TUGAS AKHIR – EC184801

DETEKSI *OUT OF DISTRIBUTION* (OOD) MENGGUNAKAN *EXTREME VALUE THEORY* (EVT) PADA KLASIFIKASI PENYAKIT KULIT

Yordan Yasin
NRP 07211640000013

Dosen Pembimbing
Dr. I Ketut Eddy Purnama, ST., MT.
Reza Fuad Rachmadi, S.T., M.T., Ph.D

DEPARTEMEN TEKNIK KOMPUTER
Fakultas Teknologi Elektro dan Informatika Cerdas
Institut Teknologi Sepuluh Nopember
Surabaya 2020

TUGAS AKHIR – EC184801

DETEKSI OUT OF DISTRIBUTION (OOD) MENGGUNAKAN EXTREME VALUE THEORY (EVT) PADA KLASIFIKASI PENYAKIT KULIT

Yordan Yasin
NRP 07211640000013

Dosen Pembimbing
Dr. I Ketut Eddy Purnama, ST., MT.
Reza Fuad Rachmadi, S.T., M.T., Ph.D

DEPARTEMEN TEKNIK KOMPUTER
Fakultas Teknologi Elektro dan Informatika Cerdas
Institut Teknologi Sepuluh Nopember
Surabaya 2020

[Halaman ini sengaja dikosongkan].

FINAL PROJECT – EC184801

OUT OF DISTRIBUTION (OOD) DETECTION USING EXTREME VALUE THEORY (EVT) IN SKIN DISEASE CLASSIFICATION

Yodan Yasin
NRP 07211640000013

Advisor
Dr. I Ketut Eddy Purnama, ST., MT.
Reza Fuad Rachmadi, S.T., M.T., Ph.D

Departement of Computer Engineering
Faculty of Intelligent Electrical and Informatics Technology
Sepuluh Nopember Institute of Technology
Surabaya 2020

[Halaman ini sengaja dikosongkan].

PERNYATAAN KEASLIAN TUGAS AKHIR

Dengan ini saya menyatakan bahwa isi sebagian maupun keseluruhan Tugas Akhir saya dengan judul "**Deteksi *Out of Distribution* (OOD) Menggunakan *Extreme Value Theory* (EVT) pada Klasifikasi Penyakit Kulit**" adalah benar-benar hasil karya intelektual mandiri, diselesaikan tanpa menggunakan bahan-bahan yang tidak diijinkan dan bukan karya pihak lain yang saya akui sebagai karya sendiri.

Semua referensi yang dikutip maupun dirujuk telah ditulis secara lengkap pada daftar pustaka.

Apabila ternyata pernyataan ini tidak benar, saya bersedia menerima sanksi sesuai peraturan yang berlaku.

Surabaya, 30 Juli 2020

Yordan Yasin

NRP. 07211640000013

LEMBAR PENGESAHAN

Deteksi Out of Distribution (OOD) Menggunakan Extreme Value Theory (EVT) pada Klasifikasi Penyakit Kulit

Tugas Akhir ini disusun untuk memenuhi salah satu syarat memperoleh gelar Sarjana Teknik di Institut Teknologi Sepuluh Nopember Surabaya

Oleh : Yordan Yasin (NRP: 07211640000013)

Tanggal Ujian : 8 Juli 2020 Periode Wisuda : September 2020

Disetujui oleh:

Dr. I Ketut Eddy Purnama, S.T.,M.T. (Pembimbing I)
NIP: 19690730 199512 1 001

Reza Fuad Rachmadi, S.T.,M.T.,Ph.D (Pembimbing II)
NIP: 19850403 201212 1 001

Mochamad Hariadi, S.T.,M.Sc.,Ph.D (Penguji I)
NIP: 19691209 199703 1 002

Susi Juniastuti, ST., M.Eng. (Penguji II)
NIP. 19650618 199903 2 001

Dr. Eko Mulyanto Yuniarno, ST., MT. (Penguji III)
NIP. 19680601 199512 1 009

Mengetahui
Kepala Departemen Teknik Komputer

Dr. Supeno Mardi Susiki Nugroho, ST., MT.
NIP. 197003131995121001

ABSTRAK

Nama Mahasiswa	:	Yordan Yasin
Judul Tugas Akhir	:	Deteksi *Out of Distribution* (OOD) Menggunakan *Extreme Value Theory* (EVT) pada Klasifikasi Penyakit Kulit
Pembimbing	:	1. Dr. I Ketut Eddy Purnama, ST., MT. 2. Reza Fuad Rachmadi, S.T., M.T., Ph.D

Sampai saat ini tingginya tingkat penderita penyakit kulit dan tidak meratanya tenaga medis masih menjadi masalah utama dalam bidang kesehatan di Indonesia. Dengan kemajuan Teknologi Informasi dan Komunikasi, diagnosis penyakit kulit dapat dilakukan sedini mungkin agar pasien segera mendapatkan perawatan. Salah satunya adalah dikembangkan sebuah sistem klasifikasi penyakit kulit yang akan diterapkan pada teknologi *Teledermatology*. Sistem ini akan mengklasifikasi penyakit kulit pada citra *dermoscopic* menggunakan algoritma *Deep Learning* yaitu *Convolutional Neural Network*. Teknologi ini sudah mampu mengklasifikasikan 7 penyakit kulit yang termasuk dalam kategori kanker kulit dengan akurasi sebesar 72%. Namun, 7 penyakit yang dapat diklasifikasikan tersebut bukanlah penyakit yang umum terjadi di Indonesia. Selain itu *Deep Learning* yang digunakan pada sistem klasifikasi tersebut saat ini masih bersifat closed set dan belum dapat menanangani data gambar yang bersifat open set. Oleh karena itu, perlu sistem klasifikasi yang mampu mengklasifikasikan penyakit yang umum terjadi di negara tropis dan mendeteksi Out of Distribution (OOD) agar sistem mampu menangani data gambar yang bersifat open set dengan menggunakan Extreme Value Theory (EVT).

Kata Kunci : Penyakit Kulit, *Deep Learning, Out of Distribution, Extreme Value Theory*

Halaman ini sengaja dikosongkan

ABSTRACT

Name : Yordan Yasin
Title : *Out of Distribution (OOD) Detection Using Extreme Value Theory (EVT) in Skin Disease Classification*
Advisors : 1. Dr. I Ketut Eddy Purnama, ST., MT.
2. Reza Fuad Rachmadi, S.T., M.T., Ph.D

Until now the high rate of sufferers of skin diseases and unequal medical personnel is still a major problem in health sector in Indonesia. With the advancement of Information and Communication Technology, the diagnosis of skin diseases can be made as early as possible so that patients immediately get treatment. False one of them was developed a skin disease classification system that will be applied to teledermatology technology. This system will classify skin diseases in dermoscopic images using the Deep Learning algorithm, Convolutional Neural Network. This technology is able to classify 7 diseases skin that is included in the category of skin cancer with accuracy by 72%. However, 7 diseases that can be classified are not a common disease in Indonesia. other than that Deep Learning used in the classification system is currently still closed set and cannot handle data images that are open set. Therefore, it needs a classification system that is able to classify common diseases that occur in tropical countries and detect Out of Distribution (OOD) so that the system is able to handle image data that is open set with using Extreme Value Theory (EVT).

