck, dan prediction. Bagian backbone menggunakan algoritma EfficientNet untuk melakukan perubahan skala pada input gambar. Bagian neck merupakan arsitektur *Bi-directional Feature Pyramid Network (BiFPN)* yang melakukan sampling objek untuk setiap tingkat, dimulai dari tingkat P_3 sampai P_7 dan akan melakukan prediksi objek.

Setelah itu, akan didapatkan sebuah hasil prediksi berupa *box prediction*. EfficientDet menggunakan teknik *transfer learning* untuk mempercepat proses pembelajaran. *Transfer learning* adalah teknik dimana model yang sudah dilatih pada satu dataset, kemudian diaplikasikan pada dataset yang lain dengan harapan performa model tersebut akan lebih baik daripada model yang dilatih dari awal pada dataset yang baru.

a. *BiFPN*



Gambar 2. 5 Feature Network Design

(Cao et al., 2020)

FPN tradisional mengumpulkan karakteristik multi-skala dari atas ke bawah, seperti yang ditunjukkan pada Gambar 2.5 (a), peta fitur keluaran dari lapisan 7 diperoleh setelah konvolusi peta fitur masukan dari lapisan 7. Peta fitur keluaran dari lapisan 6 dapat diperoleh dengan mengkonvolusi peta fitur fusi yang diperoleh dengan melakukan up-sampling pada peta fitur keluaran dari lapisan 7 dan menambahkan peta fitur masukan dari lapisan 6. Dan seterusnya, peta fitur keluaran dari lapisan 3 dapat diperoleh dengan mengkonvolusi peta fitur fusi yang diperoleh dengan melakukan *up-sampling* pada peta fitur keluaran dari lapisan 4 dan menambahkan peta fitur masukan dari lapisan 3. FPN konvensional menggabungkan fitur multi-skala dengan cara *top-down*:

$$P_{out\ 7} = \text{Conv}(P_{in\ 7})$$

$$P_{out\ 6} = \text{Conv}(P_{in\ 6} + \text{Resize}(P_{out\ 7}))$$

...

$$P_{out\ 3} = \text{Conv}(P_{in\ 3} + \text{Resize}(P_{out\ 4}))$$

Resize biasanya merupakan operasi *up-sampling* atau *down-sampling* untuk pencocokan resolusi, dan Conv biasanya merupakan operasi konvolusi untuk pemrosesan fitur.

Seperti yang ditunjukkan pada Gambar 2.5 (b), PANet meningkatkan metode fusi fitur FPN, tidak hanya mengadopsi metode *top-down*, tetapi juga metode *bottom-up*. Gambar 2.5 (c) menunjukkan bahwa topologi jaringan karakteristik tidak teratur, NAS-FPN, ditemukan dengan menerapkan metode pencarian arsitektur neural (NAS). Untuk mendapatkan hasil ini, dibutuhkan waktu komputasi GPU yang besar.

BiFPN yang ditunjukkan pada Gambar 2.5 (d) dioptimalkan dalam tiga cara: yang pertama, Jika sebuah node (peta fitur) hanya memiliki satu input, kontribusinya terhadap fusi fitur kecil dan dapat dihapus. Yang kedua, Koneksi lompatan dibangun di setiap level untuk menggabungkan lebih banyak fitur dengan biaya rendah. Dan yang ketiga, Diusulkan bahwa setiap BiFPN dapat disambungkan sebagai modul, dan output dari BiFPN sebelumnya dapat digunakan sebagai input dari BiFPN berikutnya. Jumlah spesifik struktur semacam itu yang diperlukan tergantung pada situasi.

b. Tiga Metode Fusi Fitur

Ketika fitur dari skala yang berbeda digabungkan, praktik umum adalah menyamakan skala terlebih dahulu, dan kemudian menambahkan fitur yang sesuai. Ini mengasumsikan bahwa bobot fitur yang berbeda terhadap fitur fusi akhir adalah sama. Padahal, sebenarnya fitur input yang berbeda harus memberikan kontribusi yang berbeda pada fitur fusi akhir karena resolusi mereka yang berbeda. Oleh karena itu, tiga metode fusi fitur berbobot diusulkan:

Fusi tanpa batas:

$$O = \sum_i w_i \cdot I_i$$

Dimana w_i adalah bobot yang dapat dipelajari yang bisa berupa skalar (per-fitur), vektor (per-saluran), atau tensor multi-dimensi (per-piksel). Bobot skalar adalah yang paling murah untuk dihitung tanpa kehilangan presisi yang jelas, sehingga bobot skalar diadopsi. Kekurangan dari metode pembobotan ini adalah tidak ada batasan pada bobot, sehingga pelatihan model mungkin tidak stabil.

Fusi berbasis Softmax:

$$O = \sum_i \frac{e^{w_i}}{\sum_j e^{w_j}} \cdot I_i$$

Rumus ini menormalkan bobot dari rumus sebelumnya, tetapi metode ini memiliki kekurangan yaitu meningkatkan jumlah komputasi. Untuk meminimalkan biaya latensi ekstra, Peneliti lebih lanjut mengusulkan pendekatan fusi cepat.

Fusi normalisasi cepat:

$$O = \sum_i \frac{w_i}{\varepsilon + \sum_j w_j} \cdot I_i$$

Dimana $w_i \geq 0$ dijamin dengan menerapkan Relu setelah setiap w_i , dan $\varepsilon = 0.0001$ (epsilon) adalah nilai kecil untuk menghindari ketidakstabilan numerik. Studi ablation menunjukkan bahwa hasil metode pembobotan ini mirip dengan fusi fitur softmax, dan kecepatan komputasi di GPU meningkat sebesar 30%.

c. Skala Gabungan

Dengan tujuan mengoptimalkan baik akurasi maupun efisiensi, kami ingin mengembangkan sebuah keluarga model yang dapat memenuhi berbagai batasan sumber daya. Tantangan utama di sini adalah bagaimana meningkatkan model baseline EfficientDet.

Jaringan Backbone – menggunakan EfficientNet-D1, dengan lebar dan kedalaman yang sama untuk tujuh jaringan tersebut.

Jaringan BiFPN – Lebar jaringan BiFPN meningkat secara eksponensial sementara kedalamannya meningkat secara linier, yang memenuhi rumus berikut:

$$W_{bifpn} = 64 \cdot (1.35^\phi), D_{bifpn} = 3 + \phi$$

Jaringan prediksi kotak/kelas – Lebar jaringan ini sama dengan keluaran bagian BiFPN sebelumnya. Namun, kedalamannya memenuhi rumus berikut:

$$D_{\text{box}} = D_{\text{class}} = 3 + \lfloor \phi/3 \rfloor$$

Resolusi gambar input – Karena input BiFPN adalah lapisan P3 hingga P7 dari jaringan backbone yang digunakan, resolusi gambar input harus dapat dibagi dengan 2 pangkat 7. Resolusi gambar harus memenuhi rumus berikut:

$$R_{\text{input}} = 512 + \phi \cdot 128$$

Adapun kelebihan-kelebihan Algoritma EfficientDet yaitu:

1. Memiliki nilai akurasi yang baik, EfficientDet dirancang untuk memberikan kinerja yang sangat baik dengan menggunakan sumber daya yang lebih sedikit. Model EfficientDet mampu mencapai tingkat akurasi yang tinggi dengan jumlah parameter yang relatif kecil dibandingkan dengan pendekatan deteksi objek lainnya.
2. Skala Resolusi, Algoritma ini memiliki kemampuan untuk mendeteksi objek pada berbagai skala resolusi gambar. Dengan menggunakan teknik penskalaan multi-level, EfficientDet dapat mengenali objek kecil maupun objek besar dengan baik.

2.13. Penelitian Terdahulu

Adapun penelitian terdahulu yang digunakan pada penelitian ini sebagaimana terlihat pada Tabel 2.1.

Tabel 2. 1 Penelitian Terdahulu

No.	Penelitian	Judul	Metode	Keterangan
1.	Susanto L (2023)	Klasifikasi Citra Lesi Kulit Serupa virus Monkeypox Menggunakan VGG-19 Convolutional Neural Network	VGG-19 pada arsitektur CNN	Melakukan penelitian tentang virus cacar monyet (Monkeypox). Hasil yang diperoleh diperoleh tingkat akurasi sebesar 76%, sensitivitas sebesar 76%, dan spesifitas sebesar 92%.
2.	Victory G (2023)	Klasifikasi Monkeypox dan	Ekstraksi LBP dan	Hasil pengujian menggunakan fitur ekstraksi

		non Monkeypox Menggunakan Fitur Ekstraksi LBP dan Algoritma Random Forest	Random Forest	LBP dan metode Random Forest terhadap kelas monkeypox dan non- monkeypox dengan proporsi dataset 80:20 mendapatkan n_estimator terbaik yaitu 500 dengan nilai accuracy tertinggi sebesar 85%.
3.	Wijaya W (2023)	Klasifikasi Monkeypox Menggunakan Ekstraksi Fitur GLCM dan Algoritma Random Forest	Ekstraksi GLCM dan Random Forest	Mengklasifikasikan Monkeypox. Dari empat pengujian yang sudah dilakukan, disimpulkan bahwa n_estimator terbaik adalah 100 dengan proporsi rasio dataset 90:10, yang menghasilkan nilai akurasi sebesar 77%, precision 77% ,recall 77%, dan f1-score 76,5%.
4.	(Nayak et al., 2023)	<i>Deep learning based detection of monkeypox virus using skin lesion images</i>	<i>Ensemble Learning (Inception V3, Xception, dan DenseNet169)</i>	Melakukan penelitian Deteksi virus Monkeypox berdasarkan citra gambar lesi kulit menggunakan Deep learning. Hasil penelitian pada dataset gambar lesi kulit monkeypox mencapai rata- rata akurasi 93,39%.
5.	Sahin V (2022)	<i>Human Monkeypox Classification from Skin Lesion Images with Deep</i>	<i>Tenserflow Lite dengan Deep Pre- Trained</i>	Penelitian ini menggunakan kamera kemudian dikirim ke jaringan saraf konvolusional

		<i>Pre-trained Network using Mobile Application</i>		dalam perangkat yang sama. Jaringan kemudian mengklasifikasikan gambar sebagai positif atau negatif untuk deteksi monkeypox. Hasil yang diperoleh sistem ini dapat mengklasifikasikan gambar dengan akurasi 91,11%.
6.	Nainggolan (2023)	Implementasi Algoritma EfficientDet untuk Klasifikasi Penyakit Alzheimer berdasarkan citra MRI Otak	EfficientDet	Penelitian ini mengklasifikasikan 4 penyakit Alzheimer dengan jumlah dataset 196 data dan hasil akurasi 97,4%

Metode yang akan digunakan dalam penelitian ini membedakan dari penelitian sebelumnya yaitu penelitian ini akan menggunakan EfficientDet untuk deteksi penyakit cacar monyet. Berbeda dengan penelitian sebelumnya yang dilakukan oleh Susanto (2023) menggunakan metode VGG 19 dan hanya mengklasifikasikan cacar dengan output Monkeypox dan bukan Monkeypox saja. Sedangkan Victory (2023) dan Wijaya (2023) sama-sama menggunakan metode Random Forest namun berbeda pada pemilihan ekstraksi fitur dengan menggunakan LBP dan GLCM, namun juga hanya memberikan output Monkeypox dan bukan Monkeypox saja. Sedangkan penelitian yang akan dibuat tidak hanya mengeluarkan output penyakit Cacar Monyet saja, melainkan juga mengeluarkan output penyakit kulit serupa lainnya yaitu, Cacar air, Hand foot mouth disease, dan Normal. Juga Penelitian sebelumnya berbasis Web saja, sedangkan penelitian ini akan dibuat berbasis mobile Android. Penelitian lain yang telah menggunakan mobile dilakukan oleh Idris (2023) dan Sahin (2022) dinilai belum cukup efektif untuk menghasilkan model terbaik dalam mendeteksi cacar monyet.