Keywords : *Skin Disease, Deep Learning, Out of Distribution, Extreme Value Theory*

Halaman ini sengaja dikosongkan

KATA PENGANTAR

Puji dan syukur kehadirat Allah SWT atas segala limpahan berkah, rahmat, serta ridho-Nya, penulis dapat menyelesaikan penelitian ini dengan judul **Deteksi *Out of Distribution* (OOD) Menggunakan *Extreme Value Theory* (EVT) pada Klasifikasi Penyakit Kulit**.

Penelitian ini disusun dalam rangka pemenuhan bidang riset di Departemen Teknik Komputer ITS, serta digunakan sebagai persyaratan menyelesaikan pendidikan Sarjana. Penelitian ini dapat terselesaikan tidak lepas dari bantuan berbagai pihak. Oleh karena itu, penulis mengucapkan terima kasih kepada :

1. Semua keluarga, kedua orang tua dan kedua kakak saya yang telah memberikan dorongan spiritual dan material dalam penyelesaian buku penelitian ini.
2. Bapak Kepala Departemen Teknik Komputer ITS Dr. Supeno Mardi Susiki Nugroho, S.T.,M.T..
3. Bapak Dr. I Ketut Eddy Purnama, S.T.,M.T., dan Bapak Reza Fuad Rachmadi, S.T., M.T., Ph.D yang senantiasa memberikan motivasi, bimbingan, dan arahan selama mengerjakan penelitian.
4. Bapak-ibu dosen pengajar Departemen Teknik Komputer ITS, atas pengajaran, bimbingan, serta perhatian yang diberikan kepada penulis selama ini.
5. Seluruh teman-teman Teknik Komputer 2016 yang selalu membantu dalam penelitian ini.
6. Teman-teman anggota Lab B401 yang sudah banyak membantu penulis selama proses penelitian dan pembuatan buku.
7. Teman-teman angkatan e56 yang selalu memberikan dukungan kepada penulis.

Surabaya, Juni 2020

Penulis

Halaman ini sengaja dikosongkan

DAFTAR ISI

ABSTRAK **i**

ABSTRACT **iii**

KATA PENGANTAR **v**

DAFTAR ISI **vii**

DAFTAR GAMBAR **xi**

DAFTAR TABEL **xiii**

NOMENKLATUR **xv**

1 PENDAHULUAN **1**
1.1 Latar belakang 1
1.2 Permasalahan 2
1.3 Tujuan 2
1.4 Batasan masalah 3
1.5 Sistematika Penulisan 3

2 TINJAUAN PUSTAKA **5**
2.1 *Artificial Intelligence* 5
2.2 *Machine Learning* 6
2.3 *Artificial Neural Network* 7
2.4 *Deep Learning* 9
2.5 *Convolutional Neural Network* 10
2.5.1 *Convolutional Layer* 11
2.5.2 *Pooling Layer* 12
2.5.3 *Rectified Linear Unit (ReLU) Activation Function* 13
2.5.4 *Softmax* 14
2.6 *Transfer Learning* 14
2.6.1 *Inception-v3* 15
2.6.2 *MobileNetV1* 15
2.7 *Confusion Matrix* 16

2.8 *TensorFlow* 17
2.8.1 *Keras* 18
2.9 *Extreme Value Theory* 19
2.9.1 *Weibull Distribution* 19
2.10 *Closed-set Classification* dan *Open-set Recognition* . 20

3 DESAIN DAN IMPLEMENTASI SISTEM 21
3.1 Desain Sistem 21
3.2 Datasets DERMNET 23
3.2.1 *Class Dataset* 24
3.3 Pemisahan dan Augmentasi Data 27
3.4 Pra-pemrosesan Data 30
3.5 *Training Data* 30
3.5.1 Load Pretrained Model 31
3.6 Validasi Data 31
3.7 Ekstrasi nilai *Activation Vector* (AV) pada *Penultimate Layer* 32
3.8 *Fitting* distribusi *Weibull* 34
3.9 Penentuan Tipe Penyakit Kulit 35

4 PENGUJIAN DAN ANALISA 37
4.1 Pengujian Jenis Model 37
4.1.1 Pengujian Model *MobileNet* v1 37
4.1.2 Pengujian Model *Inception* V3 39
4.2 Pengujian Hasil Klasifikasi 40
4.2.1 Hasil Klasifikasi *MobileNet* V1 41
4.2.2 Hasil Prediksi *Inception* V3 48
4.3 Analisa Perbandingan Model 55
4.4 Pengujian Hasil Implementasi *Extreme Value Theory* Pada Proses Klasifikasi (*Open-set Recognition*) . . . 56
4.5 Analisa Perbandingan Hasil Pengujian Implementasi *Extreme Value Theory* (*Open-set Recognition*) 60

5 PENUTUP 63
5.1 Kesimpulan 63
5.2 Saran 63

DAFTAR PUSTAKA 65

Biografi Penulis **71**

Halaman ini sengaja dikosongkan

DAFTAR GAMBAR

2.1 Ilustrasi Artificial Intelligence [1] 5
2.2 Ilustrasi Machine Learning [2] 7
2.3 Operasi *Perceptron* pada *Neural Network* 8
2.4 Perbandingan dari *Artificial Intelligence, Machine Learning ,Artificial Neural Network* dan *Deep Learning* [3] 10
2.5 *Convolutional Neural Network* Arsitektur [4] 11
2.6 Convolution Layer [5] 12
2.7 *Max Pooling dengan 2 x 2 filter dan stride = 2.* [5] . 13
2.8 Fungsi Aktivasi ReLU [6] 14
2.9 Arsitektur dari *Inception-v3* [7] 15
2.10 Arsitektur dari *MobileNet V1* [8] 15
2.11 *confusion matrix* [9] 16
2.12 *Workflow* pada *TensorFlow.* [10] 18
2.13 Ilustrasi nilai ekstrim. [11] 19
2.14 Ilustrasi Closed-set Classification dan Open-Set Recognition [12] 20

3.1 Bagan Umum Metodologi Sistem 21
3.2 Dataset DERMNET [13] 24
3.3 Visualisasi dari Pemisahan Data [14] 27
3.4 ilustrasi *Cross Validation* [15] 32
3.5 Ilustrasi Ekstrasi nilai Aactivation Vector 33
3.6 Ilustrasi Image Classification 35

4.1 Grafik nilai *loss* training Mobilenet V1 37
4.2 Grafik nilai *accuracy* training Mobilenet V1 38
4.3 Grafik nilai *loss* training Inception V3 39
4.4 Grafik nilai *accuracy* training Inception V3 40
4.5 Grafik nilai klasifikasi *acne* MobileNet V1 41
4.6 Grafik nilai klasifikasi *disez* MobileNet V1 42
4.7 Grafik nilai klasifikasi *dmta* MobileNet V1 43
4.8 Grafik nilai klasifikasi *frncl* MobileNet V1 44
4.9 Grafik nilai klasifikasi *herp* MobileNet V1 45
4.10 Grafik nilai klasifikasi *psr* MobileNet V1 46

4.11 Grafik nilai klasifikasi *ring* MobileNet V1 47
4.12 Grafik nilai klasifikasi *acne* Inception V3 48
4.13 Grafik nilai klasifikasi *disez* Inception V3 49
4.14 Grafik nilai klasifikasi *dmta* Inception V3 50
4.15 Grafik nilai klasifikasi *frncl* Inception V3 51
4.16 Grafik nilai klasifikasi *herp* Inception V3 52
4.17 Grafik nilai klasifikasi *psr* Inception V3 53
4.18 Grafik nilai klasifikasi *ring* Inception V3 54
4.19 Grafik rata-rata performa klasifikasi *MobileNet* v1(%) 55
4.20 Grafik rata-rata performa klasifikasi *Inception* v3(%) 56
4.21 Hasil pengujian proses klasifikasi dataset DERMNET dengan input gambar 7 kelas penyakit kulit 57
4.22 Hasil pengujian proses klasifikasi dataset STL-10 dengan input gambar *open-set* 58
4.23 Hasil pengujian proses klasifikasi dataset HAM10000 dengan input gambar *open-set* penyakit kulit 59
4.24 Grafik performa *Open-set Recognition* pada dataset DERMNET 60
4.25 Grafik performa *Open-set Recognition* dengan dataset STL-10 61
4.26 Grafik performa *Open-set Recognition* dengan dataset HAM10000 61

DAFTAR TABEL

3.1 *Beberapa contoh metadata dataset DERMNET* . . . 28
3.2 *Beberapa contoh metadata dataset DERMNET setelah proses pemisahan data* 28
3.3 *Tabel hasil pembagian data* 29
3.4 *Tabel hasil penambahan data* 29

4.1 Tabel nilai *confusion matrix* Mobilenet V1 38
4.2 Tabel nilai *confusion matrix* Inception V3 40

Halaman ini sengaja dikosongkan

NOMENKLATUR

$V(X)$	:	Nilai aktivasi vector pada penultimate layer
j	:	kelas yang diklasifikasikan
$S_{i,j}$	:	kelas yang terklasifikasi benar
μ	:	Nilai rata-rata aktivasi vektor yang terklasifikasi benar
η	:	nilai jarak terbesar
ρ	:	fungsi distribusi *Weibull*
τ_i	:	Fungsi pergeseran data pada *Weibull Fitting*
κ_i	:	*Shape parameter* pada distribusi *Weibull*
λ_i	:	*Scale parameter* pada distribusi *Weibull*

Halaman ini sengaja dikosongkan

BAB 1
PENDAHULUAN

Penelitian ini di latar belakangi oleh berbagai kondisi yang menjadi acuan. Selain itu juga terdapat beberapa permasalahan yang akan dijawab sebagai luaran dari penelitian.

1.1 Latar belakang

Penyakit kulit merupakan salah satu penyakit yang sering menjangkit penduduk Indonesia. Hal ini dibuktikan dari profil kesehatan Indonesia tahun 2009 yang menunjukkan bahwa penyakit kulit dan jaringan subkutan menjadi peringkat ketiga dari 10 penyakit terbanyak pada pasien rawat jalan di rumah sakit se-Indonesia[16]. Untuk memberikan perawatan kepada pasien di seluruh daerah, pusat-pusat pelayanan kesehatan didirikan di seluruh wilayah di Indonesia, namun hal ini tidak diimbangi dengan pemerataan tenaga dan ahli medis yang cenderung menumpuk di perkotaan besar[17].

Dengan kemajuan teknologi informasi dan komunikasi, pemerataan layanan kesehatan bisa dilakukan melalui sistem *telemedicine*. Salah satu layanan *telemedicine* yang paling mungkin dilakukan adalah layanan penyakit kulit atau *teledermatology*. Layanan ini mengacu pada pemanfaatan teknologi telekomunikasi untuk diagnosis medis penyakit kulit, pelayanan kesehatan, dan perawatan pasien. Teledermatology memiliki potensi besar untuk merevolusi pengiriman data medis guna memenuhi layanan pengobatan penyakit kulit (dermatologi), memberikan layanan yang merata ke daerah terpencil serta memungkinkan dokter spesialis kulit untuk memberikan rujukan kepada pasien melakukan pemeriksaan lanjutan ke pusat dermatologi unggulan[18]. Teledermatology yang sedang dikembangkan saat ini adalah sistem klasifikasi penyakit kulit menggunakan *Convolutional Neural Network* (CNN) pada citra *dermoscopic*.

Sistem klasifikasi penyakit kulit menggunakan *Convolutional Neural Network* (CNN) pada citra *dermoscopic* menggunakan dataset publik MNIST HAM1000 yang terdiri dari 10.015 *dermoscopic*. Sistem tersebut dapat mengklasifikasikan tujuh kelas penya-

kit kulit, yaitu *Melanocytic Nevi, Melanoma, Benign keratosis-like lesions, Basal cell carcinoma, Actinic keratoses, Vascular lesions, dan Dermatofibroma*[19]. Namun Tujuh kelas penyakit kulit yang diklasifikasikan tersebut bukan merupakan penyakit kulit yang sering terjadi pada negara tropis seperti *Dermatitis Atopik, Discoid Eczema, Psoriasis, Herpes, ringworm, dan furunkel.*

Dalam pengaplikasiannya, sistem klasifikasi penyakit kulit yang ada, sudah dapat mengklasifikasikan penyakit kulit dalam citra demoscropic menggunakan model CNN *Inception V3* dengan nilai akurasi sebesar 72%. Namun sistem tersebut bersifat *closed set* atau memaksa sistem untuk memilih dari salah satunya kelas yang diketahui, walapun gambar yang akan ditest tidak termasuk dalam kelas yang diklasifikasikan. Hal ini menyebabkan adanya *Out of Distribution* (OOD) atau adanya data anomali yang menyimpang dari pendistribusian data pada proses pengklasifikasian data. OOD dapat dideteksi dengan menggunakan *Extreme Value Theory* (EVT) yang nantinya sistem dapat menolak data yang tidak termasuk dalam kelas yang diklasifikasikan..