BAB 3

ANALISIS DAN PERANCANGAN

3.1 Dataset yang digunakan

Data yang digunakan dalam penelitian ini adalah data lesi kulit penyakit Cacar Monyet dan penyakit kulit serupa yaitu Cacar Air, *Hand Foot Mouth Disease* (HFMD), dan Normal, data ini telah dicari oleh penulis dan didapatkan dari situs yang disediakan dari situs Kaggle dan Dermnet. Adapun total dari keseluruhan data yang dikumpulkan yaitu sebanyak 4.856 gambar.

Selanjutnya peneliti akan membagi menjadi tiga bagian data yaitu data *training*, *testing* dan data *validasi* data tersebut dapat dilihat pada Tabel 3.1. Untuk data *training* sendiri diterapkan dalam proses pembelajaran algoritmanya dalam mendeteksi jenis penyakit kulitnya. Sementara data *testing* digunakan untuk memverifikasi kebenaran dan kinerja dari sistem. Adapun untuk pembagian data *training* sebesar 3.714 data, data *testing* sebesar 80 data dan data validasi sebesar 1.062 data, seluruh data berekstensi Joint Photographic Group (JPG) dengan size 640x640.

Tabel 3. 1 Jumlah Data

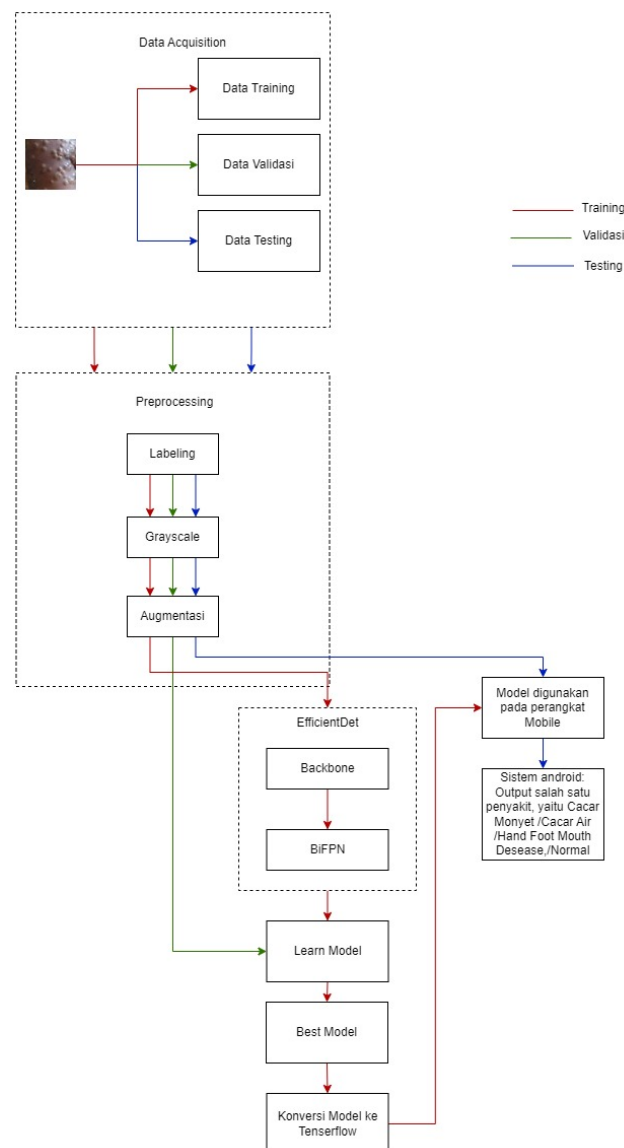
NO.	Dataset	Jumlah Data <i>Training</i>	Jumlah Data Validasi	Jumlah Data <i>Testing</i>
1.	Cacar Monyet	1.500	452	20
2.	Cacar Air	464	130	20
3.	HFMD	950	280	20
4.	Normal	800	200	20

Dari tabel 3.1 jumlah keseluruhan data *training* adalah 3.714, data validasi adalah 1.062 dan data *testing* adalah 80 sehingga total keseluruhan data adalah 4.856.

3.2 Analisis Sistem

Dalam penelitian ini telah dilakukan beberapa langkah-langkah dalam kasus deteksi penyakit Cacar Monyet dan penyakit kulit serupa yaitu Cacar Air, *Hand Foot Mouth Disease* (HFMD). Langkah pertama yang dilakukan adalah mengumpulkan data dalam bentuk citra lesi kulit berekstensi *Joint Photographic Group* (JPG). Kemudian dilakukan pembagian dataset yaitu data training, data validasi, dan data testing. Selanjutnya dilakukan tahap *pre-processing* terhadap dataset, seluruh data *training* kemudian dilakukan *labelling*, *grayscale*, dan Augmentasi data.

Tahapan-tahapan tersebut telah detail dijelaskan pada arsitektur umum yang ada pada gambar 3.1.



Gambar 3. 1 Arsitektur Umum

3.2.1 Image Acquisition

Tahap ini merupakan tahap pengambilan input, yang kemudian datanya akan dibagi menjadi 3 yaitu, *Data Training*, *Data Testing*, dan *Data Validasi*. *Data testing* adalah kumpulan data yang akan dilatih menggunakan model. Sedangkan, *Data Validasi* Adalah kumpulan data yang berisi sampel berbeda untuk mengevaluasi model yang telah terlatih. *Data testing* adalah data yang digunakan untuk percobaan dan melihat performa keakuratan algoritma yang sudah dilatih.

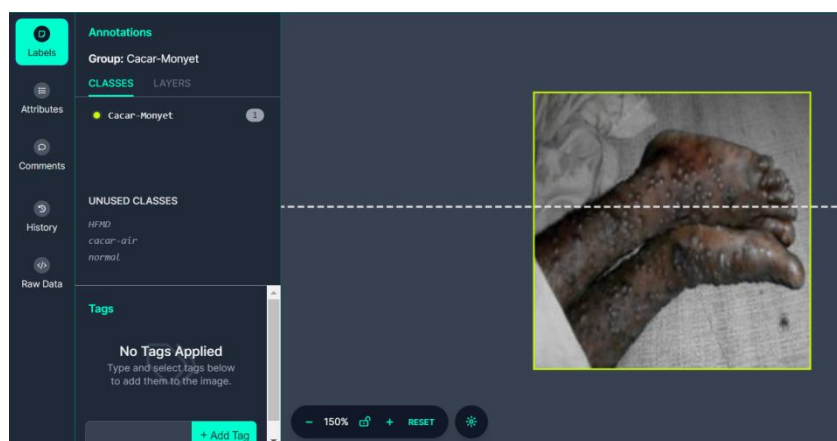
Saat proses *training* selesai dilakukan dengan menggunakan algoritma *EfficientDet*, maka hasil yang diharapkan adalah pembentukan model dari proses *training*.

3.2.2 Pre-processing

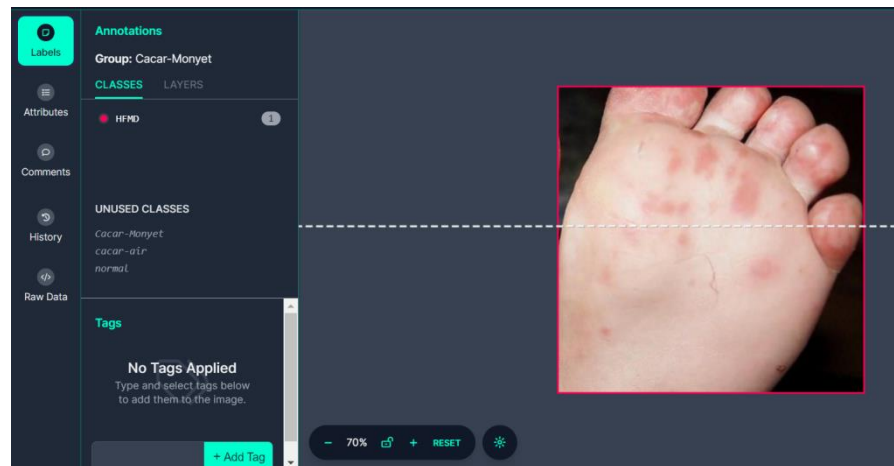
Tahap *Pre-processing* digunakan untuk mempermudah data pada saat diproses dan meningkatkan akurasi dari hasil *training*. Seluruh tahap pra pemrosesan dilakukan melalui platform Roboflow Beberapa tahapan yang dilakukan pada *pre-processing* yaitu :

a. Labelling

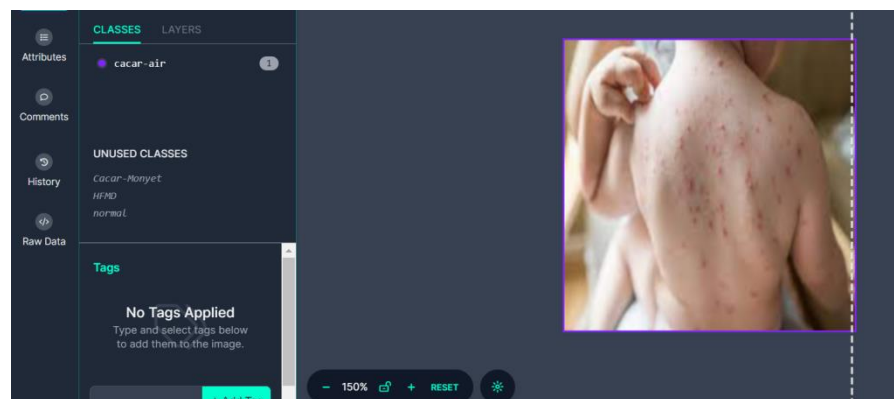
Pelabelan gambar merupakan langkah awal dalam proses preprocessing data yang mana dataset akan diberi label sesuai dengan kategori atau class objek. Data gambar yang telah terkumpul akan diberikan label satu per satu dengan menggunakan aplikasi Roboflow, dengan label sesuai dengan 4 dataset yaitu “Cacar Monyet” , “Cacar Air”, “HFMD”, dan “Normal”. Langkah yang dilakukan untuk melakukan *labelling* dapat dilihat pada Gambar 3.2, Gambar 3.3, Gambar 3.4. dan pada Gambar 3.5.