1.2 Permasalahan

Berdasarkan data yang telah dipaparkan di latar belakang, dapat dirumuskan beberapa rumusan masalah, sebagai berikut:

1. Pada penelitian sebelumnya milik Arta Kusuma Hernanda baru dapat mengklasifikasikan tujuh kelas penyakit, dan belum dapat mengklasifikasi penyakit kulit lain yang sering terjadi pada negara tropis khususnya Indonesia.

2. Pada penelitian sebelumnya belum dapat menolak gambar *Open-Set* yang tidak berkaitan dengan pendistribusian data training.

1.3 Tujuan

Tujuan dari penelitian ini adalah mengembangkan sistem deteksi *Out of Distribution* menggunakan *Extreme Value Theory* pada klasifikasi penyakit kulit untuk dapat menolak gambar *Open-Set* yang tidak berkaitan dengan pendistribusian data training dan mengklasifikasikan penyakit kulit yang sering terjadi pada negara

tropis khususnya Indonesia untuk memperbaiki sistem klasifikasi penyakit kulit yang ada saat ini dengan.

1.4 Batasan masalah

Untuk memfokuskan permasalahan yang akan diangkat maka dilakukan pembatasan masalah. Batasan-batasan masalah tersebut di antaranya adalah :

1. Menggunakan dataset publik terkait penyakit kulit yang sering terjadi pada negara tropis.
2. Klasifikasi pada sistem ini terbatas pada depalan kelas penyakit kulit, yaitu Acne (acne), Dermatitis Atopik (dmta), Discoird Eczema (disez), Furunkle (frncl), Herpes (herp), Psoriasis (psr), Ringworm (ring) (Closed Dataset).
3. Menggunakan dataset publik terkait penyakit kulit yang sering terjadi pada negara tropis.

1.5 Sistematika Penulisan

Laporan penelitian tugas akhir ini tersusun dalam sistematika dan terstruktur sehingga mudah dipahami dan dipelajari oleh pembaca maupun seseorang yang ingin melanjutkan penelitian ini. Alur sistematika penulisan laporan penelitian ini yaitu :

1. BAB I Pendahuluan
   Bab ini berisi uraian tentang latar belakang permasalahan, penegasan dan alasan pemilihan judul, sistematika laporan, tujuan dan metodologi penelitian.
2. BAB II Dasar Teori
   Pada bab ini berisi tentang uraian secara sistematis teori-teori yang berhubungan dengan permasalahan yang dibahas pada penelitian ini. Teori-teori ini digunakan sebagai dasar dalam penelitian, yaitu algoritma *Deep Convolutional Neural Network*, *TensorFlow*, *Extreme Value Theory* dan teori-teori penunjang lainya.
3. BAB III Perancangan Sistem dan Impementasi
   Bab ini berisi tentang penjelasan-penjelasan terkait eksperimen yang akan dilakukan dan langkah-langkah pengolahan dataset, proses training model CNN dan penerapan *Extreme Value Theory* untuk mendeteksi *Out of Distribution.* Guna men-

dukung hal tersebut, digunakanlah blok diagram atau *work flow* agar sistem yang akan dibuat dapat terlihat dan mudah dibaca untuk implentasi pada pelaksanaan tugas akhir.

4. BAB IV Pengujian dan Analisa
   Bab ini menjelaskan tentang pengujian model CNN hasil training menggunakan cross validation, serta analisis hasil pengujian penerapan *Extreme Value Theory* untuk mendeteksi *Out of Distribution*.

5. BAB V Penutup
   Bab ini merupakan penutup yang berisi kesimpulan yang diambil dari penelitian dan pengujian yang telah dilakukan. Saran dan kritik yang membangun untuk pengembangkan lebih lanjut juga dituliskan pada bab ini.

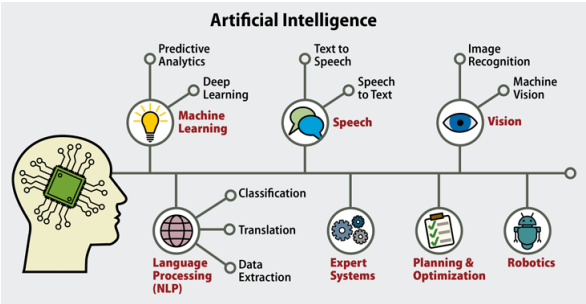
BAB 2
TINJAUAN PUSTAKA

Demi mendukung penelitian ini, dibutuhkan beberapa teori penunjang sebagai bahan acuan dan refrensi. Dengan demikian penelitian ini menjadi lebih terarah.

2.1 *Artificial Intelligence*

Artificial Intelligence (AI) adalah bidang studi yang menggambarkan kemampuan mesin dalam mempelajari sesuatu seperti layaknya manusia dan kemampuan untuk menanggapi perilaku tertentu. AI memiliki keunggulan dalam efektivitas biaya, memecahkan masalah yang rumit, dan menentukan keputusan. Dengan banyaknya keunggulan tersebut, AI diterapkan pada sebagian besar bidang bisnis dan teknik[20].

AI bekerja dengan cara menggabungkan sejumlah besar data dengan pemrosesan yang cepat, berulang dan algoritma cerdas, memungkinkan perangkat lunak untuk belajar secara otomatis dari pola atau fitur dalam data. AI adalah bidang studi luas yang mencakup banyak teori, metode, dan teknologi, serta subbidang utama yaitu *Machine Learning, Neural Network, Cognitive Computer, Computer Vision,* dan *Natural Languange Processing.* Ilustrasi dari *Artificial Intelligence* ditunjukkan pada gambar 2.1.