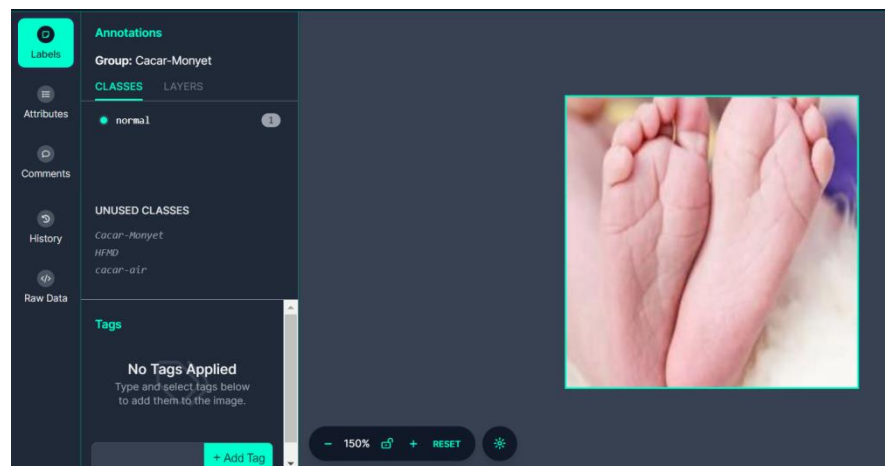
Gambar 3. 2 Label Cacar Monyet



Gambar 3. 3 Label HFMD



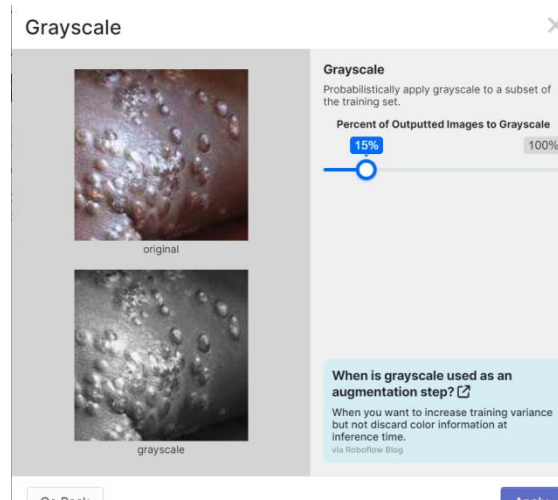
Gambar 3. 4 Label Cacar Air



Gambar 3. 5 Label Normal

b. *Grayscale*

Mengacu pada representasi citra yang hanya memiliki tingkat keabuan (nilai intensitas) tanpa informasi warna. Citra *grayscale* dihasilkan dengan mengonversi citra berwarna menjadi citra dengan satu saluran intensitas saja.

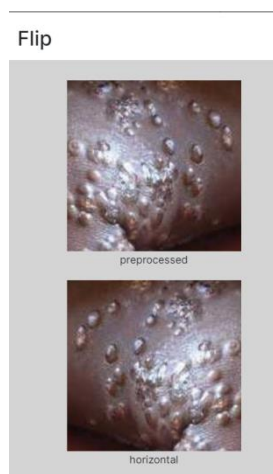


Gambar 3. 6 Grayscale

c. Augmentasi data

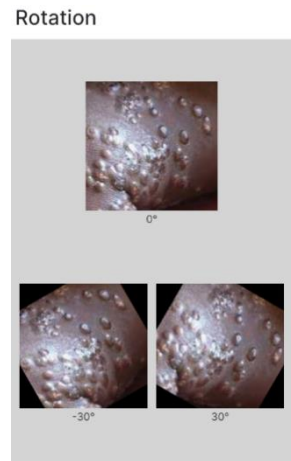
Augmentasi data adalah teknik yang umum digunakan dalam pembelajaran mesin untuk meningkatkan jumlah dan variasi data pelatihan dengan membuat modifikasi pada data pelatihan yang ada. Tujuan utamanya adalah untuk meningkatkan kinerja dan generalisasi model dengan memperkenalkan variasi yang lebih besar dalam dataset training.

Melakukan proses augmentasi dengan flip horizontal, proses ini ditunjukkan pada Gambar 3.7.



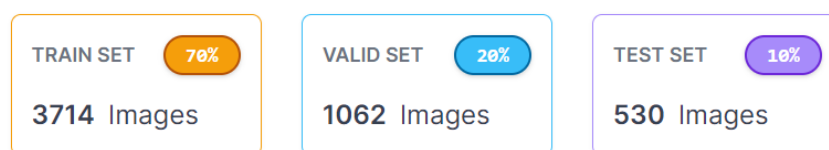
Gambar 3. 7 Proses Flip

Kemudian ada proses augmentasi berupa rotasi juga dilakukan pada citra di mana Citra diputar sebesar 30° ke kiri dan 30° ke kanan rotasi Citra dapat dilihat pada gambar 3.8



Gambar 3. 8 Rotasi

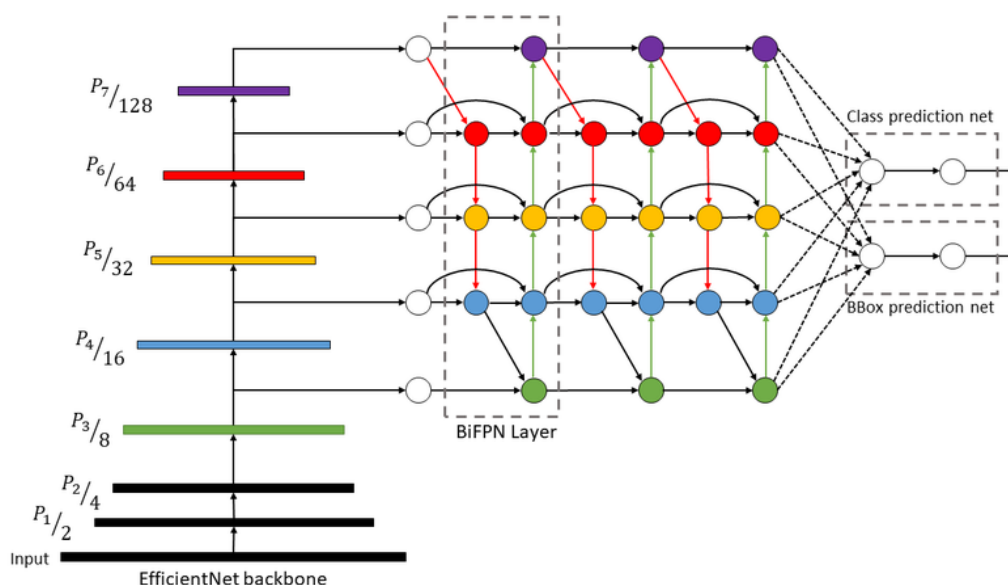
Setelah dilakukan augmentasi data citra bertambah 1.314, sebagai contoh data citra dari training data yang sebelumnya berjumlah 2400 data, menjadi 3.714 data, yang kemudian dibagi menjadi 70% untuk data training dan 20% untuk data validation dan 10% data testing. Dimana pembagian ini menurut (Wicaksono et al., 2021) pada penelitiannya membagi dataset dengan presentase serupa menghasilkan akurasi yang cukup baik sehingga peneliti membagi presentase dataset dengan pembagian yang sama.



Gambar 3. 9 Data Setelah Augmentasi

3.2.3 *Process*

Setelah data Citra melalui proses pre-processing, lalu data akan diproses dengan menggunakan algoritma deteksi yaitu EfficientDet D1. Berikut adalah tahapan utama dalam proses deteksi gambar menggunakan algoritma EfficientDet D1:



Gambar 3. 10 Arsitektur Backbone EfficientDet

Berikut adalah tahapan utama dalam proses deteksi gambar menggunakan algoritma EfficientDet D1:

a. Backbone Network

EfficientDet d1 menggunakan EfficientNet sebagai jaringan backbone. EfficientNet adalah model klasifikasi gambar yang sangat efisien dan mampu mengekstrak fitur-fitur penting dari gambar input. EfficientNet adalah jaringan klasifikasi yang diusulkan pada tahun 2019. Menambah kedalaman jaringan saraf, memperluas lebar lapisan fitur, dan meningkatkan resolusi gambar input untuk meningkatkan akurasi deteksi jaringan.

Untuk meningkatkan efisiensi deteksi target, sangat penting untuk menyeimbangkan tiga dimensi lebar, kedalaman, dan resolusi jaringan. EfficientNet dengan cerdas menggabungkan ketiganya fitur ini dan mengusulkan strategi penskalaan model baru. Pertama, *EfficientNet* menambah garis dasar lebar jaringan dengan menggunakan jumlah kernel konvolusi yang lebih tinggi di setiap konvolusi lapisan untuk memperluas jumlah saluran matriks fitur. Kedua, kedalaman jaringan adalah ditingkatkan dengan menambahkan lapisan yang lebih besar ke jaringan dasar. Kemudian, resolusi gambar input dinaikkan, dan tinggi serta lebar setiap matriks fitur ditingkatkan. Akhirnya, lebar, kedalaman, dan resolusi jaringan dasar ditingkatkan secara bersamaan. Dengan menggabungkan strategi penskalaan ini, menunjukkan peningkatan dengan tetap mempertahankan yang setara

jumlah jepit. Hasil eksperimen menunjukkan bahwa menggabungkan beberapa dimensi adalah metode penskalaan terbaik. Penyesuaian ini didasarkan pada teknik pencarian struktur saraf (Liu et al., 2024), yang dapat memperoleh koefisien komposit yang optimal. Melalui strategi penskalaan ini, *EfficientNet* dapat mencapai

b. BiFPN (*Bidirectional Feature Pyramid Network*)

Fitur-fitur yang diekstrak oleh *EfficientNet* diteruskan ke BiFPN. BiFPN adalah struktur jaringan yang dirancang untuk menggabungkan fitur-fitur dari berbagai level resolusi secara efisien. Ini memungkinkan model untuk mendeteksi objek dengan ukuran yang bervariasi.

c. *Head Network*

Head network terdiri dari dua sub-jaringan: satu untuk deteksi klasifikasi (menentukan jenis objek) dan satu lagi untuk *regresi bounding box* (menentukan lokasi objek). Setiap fitur dari BiFPN digunakan untuk memprediksi kelas dan bounding box.

d. Post-processing (Pasca-pemrosesan)

Non-Maximum Suppression (NMS) - Proses ini dilakukan untuk menghilangkan prediksi bounding box yang tumpang tindih, hanya menyisakan bounding box dengan skor tertinggi untuk setiap objek.

Thresholding - Prediksi dengan skor kepercayaan di bawah ambang batas tertentu diabaikan.

d. Output

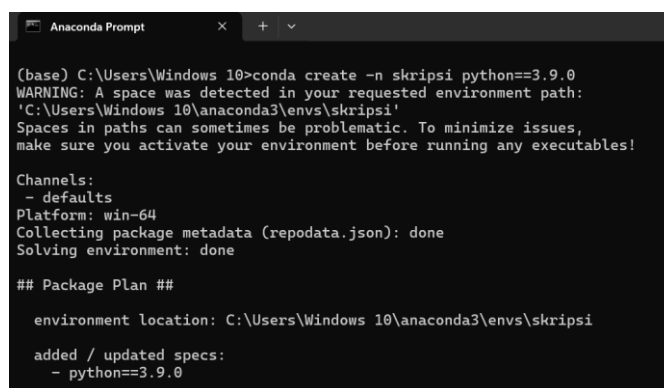
Hasil akhir adalah sekumpulan bounding box yang terdeteksi, masing-masing dengan label kelas dan skor kepercayaan.

3.2.4 *Proses Training*

Pada tahap ini akan dilakukan beberapa konfigurasi model dan parameter sebelum masuk ke proses training, data training ini nantinya akan digunakan untuk pembelajaran pada model *efficientdet*. Setelah model selesai, kemudian model akan diekspor ke *tenserflow*, yang nantinya akan bisa digunakan di aplikasi android.

1. Membuat *Environment*

Membuat environment dilakukan pada aplikasi Anaconda Prompt, yang penulis buat environmentnya adalah skripsi, Pengembangan aplikasi menggunakan tools yaitu anaconda prompt. Langkah yang dilakukan adalah membuka anaconda prompt, kemudian membuat environment baru menggunakan perintah “conda create -n skripsi python==3.9”, kemudian aktivasi environment tadi menggunakan “conda activate skripsi”. Dapat dilihat pada gambar 3.10



```
(base) C:\Users\Windows 10>conda create -n skripsi python==3.9.0
WARNING: A space was detected in your requested environment path:
'C:\Users\Windows 10\anaconda3\envs\skripsi'
Spaces in paths can sometimes be problematic. To minimize issues,
make sure you activate your environment before running any executables!

Channels:
- defaults
Platform: win-64
Collecting package metadata (repodata.json): done
Solving environment: done

## Package Plan ##

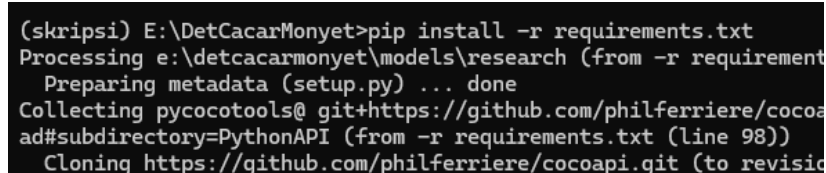
  environment location: C:\Users\Windows 10\anaconda3\envs\skripsi

added / updated specs:
- python==3.9.0
```

Gambar 3. 11 Environment

2. *Install dan Import Library*

selanjutnya adalah install library menggunakan “pip install -r requirements.txt”, library yang digunakan dapat dilihat pada gambar 3.11



```
(skripsi) E:\DetCacarMonyet>pip install -r requirements.txt
Processing e:\detcacarmonyet\models\research (from -r requirements.txt)
  Preparing metadata (setup.py) ... done
Collecting pycocotools@ git+https://github.com/philferriere/cocoapi.git#subdirectory=PythonAPI (from -r requirements.txt (line 98))
  Cloning https://github.com/philferriere/cocoapi.git (to revisio
```

Gambar 3. 12 Install Requirements

b. Unduh data pada *Tfrecord* Roboflow API

Melakukan proses install roboflow, yang kemudian dataset yang ada di roboflow nanti akan dilakukan import. Dapat dilihat pada gambar 3.12.

```

!pip install roboflow

from roboflow import Roboflow
rf = Roboflow(api_key="UjrKoHbsddcteDuc6g3")
project = rf.workspace("skripsi-vjlbo").project("deteksi-penyakit-cacar-monyet")
version = project.version(2)
dataset = version.download("tfrecord")

```

Gambar 3. 13 Roboflow API

Data citra yang sudah melewati tahap labelling dan tahap pre-processing akan di export dengan format Tensorflow TFRecord dengan menggunakan API dari Roboflow. Selanjutnya data citra yang sudah diunduh dimap ke dalam variable yang nantinya akan digunakan untuk konfigurasi pipeline model training. Proses mapping data dapat dilihat pada gambar

c. Model Initialization

Model initialization block dalam Python merujuk pada metode `__init__()` yang digunakan untuk menginisialisasi objek dari suatu kelas. Fungsi `__init__()`: pada Python yaitu secara otomatis dipanggil saat penulis membuat sebuah objek dari suatu kelas. Setiap kelas harus memiliki parameter pertama `self`, yang merujuk pada objek dari kelas tersebut.

d. Training Proses

Dilakukan proses training, dengan jumlah dataset yang akan ditraining 5.306 data citra untuk melatih model yang diajukan. Peneliti melakukan training pada model Efficientdet-d1 dengan membedakan jumlah stepnya, yaitu dimulai dari 1.000 step sampai 30.000 step dengan batch size 4. terlihat pada gambar 3.13.

```

'loss/regularization_loss': 0.012803493,
'loss/total_loss': 0.18216985,
'learning_rate': 0.0}
I0628 21:50:21.132624 139877996564480 model_lib_v2.py:708] {'loss/classification_loss': 0.16606,
'loss/localization_loss': 0.003306366,
'loss/regularization_loss': 0.012803493,
'loss/total_loss': 0.18216985,
'learning_rate': 0.0}
INFO:tensorflow:Step 29700 per-step time 0.495s
I0628 21:51:10.684267 139877996564480 model_lib_v2.py:705] Step 29700 per-step time 0.495s
INFO:tensorflow:({'loss/classification_loss': 0.097666934,
'loss/localization_loss': 0.0004492683,
'loss/regularization_loss': 0.012803493,
'loss/total_loss': 0.1109197,
'learning_rate': 0.0}
I0628 21:51:10.684577 139877996564480 model_lib_v2.py:708] {'loss/classification_loss': 0.097666934,
'loss/localization_loss': 0.0004492683,
'loss/regularization_loss': 0.012803493,
'loss/total_loss': 0.1109197,
'learning_rate': 0.0}
INFO:tensorflow:Step 29800 per-step time 0.496s
I0628 21:52:00.325499 139877996564480 model_lib_v2.py:705] Step 29800 per-step time 0.496s
INFO:tensorflow:({'loss/classification_loss': 0.06780488,
'loss/localization_loss': 0.0003863167,
'loss/regularization_loss': 0.012803493,

```

Gambar 3. 14 Proses Training

3.3 Perancangan Sistem

Pada perancangan tahap ini, akan dijelaskan tentang perancangan sistem deteksi cacar monyet dan penyakit kulit serupa berbasis android, dimana pada tahap ini akan ditampilkan perancangan menu dan antarmuka dengan tujuan agar aplikasi dapat dijalankan dengan mudah oleh pengguna, dan pada akhir perancangan sistem akan memberikan output pada setiap gambar penyakit kulit yang dimasukkan.

3.3.1 *Install Node.js*

Dilakukan pengunduhan aplikasi Node.js pada laptop, Expo bergantung pada platform Node.js untuk tool command-line dan manajemen dependensi.

3.3.2 *Install Expo*

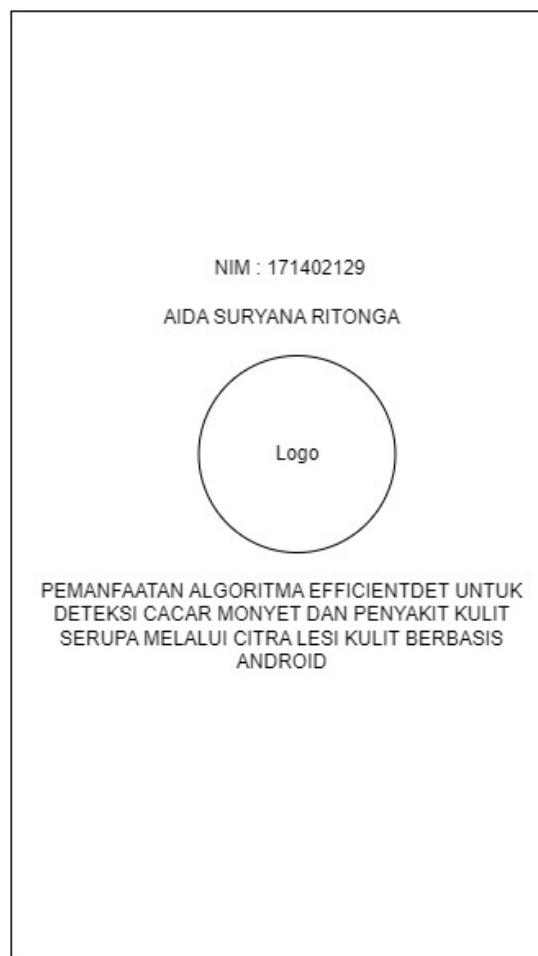
Melakukan penginstalan expo pada Handphone Android, yang nantinya akan digunakan untuk proses testing.

3.3.3 *Rancangan aplikasi*

Semua rancangan aplikasi ini menggunakan React Native, yang nantinya akan diexport ke Aplikasi Expo Go dengan menggunakan API Insomnia.

1. Rancangan Antarmuka Halaman *Splash Screen*

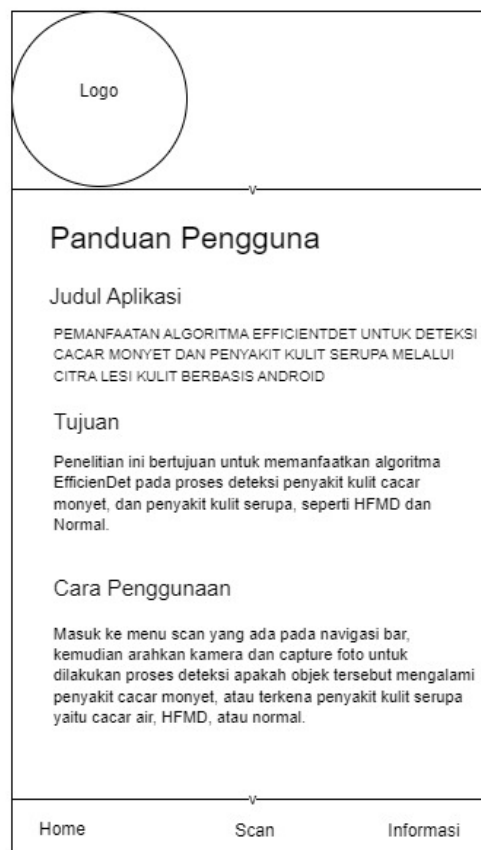
Pada halaman *splash screen* tampilan aplikasi yang pertama kali dilihat adalah tampilan judul penelitian, logo instansi, identitas penulis, dapat dilihat pada gambar 3.14.