Gambar 2.1: Ilustrasi Artificial Intelligence [1]

2.2 *Machine Learning*

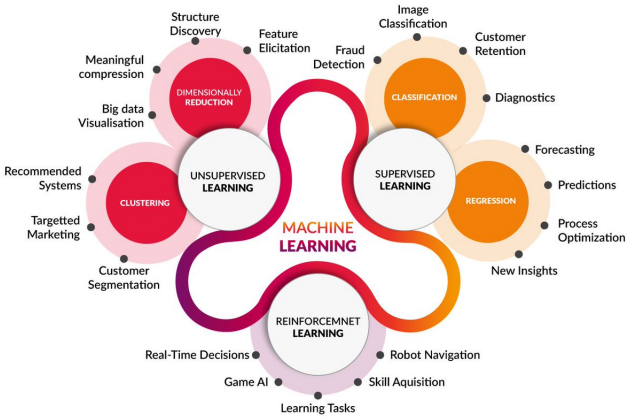
Machine Learning (ML) merupakan sebuah cabang dari *Artificial Intelligence* (kecerdasan buatan) yang memungkinkan sistem komputer untuk belajar langsung dari contoh, data, dan pengalaman yang didapatkannya. Dengan membuat komputer mampu menjalankan tugas yang spesifik secara cerdas, ML dapat melakukan algoritma yang kompleks dengan cara mempelajari data yang diberikan tanpa perlu mengikuti program yang dibuat atau diinstruksikan secara manual oleh pengguna[21].

Ciri khas dari ML adalah adanya proses pembelajaran atau *training.* Oleh karena itu, ML membutuhkan data untuk dipelajari yang disebut sebagai data *training.* Setelah berhasil melakukan *training,* maka ML dapat melakukan proses klasifikasi dan prediksi terhadap data baru yang diberikan sesuai dengan hasil *training* yang telah dilakukan. Klasifikasi adalah metode dalam ML yang digunakan oleh mesin untuk memilah atau mengklasifikasikan objek berdasarkan ciri tertentu sebagaimana manusia mencoba membedakan benda satu dengan yang lain. Sedangkan prediksi atau regresi digunakan oleh mesin untuk menerka keluaran dari suatu data masukan[22]. Beberapa contoh implementasi dari ML antara lain adalah *image processing, search recommendation, text and speech recognition,* dan *health diagnosis.*

Di dalam ML, terdapat tiga kategori pembelajaran[23], yaitu:

1. *Supervised Learning*
   Supervised learning merupakan metode pembelajaran dimana data yang akan dipelajari sudah diberikan label yang menunjukkan klasifikasi atau kelas data tersebut berada. Model yang dihasilkan adalah model prediksi dari data yang telah diberi label.
2. *Unsupervised Learning*
   Unsupervised learning merupakan metode pembelajaran dimana data yang akan dipelajari tidak memiliki label, sehingga komputer perlu melakukan analisis dan klasifikasi berdasarkan data yang ada.
3. *Reinforcement Learning*
   Reinforcement learning merupakan metode pembelajaran di-

mana diberlakukan *reward* pada hasil pembelajaran yang positif dan *punishment* pada hasil pembelajaran yang negatif. Komputer tidak diberitahu apa yang harus dilakukan, namun memiliki tujuan untuk mendapatkan *reward* yang maksimal sehingga memaksa komputer untuk belajar secara terus-menerus.

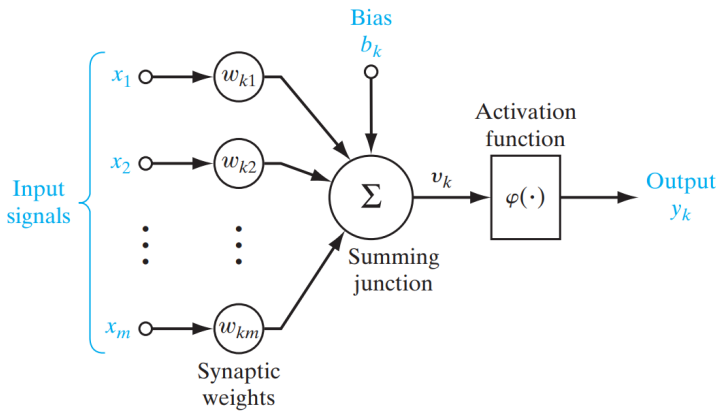


Gambar 2.2: Ilustrasi Machine Learning [2]

2.3 *Artificial Neural Network*

Artificial Neural Network (ANN) adalah teknik dalam ML yang bertujuan untuk menirukan saraf manusia yang merupakan bagian fundamental dari otak. ANN terdiri atas lapisan masukan *(input layer)* dan lapisan keluaran *(output layer)*. Setiap *layer* terdiri atas satu atau beberapa unit neuron yang mempunyai sebuah fungsi aktivasi yang menentukan *output* dari unit tersebut. Kita bisa menambahkan lapisan tersembunyi *(hidden layer)* untuk menambah kemampuan dari ANN tersebut. Neuron adalah unit pengolahan informasi dasar bagi operasi ANN[22]. Pada tahun 1958, Frank Rosenblatt mengenalkan jaringan saraf tiruan ini sebagai *perceptron*. Gambar 2.3 memperlihatkan detail dari operasi *perceptron*.

Pada Gambar 2.3, terdapat tiga elemen dasar dari neuron atau *perceptron*[24] yaitu:



Gambar 2.3: Operasi *Perceptron* pada *Neural Network*. [24]

1. *Synapses* atau penghubung yang dikategorikan berdasarkan bobot atau kekuatannya.
2. *Adder* yang berfungsi untuk menjumlahkan sinyal *input* yang telah diberikan bobot oleh *synapses*.
3. *Activation function* yang berfungsi untuk membatasi amplitudo *output* dari sebuah neuron. Biasanya, batas amplitudo normal dari *output* ditulis dalam bentuk interval satuan tertutup [0, 1] atau [-1, 1].

Secara matematis, neuron k yang terdapat di Gambar 2.3 dapat dijelaskan dengan persamaan 2.3.1 dan persamaan 2.3.2.

$$u_k = \sum_{j=1}^{m} w_{kj} \cdot x_j \tag{2.3.1}$$

dan

$$y_k = \varphi(u_k + b_k) \tag{2.3.2}$$

dimana:

$x_1, x_2, ..., x_m$ = sinyal *input*
$w_{k1}, w_{k2}, ..., w_{km}$ = bobot *synapses* dari neuron k
u_k = kombinasi linier *output* yang disebabkan oleh sinyal *input*

y_k = sinyal *output* dari neuron
b_k = bias
$\varphi(\cdot)$ = fungsi aktivasi

Penggunaan bias b_k bertujuan untuk menambahkan sebuah input *dummy* yang bernilai satu yang mana perhitungannya direpresentasikan dengan persamaan 2.3.3.

$$v_k = u_k + b_k \tag{2.3.3}$$

Bias b_k memberikan efek *affine transformation* kepada *output* u_k sehingga dihasilkan *activation potential* v_k. Dari kombinasi persamaan 2.3.1 dan 2.3.2 maka didapatkan persamaan 2.3.4 dan persamaan 2.3.5.

$$v_k = \sum_{j=0}^{m} w_{kj} \cdot x_j \tag{2.3.4}$$

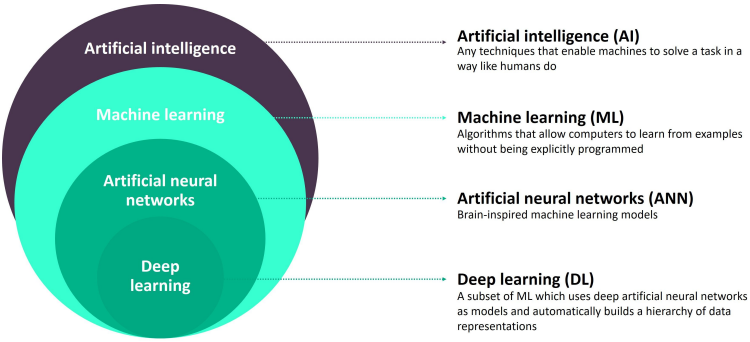
$$y_k = \varphi(v_k) \tag{2.3.5}$$

NN bisa dilatih dengan menggunakan data *training.* Semakin banyak data *training* maka akan semakin bagus performa dari NN tersebut. Namun, kemampuan NN juga terbatas pada jumlah *layer,* semakin banyak jumlah *layer* semakin tinggi kapasitas NN tersebut. Semakin banyak *layer* juga membawa kekurangan yaitu semakin banyaknya jumlah iterasi atau *training* yang dibutuhkan. Untuk mengatasi hal ini, dikembangkanlah teknik *deep learning*[22].

2.4 *Deep Learning*

Deep Learning (DL) adalah salah satu bidang yang muncul dari penelitian *Machine Learning* (ML). *Deep Learning* merupakan sebuah metode implementasi *machine learning* yang bertujuan meniru cara kerja otak manusia menggunakan jaringan saraf buatan. *Deep Learning* memungkinkan model komputasi yang terdiri dari beberapa lapisan pemrosesan untuk mempelajari representasi data dengan berbagai tingkatan abstraksi. *Deep Learning* menemukan struktur rumit dalam kumpulan data besar dengan menggunakan algoritma *backpropagation* untuk menunjukkan bagaimana suatu mesin harus mengubah parameter internal yang digunakan untuk menghitung representasi di setiap lapisan dari representasi

di lapisan sebelumnya[25]. *Deep Learning* melakukan pendekatan dalam penyelesaian masalah dengan menggunakan konsep hierarki. Dengan konsep tersebut, komputer mampu mempelajari sebuah konsep yang kompleks dengan menggabungkan konsep-konsep yang lebih sederhana. Perbandingan dari *Artificial Intelligence, Machine Learning ,Artificial Neural Network* dan *Deep Learning* dapat dilihat pada gambar 2.4.



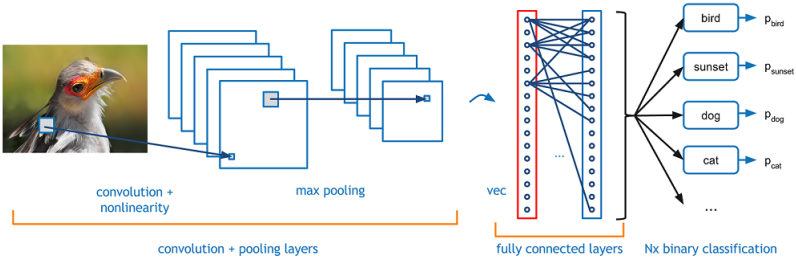
Gambar 2.4: Perbandingan dari *Artificial Intelligence, Machine Learning ,Artificial Neural Network* dan *Deep Learning* [3]

2.5 *Convolutional Neural Network*

Convolutional Neural Network (CNN) adalah variasi dari *Multilayer Perceptron* yang terinspirasi dari jaringan saraf manusia. Pemberian nama *convolutional neural network* mengindikasikan bahwa jaringan tersebut menggunakan operasi matematika yang disebut konvolusi. Konvolusi sendiri adalah sebuah operasi linier. CNN merupakan pengembangan dari *artificial neural network* yang saat ini diklaim sebagai model terbaik untuk memecahkan masalah seputar *object recognition* dan *detection.*

Secara teknis, CNN adalah arsitektur yang bisa di *training* dan terdiri dari beberapa tahap. *Input* dan *output* dari masing-masing tahap berupa *array* yang disebut *feature map* atau peta fitur. *Output* dari masing-masing tahap adalah *feature map* hasil pengolahan dari semua lokasi pada *input.* Struktur CNN dibangun dari tiga jenis *layer* utama yaitu *convolution layer, pooling layer,* dan *activation function*[26]. Ilustrasi dari CNN ditunjukan pada

gambar 2.5.



Gambar 2.5: *Convolutional Neural Network* Arsitektur [4]

2.5.1 *Convolutional Layer*

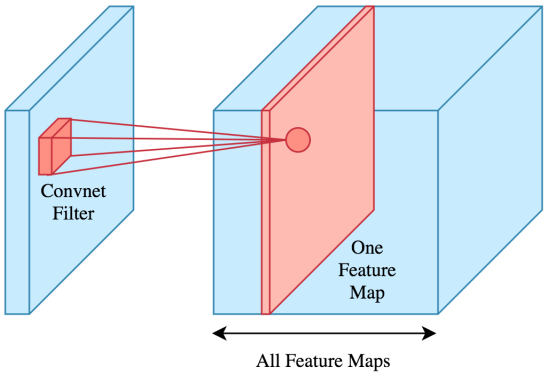
Convolutional layer adalah blok bangunan inti dari *Convolutional Neural Network*. Ilustrasi *Convolution Layer* pada *neural network* dapat dilihat pada gambar 2.6. Parameter dari *layer* terdiri dari satu set filter yang dapat dipelajari (atau *kernel*), yang memiliki bidang reseptif kecil, tetapi meperluas melalui kedalaman penuh dari *input volume*. Selama proses *forward pass*, setiap filter dilakukan proses konvolusi terhadap lebar dan tinggi dari *input volume*, menghitung *dot product* antara entri filter dan *input* dan menghasilkan *activation maps* berbentuk 2-dimensi dari filter tersebut. Sebagai hasilnya, jaringan ini mempelajari filter yang aktif ketika mendeteksi beberap jenis fitur tertentu pada beberapa posisi spasial dalam *input*. Secara umum operasi konvolusi dapat dituliskan dengan persamaan 2.5.1.1.