Gambar 3. 14 Tampilan Splash Screen

2. Desain Tampilan Halaman Awal

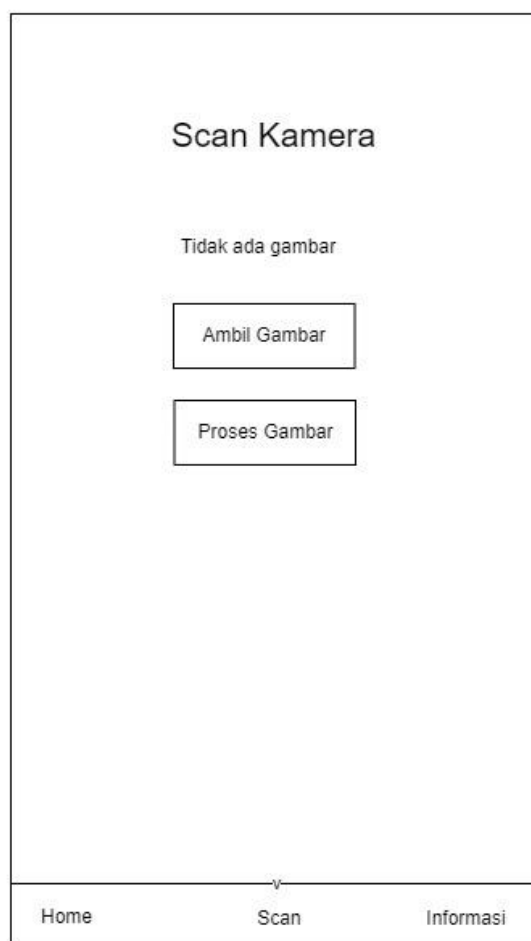
Pada halaman awal terdapat tab home, yang berisi panduan pengguna, ada judul aplikasi tujuan aplikasi, dan cara penggunaan aplikasi, dilihat pada gambar 3.15.



Gambar 3. 15 Tampilan Home

3. Desain Tampilan halaman *Scan*

Pada halaman *scan* terdapat tombol ambil gambar, dan proses gambar, dimana pada tombol ambil gambar setelah diklik akan ada menu pilihan, yaitu pilih dari galeri, atau buka kamera, jika foto dipilih dari galeri, maka pilih gambar penyakit yang ingin dideteksi atau langsung buka kamera, kemudian arahkan kamera ke kemudian *capture*, didapatkan gambar lesi penyakit. Kemudian klik menu proses gambar, untuk mengetahui hasil dari penyakit yang telah diinputkan, yang dapat dilihat pada gambar 3.12.



Gambar 3. 16 Tampilan Scan

4. Desain Tampilan Informasi

Pada halaman terdapat informasi berupa nama penulis, nama pembimbing 1 dan 2 dapat dilihat pada gambar 3.13.

<div>Logo</div>		
<h3>Informasi</h3> <p>Aida Suryana Ritonga 171402129</p> <p>Pembimbing 1 Ade Sarah Huzaifah S.Kom., M.Kom NIP : 198506302018032001</p> <p>Pembimbing 1 Fahrurrozi Lubis B.IT., M.Sc.IT NIP : 198610122018052001</p>		
Home	Scan	Informasi

Gambar 3. 17 Tampilan Informasi

BAB 4

IMPLEMENTASI DAN PENGUJIAN

4.1 Implementasi Sistem

4.1.1 Spesifikasi kebutuhan Hardware

Dalam penelitian ini, menggunakan spesifikasi perangkat keras yaitu dapat dilihat pada table 4.1.

Tabel 4. 1 Spesifikasi kebutuhan Hardware

No	Jenis Komponen	Komponen yang digunakan
1	Processor	Intel(R) Core(TM) i5-1035G1 CPU @ 1.00GHz 1.19 GHz
2	System type	64-bit operating system, x64-based processor
3	Memory (RAM)	4,00 GB (3,81 GB usable)
4	Penyimpanan	500 GB SSD

4.1.2 Spesifikasi kebutuhan Software

Dalam penelitian ini, menggunakan beberapa perangkat lunak, diantaranya:

1. Roboflow
2. Python
3. Tensorflow
4. Protoc
5. Insomnia
6. Git
7. Node.js
8. Expo Go

4.2 Implementasi Data

Dataset yang digunakan pada penelitian ini berupa citra lesi kulit. Dataset didapatkan melalui platform Kaggle dan Dermnet. Dari data yang terkumpul terdapat 4 objek yang akan dideteksi yaitu Cacar Monyet, Cacar Air, *Hand Foot Mouth Disease* (HFMD), dan terakhir kulit normal. Jumlah citra digital yang dikumpulkan sebanyak 4.856 gambar sebagai data *training*, data validasi sebanyak 1.062 dan 80 sebagai data *testing*, lalu untuk jumlah label dapat dilihat pada table 4.4.

Tabel 4. 2 Pembagian data training berdasarkan label

NO.	Dataset	Jumlah Data
1.	Cacar Monyet	1.500
2.	Cacar Air	464
3.	HFMD	950
4.	Normal	800
Jumlah Data		3.714

Tabel 4. 3 Pembagian data validasi berdasarkan label

NO.	Dataset	Jumlah Data
1.	Cacar Monyet	452
2.	Cacar Air	130
3.	HFMD	280
4.	Normal	200
Jumlah Data		1.062

Tabel 4. 4 Pembagian data validasi berdasarkan label

NO.	Dataset	Jumlah Data
1.	Cacar Monyet	20
2.	Cacar Air	20
3.	HFMD	20
4.	Normal	20
Jumlah Data		80

Contoh citra lesi kulit, berikut merupakan contoh data penyakit cacar monyet dapat dilihat pada Gambar 4.1.



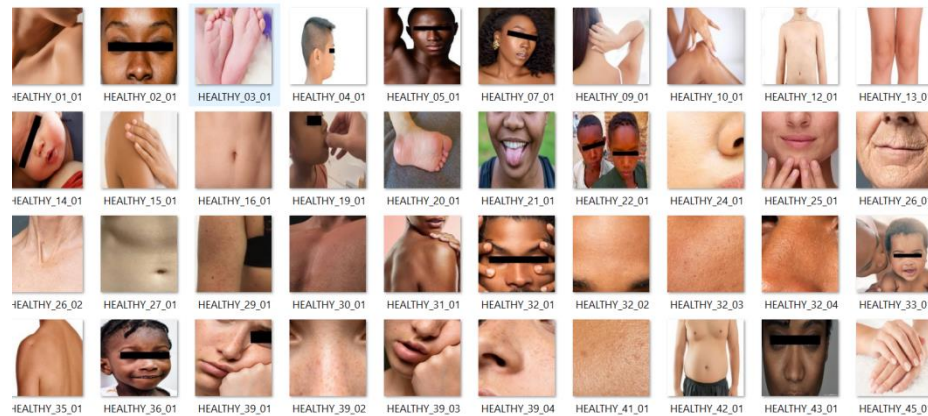
Gambar 4. 1 Data penyakit Cacar Monyet



Gambar 4. 2 Data penyakit Cacar Air



Gambar 4. 3 Data penyakit Hand foot mouth disease



Gambar 4. 4 Data kulit Normal

4.3 Training dengan Pemanfaatan Model EfficientDet

Dilakukan proses training pada model, peneliti akan melakukan konfigurasi dan penentuan hyperparameter terhadap model lalu didapatkan performa terbaik yang nantinya akan diterapkan ke dalam sistem. Adapun model yang digunakan pada penelitian ini yaitu EfficientDet-d1. Peneliti menentukan batch size senilai 4 saja. Dapat dilihat pada Gambar 4.5 yang merupakan konfigurasi pipeline untuk menentukan proses training.

```
train_config {
  batch_size: 4 # Increased batch size
  data_augmentation_options {
    random_horizontal_flip {
    }
  }
  data_augmentation_options {
    random_scale_crop_and_pad_to_square {
      output_size: 640
      scale_min: 0.8
      scale_max: 1.2
    }
  }
  sync_replicas: true
  optimizer {
    momentum_optimizer {
      learning_rate {
        cosine_decay_learning_rate {
          learning_rate_base: 0.01 # Adjusted learning rate
          total_steps: 20000 # Ensure enough training steps
          warmup_learning_rate: 0.001
          warmup_steps: 500
        }
      }
      momentum_optimizer_value: 0.9
    }
    use_moving_average: false
  }
  fine_tune_checkpoint: "training/efficientdet_d1_coco17_tpu-32/checkpoint/ckpt-0"
  num_steps: 30000 # Increased training steps
  startup_delay_steps: 0.0
  replicas_to_aggregate: 8
  max_number_of_boxes: 100
  unpad_groundtruth_tensors: false
  fine_tune_checkpoint_type: "detection"
  use_bfloat16: true
  fine_tune_checkpoint_version: V2
}
```

Gambar 4. 5 Konfigurasi pipeline (Train Config)

Dari konfigurasi tersebut dapat dirumuskan nilai epoch (iterasi) sebagai berikut :

$$\text{Epoch} = \text{num_steps} \times \text{batch_size} / \text{total_examples}$$

Dengan detail :

- num_steps = 30000
- batch_size = 4
- total_examples = 3714

jumlah epoch:

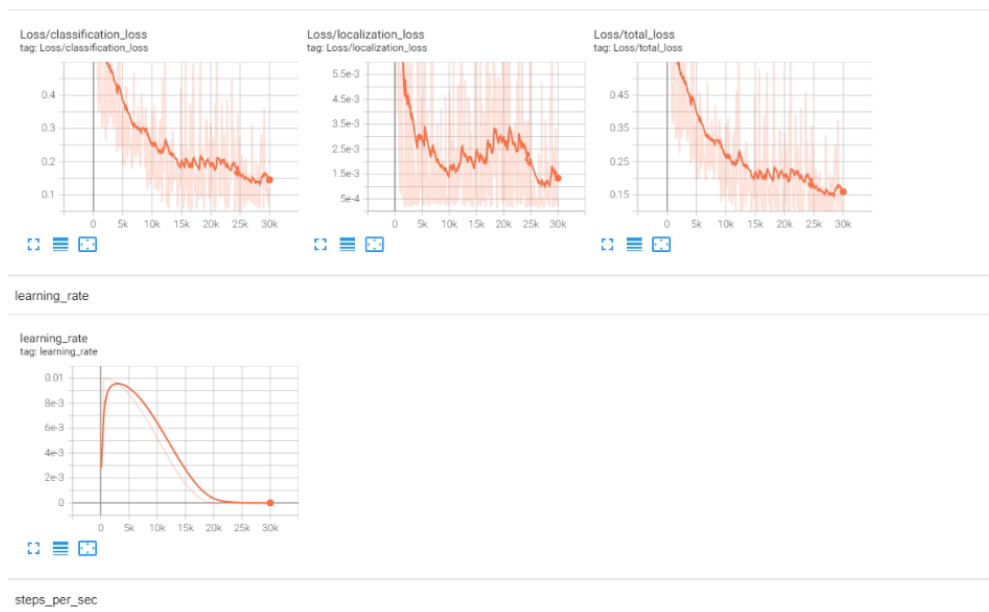
$$\text{Epoch} = 30000 \times 4 / 3714$$

$$\text{Epoch} = 120000 / 3714$$

$$\text{Epoch} = 32.3$$

Kemudian peneliti melakukan percobaan pelatihan dengan membedakan beberapa hyperparameter, yaitu jumlah steps, dimulai 24.100 (checkpoint dari 1.000) hingga 30.000 steps, dan ukuran batch 4. Seluruh proses pemindaian dilakukan menggunakan Google Colab. Percobaan ini memakan waktu yang berbeda-beda dan juga menghasilkan hasil yang berbeda-beda.