$$s(t) = (x \times w)(t) \tag{2.5.1.1}$$

Pada fungsi $s(t)$ menghasilkan *output* tunggal yaitu *feature map*. Argumen pertama berupa *input* yang merupakan x dan argumen kedua yang merupakan w sebagai *kernel* atau filter. Jika melihat *input* sebagai citra dua dimensi, maka (t) bisa diasumsikan sebagai sebuah piksel dan menggantinya dengan i dan j. Oleh karena itu, untuk operasi konvolusi dengan lebih dari satu dimensi dapat digunakan persamaan 2.5.1.2.

$$S_{(i,j)} = (K \times I)_{(i,j)} = \Sigma\Sigma I_{(i-m,j-n)} K_{(m,n)} \tag{2.5.1.2}$$

Persamaan 2.5.1.2 merupakan perhitungan dalam operasi konvolusi dengan i dan j sebagai piksel dari sebuah citra. Perhitungannya bersifat komutatif dan muncul ketika $K_{(m,n)}$ sebagai *kernel*-nya serta I sebagai *input* dan *kernel* yang dapat dibalik relatif terhadap *input.* Operasi konvolusi dapat dilihat sebagai perkalian matriks antara citra *input* dan filter dimana *output*-nya dapat dihitung dengan *dot product.*

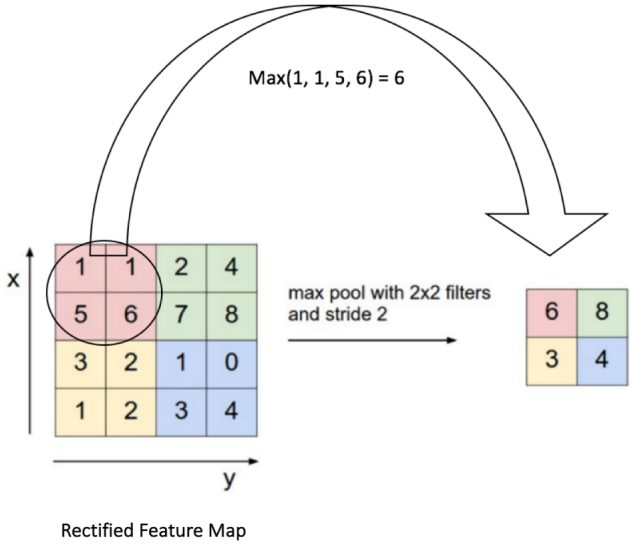


Gambar 2.6: Convolution Layer [5]

2.5.2 *Pooling Layer*

Konsep penting lain dari *Convolutional Neural Network* adalah *pooling*, yang merupakan bentuk non-linear dari *down-sampling.* Ada beberapa fungsi non-linear untuk menerapkan *pooling* di mana *max-pooling* adalah yang paling umum. *Pooling* membagi gambar *input* menjadi satu set persegi panjang dan untuk setiap sub-wilayah tersebut, nilai dari *output* akan maksimum.

Pooling Layer beroperasi secara independen pada setiap irisan kedalaman *input* dan mengubah ukurannya secara spasial. Bentuk yang paling umum adalah *pooling layer* dengan filter ukuran 2 x 2 diterapkan dengan *stride* 2 sampel bawah pada setiap irisan kedalaman *input* dengan 2 pada setiap lebar dan tinggi, sehingga membuang 75% dari aktivasi. Dalam hal ini, setiap operasi maksimal lebih dari 4 angka. Ilustrasi dari *Max Pooling* dengan 2 x 2 filter dan *stride* = 2 dapat dilihat pada gambar 2.7.

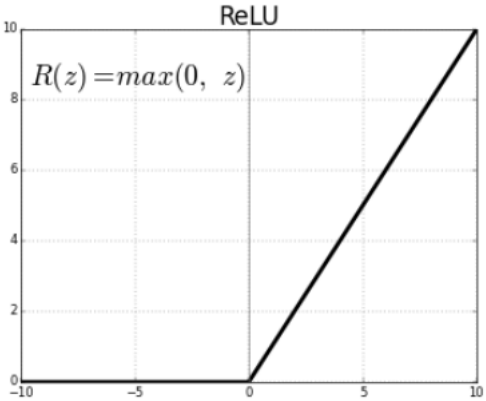


Gambar 2.7: *Max Pooling dengan 2 x 2 filter dan stride = 2.* [5]

2.5.3 *Rectified Linear Unit (ReLU) Activation Function*

Pada dasarnya fungsi ReLU melakukan treshold dari 0 hingga tidak terhingga (*infinity*). Fungsi ini menjadi salah satu fungsi aktivasi yang populer saat ini. Pada fungsi ini input dari *neuron-neuron* berupa bilangan negatif akan diterjemahkan kedalam nilai 0, dan jika masukan bernilai positif maka *output* dari *neuron* adalah nilai aktivasi itu sendiri. Fungsi aktivasi ini memiliki kelebihan yaitu dapat mempercepat proses konfigurasi yang dilakukan dengan *Stochastic Gradient Descent* 7 (SGD). Kelemahan ReLU yaitu fungsi aktivasi ini bisa menjadi rapuh pada proses *training* dan bisa membuat unit tersbut mati. Grafik fungsi aktivasi ReLU ditunjukan pada gambar 2.8 [27]. Rumus dari fungsi aktivasi *ReLU* ditunjukkan pada persamaan 2.5.3.1

$$R(z) = max(0, z) \tag{2.5.3.1}$$



Gambar 2.8: Fungsi Aktivasi ReLU [6]

2.5.4 *Softmax*

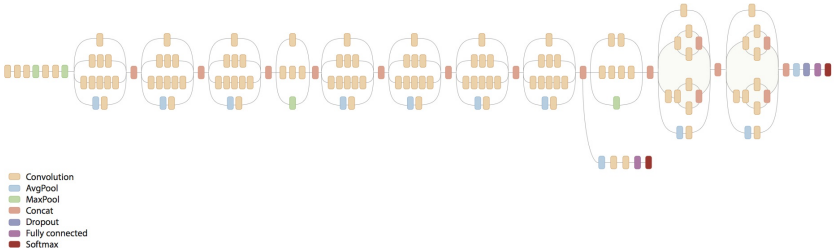
Dalam matematika, fungsi aktivasi *softmax* juga dikenal sebagai softargmax atau fungsi yang mendapatkan nilai index dari nilai maksimum (max) dalam sebuah tensor. Softmax adalah fungsi aktivasi yang mengambil *input* vektor dari bilangan real K , dan menormalkannya menjadi distribusi probabilitas yang terdiri dari probabilitas K. *Softmax* akan merubah beberapa komponen vektor negatif atau lebih besar dari satu dan mungkin tidak berjumlah 1 menjadi interval (0,1) dan komponen akan bertambah hingga 1. Sehingga hal ini dapat diartikan sebagai probabilitas. Komponen input yang lebih besar akan sesuai dengan probabilitas yang lebih besar. *Softmax* sering digunakan dalam jaringan saraf untuk memetakan output non-linear dari jaringan ke distribusi probabilitas lebih dari prediksi kelas output.