1. Step 24100 per-step time 1.177s INFO:tensorflow:{'Loss/classification_loss': 0.03653812, 'Loss/localization_loss': 0.00021618085, 'Loss/regularization_loss': 0.012803493, 'Loss/total_loss': 0.049557798, 'learning_rate': 0.0}
2. Step 24200 per-step time 0.489s INFO:tensorflow:{'Loss/classification_loss': 0.06896138, 'Loss/localization_loss': 0.00024041026, 'Loss/regularization_loss': 0.012803493, 'Loss/total_loss': 0.082005285,



Gambar 4. 6 Grafik Hasil Training

Dari grafik tersebut dapat dilihat bahwa nilai Loss semakin lama semakin menurun yang dimana menandai bahwa nilai accuracy semakin bertambah. Dengan nilai mAP seperti pada Gambar 4.7.



Gambar 4. 7 Nilai mAP

4.4 Implementasi Perancangan Aplikasi

Implementasi perancangan aplikasi antarmuka berdasarkan rancangan system yang sudah dijelaskan pada Bab 3 sebelumnya yaitu sebagai berikut:

a. Tampilan *Splash Screen*

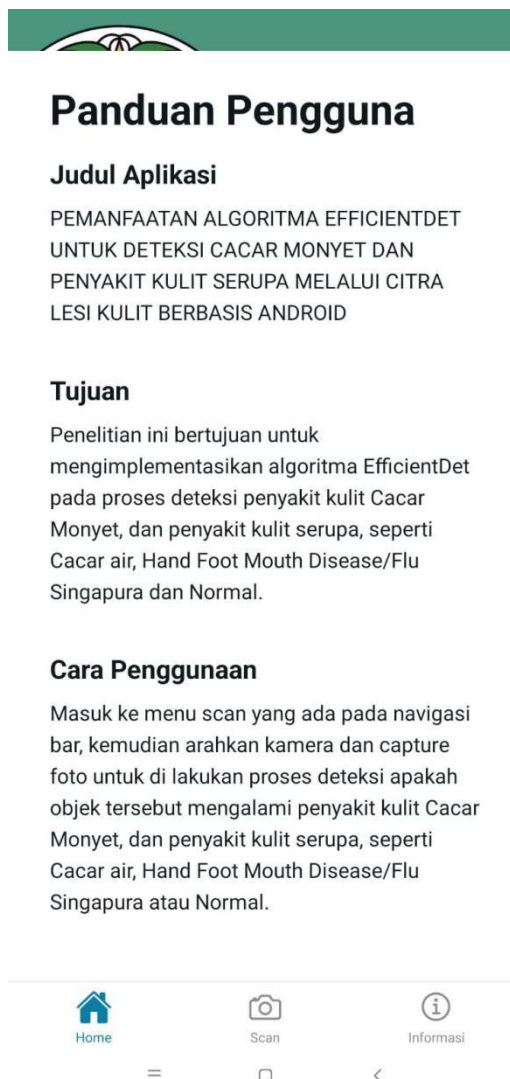
Saat membuka aplikasi, akan muncul tampilan layar sebelum masuk pada halaman utama atau *Home*. Layar awal yang muncul pada pembuka aplikasi ini disebut sebagai *splash screen*.



Gambar 4. 8 Tampilan Halaman Splash Screen

b. Tampilan Pada Halaman *Home*

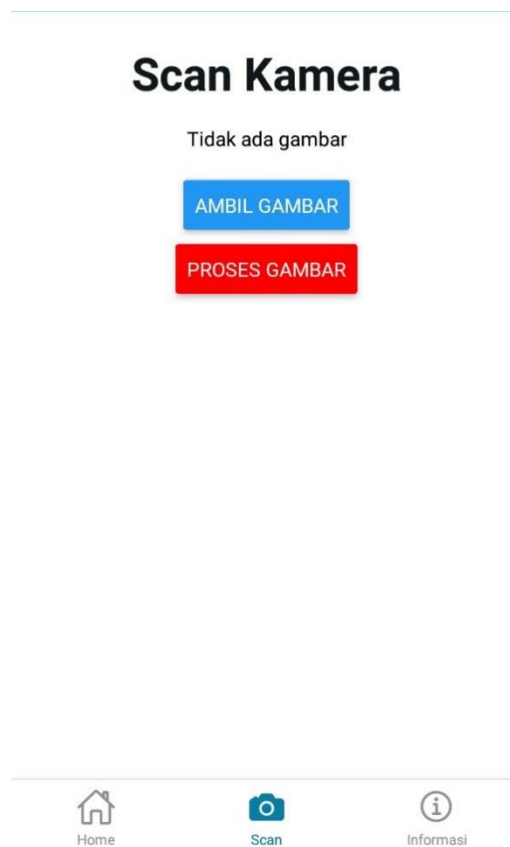
Tampilan awal aplikasi adalah tampilan *Home* yang muncul setelah *splash screen*, pada aplikasi ini terdapat judul tugas akhir, identitas pembuat aplikasi, dan terdapat juga cara penggunaan aplikasi. Ada 3 navigasi yaitu, *Home*, *Scan*, dan *Informasi*. Tampilan halaman depan dapat dilihat pada Gambar 4.9.



Gambar 4. 9 Tampilan Home

c. Tampilan pada Halaman *Scan*

Pada halaman *scan* terdapat tombol ambil gambar, dan proses gambar, dimana pada tombol ambil gambar setelah diklik akan ada menu pilihan, yaitu pilih dari galeri, atau buka kamera yang dapat dilihat pada gambar 4.10. Jika foto dipilih dari galeri, maka pilih gambar penyakit yang ingin dideteksi atau langsung buka kamera, kemudian arahkan kamera ke kemudian *capture*, didapatkan gambar lesi penyakit. Kemudian klik menu proses gambar, untuk mengetahui hasil dari penyakit yang telah diinputkan, yang dapat dilihat pada gambar 4.11.



Gambar 4. 10 Tampilan Scan Aplikasi

Scan Kamera



AMBIL GAMBAR

PROSES GAMBAR

Hasil Deteksi:

Class: normal, Score: 0.7856013774871826



Home



Scan



Informasi

Gambar 4. 11 Hasil Proses Gambar

- d. Tampilan Aplikasi Informasi Merupakan tampilan informasi berisi nama dan nomor induk mahasiswa penulis, nama pembimbing 1 dan 2 penulis, dapat dilihat pada Gambar 4.12.



Informasi 🖐️

Aida Suryana Ritonga

NIM : 171402129

Pembimbing 1

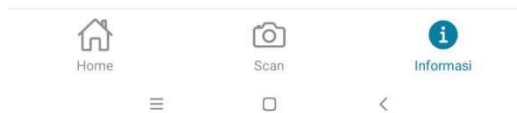
Ade Sarah Huzaifah S.Kom., M.Kom.

NIP : 198506302018032001

Pembimbing 2

Fahrurrozi Lubis B.IT., M.Sc.IT

NIP : 198610122018052001



Gambar 4. 12 Tampilan Informasi

4.5 Cara Penggunaan Aplikasi

Panduan Jarak Pengambilan Gambar Menggunakan EfficientDet pada Perangkat Android. Dalam menggunakan model EfficientDet untuk pengambilan gambar melalui perangkat Android, terdapat beberapa panduan yang perlu diikuti untuk memastikan hasil yang optimal dan akurasi yang tinggi. Panduan ini mencakup jarak ideal antara kamera dan objek yang akan difoto, baik untuk objek kecil maupun untuk seluruh tubuh manusia.

1. Pengambilan Gambar Objek Detail (Makro)

Untuk mendapatkan gambar yang detail dan jelas dari objek kecil, seperti ruam cacar monyet, cacar air, HFMD, dan normal disarankan agar pengambilan gambar dilakukan dengan jarak sangat dekat, yaitu sekitar 3 cm dari objek. Jarak ini memungkinkan kamera menangkap detail halus dan tekstur dari objek tersebut. Berikut beberapa tips yang perlu diperhatikan:

- a. Stabilitas Kamera: Pastikan kamera dalam keadaan stabil saat pengambilan gambar, gunakan tripod jika diperlukan untuk menghindari goyangan yang dapat mengaburkan detail.
- b. Pencahayaan yang Memadai: Pastikan pencahayaan cukup, jika perlu gunakan lampu tambahan atau flash untuk menghindari bayangan yang tidak diinginkan.
- c. Fokus Manual: Gunakan mode fokus manual pada kamera jika tersedia, untuk memastikan fokus tepat pada bagian objek yang diinginkan.

2. Pengambilan Gambar Seluruh Tubuh

Untuk pengambilan gambar yang mencakup seluruh tubuh manusia, jarak yang disarankan adalah tidak lebih dari 1 meter dari subjek. Jarak ini memungkinkan kamera menangkap seluruh tubuh secara proporsional tanpa distorsi. Berikut adalah beberapa hal yang perlu diperhatikan:





- a. Komposisi Gambar: Pastikan subjek berada di tengah frame dan ada ruang yang cukup di sekitar subjek untuk memberikan kesan proporsional dan seimbang.








- b. Pencahayaan yang Cukup: Gunakan pencahayaan alami atau lampu studio yang merata untuk menghindari bayangan yang mengganggu dan memastikan wajah serta tubuh subjek terlihat jelas.
- c. Latar Belakang: Pilih latar belakang yang sederhana dan tidak mengganggu perhatian dari subjek utama. Latar belakang yang terlalu ramai dapat mengalihkan fokus dari subjek.








4.6 Pengujian Sistem



Pada tahap ini sistem yang telah dibangun akan diuji yang mana tujuannya untuk menganalisa kemampuan dan kinerja sistem dalam mendeteksi penyakit Cacar Monyet dan penyakit kulit serupa yaitu, Cacar air, HFMD, dan Normal. Pengujian dilakukan terhadap 80 data. Hasil pengujian dapat dilihat pada tabel 4.5.

Tabel 4. 5 Hasil dari Pengujian Sistem

No.	Citra	Nama Citra	Output	Status
1.		Cacar Monyet	Cacar Monyet	Berhasil
2.		Cacar Monyet	Cacar Monyet	Berhasil
3.		Cacar Monyet	Cacar Monyet	Berhasil
4.		Cacar Monyet	Cacar Monyet	Berhasil

5.		Cacar Monyet	Cacar Monyet	Berhasil
6.		Cacar Air	Cacar Air	Berhasil
7.		Cacar Air	Cacar Air	Berhasil
8.		Cacar Air	Cacar Air	Berhasil
9.		Cacar Air	Cacar Air	Berhasil
10.		Cacar Air	Cacar Air	Berhasil
11.		HFMD	HFMD	Berhasil

12.		HFMD	HFMD	Berhasil
13.		HFMD	HFMD	Berhasil
14.		HFMD	HFMD	Berhasil
15.		HFMD	HFMD	Berhasil
16.		Normal	Normal	Berhasil
17.		Normal	Normal	Berhasil
18.		Normal	Normal	Berhasil

19.		Normal	Normal	Berhasil
20.		Normal	Normal	Berhasil

Dari Tabel 4.5 menunjukkan hasil pengujian sistem klasifikasi penyakit Cacar Monyet, dan penyakit kulit serupa, Cacar Air, HFMD, dan Normal sebanyak 80 data masing masing kelas 5 data sisanya berada di lampiran. Berdasarkan hasil pengujian terdapat 3 kesalahan hanya pada data Cacar Monyet, sedangkan yang lainnya berhasil terdeteksi oleh sistem. Sistem salah mendeteksi Cacar air padahal data sebenarnya adalah Cacar Monyet.

Tabel 4. 6 Tabel Confussion Matrix

Cacar Monyet	Cacar Air	HFMD	Normal	Total
17	3	0	0	20
0	20	0	0	20
0	0	20	0	20
0	0	0	20	20

Tabel 4. 7 Tabel Nilai TP, FP, dan FN

	TP	FP	FN
Cacar Monyet	17	0	3
Cacar Air	20	0	0
HFMD	20	0	0
Normal	20	0	0

Dengan perhitungan matematis dari

Confusion Matrixnya adalah sebagai berikut:

$$\begin{aligned}
 Precision &= \frac{TP \text{ Cacar Monyet}}{TP \text{ Cacar Monyet} + FP \text{ Cacar Monyet}} = \frac{17}{17} = 1 \\
 &= \frac{TP \text{ Cacar Air}}{TP \text{ Cacar Air} + FP \text{ Cacar Air}} = \frac{20}{20} = 1
 \end{aligned}$$

$$= \frac{TP_{HFMD}}{TP_{HFMD} + FP_{HFMD}} = \frac{20}{20} = 1$$

$$= \frac{TP_{Normal}}{TP_{Normal} + FP_{Normal}} = \frac{20}{20} = 1$$

$$Recall = \frac{TP_{Cacar\ Monyet}}{TP_{Cacar\ Monyet} + FN_{Cacar\ Monyet}} = \frac{17}{17+3} = 0,85$$

$$= \frac{TP_{Cacar\ Air}}{TP_{Cacar\ Air} + FN_{Cacar\ Air}} = \frac{20}{20} = 1$$

$$= \frac{TP_{HFMD}}{TP_{HFMD} + FN_{HFMD}} = \frac{20}{20} = 1$$

$$= \frac{TP_{Normal}}{TP_{Normal} + FN_{Normal}} = \frac{20}{20} = 1$$

Adapun berdasarkan penelitian yang telah dibuat hasil akurasi nya sebagai berikut:

$$Accuracy = \frac{Jumlah\ Data\ yang\ Benar}{Jumlah\ Keseluruhan\ Data} \times 100\% = \frac{17+20+20+20}{80} \times 100\% = 96,25\%$$

Tabel 4.8 Nilai Recall, Precision

	Precision	Recall
Cacar Monyet	1	0,85
Cacar Air	1	1
HFMD	1	1
Normal	1	1
Average	0,96	0,96

Penelitian dengan judul Pemanfaatan Algoritma EfficientDet untuk deteksi Cacar Monyet dan Penyakit Kulit Serupa Berdasarkan citra lesi kulit berbasis Android berdasarkan tabel 4.7 mampu mendeteksi penyakit Cacar Monyet dan penyakit kulit serupa dengan rata-rata Precision sebesar 0,96, Recall 0,96 dan tingkat akurasi yaitu 96%.

BAB 5

KESIMPULAN DAN SARAN

5.1. Kesimpulan

Dari hasil uji yang telah dilakukan terhadap sistem deteksi penyakit cacar monyet dan penyakit kulit serupa yaitu Cacar air, Hand foot mouth disease, dan normal, dapat diperoleh kesimpulan yaitu sebagai berikut:

1. Pemanfaatan Algoritma EfficientDet terbukti bisa mendeteksi penyakit Cacar Monyet dan penyakit kulit serupa, yaitu Cacar Air, HFMD, dan Normal.
2. Sistem mampu mendeteksi penyakit Cacar Monyet dan penyakit kulit serupa dengan rata-rata Precision sebesar 0,96, Recall 0,96, dan tingkat akurasi yaitu 96%.
3. Percobaan dilakukan pada model efficientdet dengan melakukan tuning pada parameternya menghasilkan model terbaik yaitu pada efficientdet-d1 dengan batch size 4 dan step 30.000.

5.2. Saran

Dari hasil penelitian ini saran yang dapat dipertimbangkan untuk pengembangan sistem dan penelitian kedepannya yaitu:

1. Dalam proses training model, disarankan untuk melakukan hyperparameter tuning yang lebih bervariasi agar mendapatkan hasil maksimal.
2. Pada penelitian ini adalah hanya menggunakan 1 ukuran citra yaitu 720 x 720 piksel, sehingga ketika menggunakan ukuran citra yang berbeda ada kemungkinan tidak mampu mendeteksi objek
3. Memori GPU Tidak Cukup karena kekurangan sumber daya saat runtime di GPU karena model yang dilatih atau data yang diproses terlalu besar untuk memori GPU yang tersedia. Hal ini terjadi karena dataset yang sangat besar. Setiap batch dari data pelatihan bisa memakan banyak memori, sehingga yang bisa dilakukan hanya pada batch 4 saja dan jika batch size terlalu besar, itu bisa mengakibatkan kekurangan memori.

4. Penelitian selanjutnya diharapkan dapat memanfaatkan algoritma yang lain dalam mendeteksi penyakit cacar monyet dan penyakit kulit serupa untuk dibandingkan dengan penggunaan Efficientdet.

DAFTAR PUSTAKA

- Anggeriyane, E., Rahayu, S. F., Salamiah, D., Murizki, D. B., & Maulida, M. H. (2022). Edukasi Pentingnya Menjaga Diri dari Bahaya Cacar Air Melalui Media Pembelajaran Audiovisual. *Jurnal Pengabdian Masyarakat*, 01(03), 83–91.
- Bala, D., Hossain, M. S., Hossain, M. A., Abdullah, M. I., Rahman, M. M., Manavalan, B., Gu, N., Islam, M. S., & Huang, Z. (2023). MonkeyNet: A robust deep convolutional neural network for monkeypox disease detection and classification. *Neural Networks*, 161, 757–775. <https://doi.org/10.1016/j.neunet.2023.02.022>
- Belazarian, L. T., Lorenzo, M. E., Pearson, A. L., Sweeney, S. M., & Wiss, K. Exanthematous Viral Diseases. *Fitzpatrick's Dermatology in General Medicine*. 2011. Vol. 2, pp. 2373-2367. New York: McGraw Hill.
- Brownlee, J. (2021). A Gentle Introduction to Object Recognition With Deep Learning. *Deep Learning for Computer Vision*.
- Bunge, E. M., Hoet, B., Chen, L., Lienert, F., Weidenthaler, H., Baer, L. R., & Steffen, R. (2022). The Changing Epidemiology of Human Monkeypox—A Potential Threat? A systematic Review. *PLoS Neglected Tropical Diseases*, 16(2), 1–20. <https://doi.org/10.1371/journal.pntd.0010141>
- Cao, L., Zhang, X., Pu, J., Xu, S., Cai, X., & Li, Z. (2020). The Field Wheat Count Based on the Efficientdet Algorithm. *Proceedings of 2020 IEEE 3rd International Conference on Information Systems and Computer Aided Education, ICISCAE 2020*, 557–561. <https://doi.org/10.1109/ICISCAE51034.2020.9236918>
- Ferian, M., Akbari, R., Rahayudi, B., & Muflikhah, L. (2023). Implementasi Deep Learning menggunakan Algoritma EfficientDet untuk Sistem Deteksi Kelayakan Penerima Bantuan Langsung Tunai berdasarkan Citra Rumah di Wilayah Kabupaten Kediri. *Jurnal Pengembangan Teknologi Informasi Dan Ilmu Komputer*, 7(4), 1817–1825. <https://j-ptiik.ub.ac.id/index.php/j-ptiik/article/view/12596>








- Gilbourne, M. (2022). Mpox (Monkeypox). <https://dermnetnz.org/topics/mpox>, 10 Mei 2024 (diakses 10 Mei 2024).
- Hidayatullah, Priyanto. (2017). Pengolah *Citra Digital* Teori dan Aplikasi. Penerbit Informatika. Bandung.
- Kaler, J., Hussain, A., Flores, G., Kheiri, S., & Desrosiers, D. (2022). Monkeypox: A Comprehensive Review of Transmission, Pathogenesis, and Manifestation. *Cureus*, 14(7), 1–11. <https://doi.org/10.7759/cureus.26531>
- Lecun, Y., Bengio, Y., & Geoffrey, H. (2015). *Deep Learning*. <https://doi.org/10.1038/nature14539>.
- Liu, Q., Zhao, Y., Gao, R., Bu, X., & Hanajima, N. (2024). SpanEffiDet: Span-Scale and Span-Path Feature Fusion for Object Detection. *Neural Processing Letters*, 56(3), 1–32. <https://doi.org/10.1007/s11063-024-11653-6>
- Ludji, D. G., & Buan, F. C. H. (2023). Penerapan Metode Runge-Kutta Orde 4 pada Pemodelan Penularan Penyakit Cacar Monyet. *Journal of Mathematics Computations and Statistics*, 6(1), 1. <https://doi.org/10.35580/jmathcos.v6i1.37110>
- Marisah, Hilmi, I. L., & Salman. (2022). Studi Dan Tatalaksana Terkait Penyakit Cacar Monyet (Monkeypox) Yang Menginfeksi Manusia. *Jurnal Farmasetis*, 11(3), 201–208.
- Nainggolan, A. M. (2023). *Implementasi Algoritma EfficientDet untuk Klasifikasi Penyakit Alzheimer Berdasarkan Citra MRI Otak*. Skripsi. Universitas Sumatera Utara.
- Nayak, T., Chadaga, K., Sampathila, N., Mayrose, H., Gokulkrishnan, N., Bairy G, M., Prabhu, S., S, S. K., & Umakanth, S. (2023). Deep learning based detection of monkeypox virus using skin lesion images. *Medicine in Novel Technology and Devices*, 18(March), 100243. <https://doi.org/10.1016/j.medntd.2023.100243>
- Ng J., “Perbedaan Cacar Monyet dan Cacar Air,” 2021. <https://aido.id/health-articles/perbedaan-cacar-monyet-dan-cacar-air/detail> (accessed Dec. 13, 2022).