2.6 *Transfer Learning*

Transfer learning adalah suatu teknik atau metode yang memanfaatkan model yang sudah dilatih terhadap suatu dataset untuk menyelesaikan permasalahan lain yang serupa dengan cara menggunakannya sebagai starting point, memodifikasi dan memperbarui parameternya sehingga sesuai dengan dataset yang baru[27]. *Transfer Learning* mampu mempercepat proses training dengan meng-

gunakan model yang sudah dikembangkan sebelumnya. model yang sudah dilatih atau dikembangkan sebelumnya disebut sebagai Pre-Trained Model. Pada tugas akhir ini menggunakan dua *Pre-Trained Models* berbeda, yaitu *MobileNet V1* dan *Inception V3*.

2.6.1 *Inception-v3*

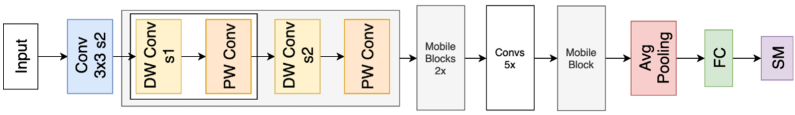


Gambar 2.9: Arsitektur dari *Inception-v3* [7]

Inception-v3 adalah salah satu arsitektur dalam *CNN* untuk melakukan *Image Classification* yang banyak digunakan yang dapat mencapai akurasi yang signifikan[27]. Arsitektur dari *Inception-v3* ditunjukkan pada gambar 2.9. Model ini memiliki campuran dari *symmetric* maupun *asymmetric building blocks*, meliputi :

1. *convolutions*
2. *average pooling*
3. *max pooling*
4. *concats*
5. *dropouts*
6. *fully connected layers*

2.6.2 *MobileNet V1*



Gambar 2.10: Arsitektur dari *MobileNet V1* [8]

MobileNet merupakan model yang sangat efisien dan dikhususkan untuk penggunaan pada *mobile device* maupun *embedded vision applications*[28]. *MobileNet* didasarkan pada arsitektur ramping yang menggunakan konvolusi yang terpisah untuk membangun *Deep Neural Network* yang ringan. *MobileNet* memperkenalkan paramater sederhana yang efektif untuk meningkatkan latensi maupun tingkat akurasi. Dengan adanya parameter ini, pengguna dapat membuat dan mengatur model berdasarkan permasalahan yang ingin diselesaikan. Model ini dapat menyelesaikan permasalahan klasifikasi, pendeteksian, dan segmentasi seperti model *Inception*. Arsitektur *MobileNet V1* dapat dilihat pada gambar 2.10.

2.7 *Confusion Matrix*

Untuk mendapatkan nilai *precision* dan *recall* dapat dilakukan dengan memanfaatkan *Confusion Matrix* seperti pada Gambar 2.11. *Confusion Matrix* menyatakan jumlah data uji yang benar dan jumlah data uji yang salah saat dilakukan proses pendeteksian. Keberadaan *confousion matrix* ini dapat digunakan untuk membantu dalam menghitung nilai-nilai *precision* dan *recall* untuk menghitung besarnya nilai rata-rata harmonik nilai *precision* dan *nilai recall* tersebut (*f-measure*). *Confusion matrix* juga berguna untuk menghitung nilai *accuracy* yang pada umumnya digunakan untuk mengukur atau menguji preformasi sebuah teknik atau metode.

		Actual	
		Positive	Negative
Predicted	Positive	True Positive (TP)	False Positive (FP)
	Negative	False Negative (FN)	True Negative (TN)

Confusion Table

Gambar 2.11: *confusion matrix* [9]

1. *Precision Precision* dapat dinyatakan sebagai data yang bernilai tepat antara prediksi dengan kondisi aktualnya dibagi dengan total data yang prediksinya bernilai tepat. Atau seperti yang dituliskan pada persamaan. 2.7.1.

$$precision = \frac{TP}{TP + FP} \tag{2.7.1}$$

2. *Recall Recall* dapat dinyatakan sebagai pembagian antara data yang bernilai tepat antara prediksi dan kondisi aktualnya dengan total data yang kondisi aktualnya bernilai tepat. Dapat dinyatakan sebagai persamaan 2.7.2.

$$recall = \frac{TP}{TP + FN} \tag{2.7.2}$$

3. *F-Measure F-measure* atau yang biasa juga disebut dengan F1 score memiliki pengertian sebagai nilai rata-rata harmonik dari nilai *precision* dan nilai *recall*. *F-measure* dapat dinyatakan sebagai persamaan 2.7.3.

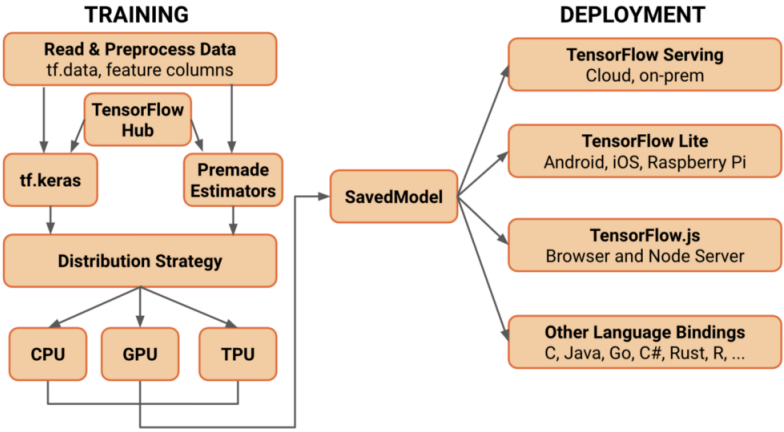
$$F - measure = \frac{2 \times precision \times recall}{precision + recall} \tag{2.7.3}$$

4. *Accuracy Accuracy* merupakan besarnya seluruh data hasil prediksi yang sesuai dengan kondisi sebenarnya terhadap seluruh data yang ada. *Accuracy* dapat dinyatakan sebagai persamaan 2.7.4.

$$accuracy = \frac{TP + TN}{totalpopulation} \tag{2.7.4}$$

2.8 *TensorFlow*

TensorFlow dibuat oleh tim dari *Google Brain*,*TensorFlow* adalah *open source library* untuk komputasi numerik dan *Machine Learning* skala besar. *TensorFlow* menggabungkan satu set model dari algoritma *Machine