- Nursulistio, F., Kurniawardhani, A., & Fudholi, D. H. (2022). Deteksi Objek Masker Menggunakan EfficientDet-Lite3. *Automata*, 3(2).
<https://journal.uui.ac.id/AUTOMATA/article/view/24155>
- Pang, B., Nijkamp, E., & Wu, Y.N.(2020). Deep Learning With TenseorFlow: A Review. *Journal of Educational and Behavioral Statistic*,45(2),227-248.
- Petersen, E., Abubakar, I., Ihekweazu, C., Heymann, D., Ntoumi, F., Blumberg, L., Asogun, D., Mukonka, V., Lule, S. A., Bates, M., Honeyborne, I., Mfinanga, S., Mwaba, P., Dar, O., Vairo, F., Mukhtar, M., Kock, R., McHugh, T., Ippolito, G., & Zumla, A. (2019). Monkeypox — Enhancing Public Health Preparedness for an Emerging Lethal Human Zoonotic Epidemic Threat in The Wake of The Smallpox Post-Eradication Era. *International Journal of Infectious Diseases*, 78, 78–84.
<https://doi.org/10.1016/j.ijid.2018.11.008>
- Purmana, A., 2021. Implementasi Metode Deep Learning Dengan Menggunakan Algoritma Convolution Neural Network (CNN) Pada Citra Tulisan Tangan Aksara Sunda.
- Sahin, V. H., Oztel, I., & Yolcu Oztel, G. (2022). Human Monkeypox Classification from Skin Lesion Images with Deep Pre-trained Network using Mobile Application. *Journal of Medical Systems*, 46(11). <https://doi.org/10.1007/s10916-022-01863-7>
- Sinto, R., Sharifah, S., Epid, M., Ibrahim., dan Mendra, L. (2022). Penilaian risiko cepat Penyakit Cacar Monyet (*Monkeypox*). Kementerian Kesehatan Republik Indonesia.
- Susanto, L. A., Nilogiri, A., & Handayani, L. (2023). Klasifikasi Citra Lesi Kulit Serupa Virus Monkeypox Menggunakan VGG-19 Convolutional Neural Network. *JUSTINDO (Jurnal Sistem Dan Teknologi Informasi Indonesia)*, 8(1), 1–9.
<https://doi.org/10.32528/justindo.v8i1.168>
- Susanto, N. A. (2022). Asuhan Keperawatan Pada An. dengan Varicella (Chickenpox) atau Cacar Air. Universitas Muhammadiyah Malang.
- Tan, M., Pang, R., & Le, Q. V. (2020). EfficientDet: Scalable and efficient object detection. *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 10778–10787.








<https://doi.org/10.1109/CVPR42600.2020.01079>








- Tian, Y. (2020). Artificial Intelligence Image Recognition Method Based on Convolutional Neural Network Algorithm. *IEEE Access*, 8, 125731–125744.
- Victory, G. E. (2023). Klasifikasi *Monkeypox* dan *Non-monkeypox* Menggunakan Fitur Ekstraksi *LBP* dan Algoritma *Random Forest*. Skripsi. Universitas Multi Data Palembang.
- Wern, A (2018) Pengembangan Lebih mudah React Native dengan Expo | Envato Tuts+ (tutsplus.com) accessed 5 july 2024.
- Wicaksono, A., Purnomo, M. H., & Yuniarno, E. M. (2021). Deteksi Pejalan Kaki pada Zebra Cross untuk Peringatan Dini Pengendara Mobil Menggunakan Mask R-CNN. *Jurnal Teknik ITS*, 10(2). <https://doi.org/10.12962/j23373539.v10i2.80219>
- Wijaya, W., Pribadi, M. R., & Widiyanto, E. P. (2023). Klasifikasi Monkeypox Menggunakan Ekstraksi Fitur GLCM dan Algoritma Random Forest. *MDP Student Conference*, 2(1), 172–178. <https://doi.org/10.35957/mdp-sc.v2i1.4435>
- Wolf, K., Johnson, R.A., Suurmond, D. (2005). Fitzpatrick's Color Atlas and Synopsis of Clinical Dermatology. 5th ed. New York: McGraw-Hill.
- World Health Organisation. 2022. Multi-Country Monkeypox; World Health Organisation: Geneva, Switzerland.
- Wulandari, I., Yasin, H. & Widiharhi, T., 2020. Klasifikasi citra digital bumbu dan rempah dengan algoritma convolutional neural network (cnn). *Jurnal Gaussian*, 9(3), pp.273-282. Fakultas
- Zhao, Z. Q., Zheng, P., Xu, S. T., & Wu, X. (2019). Object Detection with Deep Learning: A Review. *IEEE Transactions on Neural Networks and Learning Systems*, 30(11), 3212–3232. <https://doi.org/10.1109/TNNLS.2018.2876865>








Lampiran






1.		Cacar Monyet	Cacar Monyet	Berhasil
2.		Cacar Monyet	Cacar Monyet	Berhasil
3.		Cacar Monyet	Cacar Monyet	Berhasil
4.		Cacar Monyet	Cacar Monyet	Berhasil
5.		Cacar Monyet	Cacar Air	Gagal
6.		Cacar Monyet	Cacar Air	Gagal
7.		Cacar Monyet	Cacar Monyet	Berhasil








8.		Cacar Monyet	Cacar Monyet	Berhasil
9.		Cacar Monyet	Cacar Air	Gagal
10.		Cacar Monyet	Cacar Monyet	Berhasil
11.		Cacar Monyet	Cacar Monyet	Berhasil
12.		Cacar Monyet	Cacar Monyet	Berhasil
13.		Cacar Monyet	Cacar Monyet	Berhasil
14.		Cacar Monyet	Cacar Monyet	Berhasil



15.		Cacar Monyet	Cacar Monyet	Berhasil
16.		Cacar Air	Cacar Air	Berhasil
17.		Cacar Air	Cacar Air	Berhasil
18.		Cacar Air	Cacar Air	Berhasil
19.		Cacar Air	Cacar Air	Berhasil
20.		Cacar Air	Cacar Air	Berhasil
21.		Cacar Air	Cacar Air	Berhasil





22.		Cacar Air	Cacar Air	Berhasil
23.		Cacar Air	Cacar Air	Berhasil
24.		Cacar Air	Cacar Air	Berhasil
25.		Cacar Air	Cacar Air	Berhasil
26.		Cacar Air	Cacar Air	Berhasil
27.		Cacar Air	Cacar Air	Berhasil
28.		Cacar Air	Cacar Air	Berhasil

29.		Cacar Air	Cacar Air	Berhasil
30.		Cacar Air	Cacar Air	Berhasil
31.		HFMD	HFMD	Berhasil
32.		HFMD	HFMD	Berhasil
33.		HFMD	HFMD	Berhasil
34.		HFMD	HFMD	Berhasil
35.		HFMD	HFMD	Berhasil

36.		HFMD	HFMD	Berhasil
37.		HFMD	HFMD	Berhasil
38.		HFMD	HFMD	Berhasil
39.		HFMD	HFMD	Berhasil
40.		HFMD	HFMD	Berhasil
41.		HFMD	HFMD	Berhasil
42.		HFMD	HFMD	Berhasil

43.		HFMD	HFMD	Berhasil
44.		HFMD	HFMD	Berhasil
45.		HFMD	HFMD	Berhasil
46.		Normal	Normal	Berhasil
47.		Normal	Normal	Berhasil
48.		Normal	Normal	Berhasil
49.		Normal	Normal	Berhasil

50.		Normal	Normal	Berhasil
51.		Normal	Normal	Berhasil
52.		Normal	Normal	Berhasil
53.		Normal	Normal	Berhasil
54.		Normal	Normal	Berhasil
55.		Normal	Normal	Berhasil
56.		Normal	Normal	Berhasil

57.		Normal	Normal	Berhasil
58.		Normal	Normal	Berhasil
59.		Normal	Normal	Berhasil
60.		Normal	Normal	Berhasil



**KEMENTERIAN PENDIDIKAN, KEBUDAYAAN,
RISET, DAN TEKNOLOGI**
UNIVERSITAS SUMATERA UTARA
FAKULTAS ILMU KOMPUTER DAN TEKNOLOGI INFORMASI

Jalan Universitas No. 9A Gedung A, Kampus USU Medan 20155, Telepon: (061) 821007
Laman: <http://Fasikomti.usu.ac.id>

KEPUTUSAN
DEKAN FAKULTAS ILMU KOMPUTER
DAN TEKNOLOGI INFORMASI
NOMOR : 2743/UN5.2.14.D/SK/SPB/2024
DEKAN FAKULTAS ILMU KOMPUTER
DAN TEKNOLOGI INFORMASI UNIVERSITAS SUMATERA UTARA

- Membaca** : Surat Permohonan Mahasiswa Fasikom-TI USU tanggal 11 Juli 2024 perihal permohonan ujian skripsi:
Nama : AIDA SURYANA RITONGA
NIM : 171402129
Program Studi : Sarjana (S-1) Teknologi Informasi
Judul Skripsi : Pemanfaatan Algoritma Efficient pada Deteksi Penyakit Cacar Monyet dan Penyakit Kulit Scrupa Berbasis Android
- Memperhatikan** : Bahwa Mahasiswa tersebut telah memenuhi kewajiban untuk ikut dalam pelaksanaan Meja Hijau Skripsi Mahasiswa pada Program Studi Sarjana (S-1) Teknologi Informasi Fakultas Ilmu Komputer dan Teknologi Informasi Universitas Sumatera Utara TA 2023/2024.
- Menimbang** : Bahwa permohonan tersebut diatas dapat disetujui dan perlu ditetapkan dengan surat keputusan
- Mengingat** : 1. Undang-undang Nomor 20 Tahun 2003 tentang Sistem Pendidikan Nasional.
2. Peraturan Pemerintah Nomor 17 tahun 2010 tentang pengelolaan dan penyelenggara pendidikan.
3. Keputusan Rektor USU Nomor 03/UN5.1.R/SK/SPB/2021 tentang Peraturan Akademik Program Sarjana Universitas Sumatera Utara.
4. Surat Keputusan Rektor USU Nomor 1876/UN5.1.R/SK/SDM/2021 tentang pengangkatan Dekan Fasikom-TI USU Periode 2021-2026
- MEMUTUSKAN**
- Menetapkan** :
Pertama : Membentuk dan mengangkat Tim Penguji Skripsi mahasiswa sebagai berikut:
Ketua : Dr. Marischa Elveny S.TI, M.Kom
NIP: 199003272017062001
Sekretaris : Umayu Ramadhani Putri Nasution S.TI, M.Kom.
NIP: 199104112021022001
Anggota Penguji : Ade Sarah Huzaifah S.Kom., M.Kom
NIP: 198506302018032001
Anggota Penguji : Fahrurrozi Lubis B.IT., M.Sc.IT
NIP: 198610122018052001
Moderator : -
Panitera : -
- Kedua** : Segala biaya yang diperlukan untuk pelaksanaan kegiatan ini dibebankan pada Dana Penerimaan Bukan Pajak (PNPB) Fasikom-TI USU Tahun 2024.
- Ketiga** : Keputusan ini berlaku sejak tanggal ditetapkan dengan ketentuan bahwa segala sesuatunya akan diperbaiki sebagaimana mestinya apabila dikemudian hari terdapat kekeliruan dalam surat keputusan ini.

Tembusan :

1. Ketua Program Studi Sarjana (S-1) Teknologi Informasi
2. Yang bersangkutan
3. Arsip

Medan
Ditandatangani secara elektronik oleh:
Dekan



Maya Silvi Lydia
NIP 197401272002122001

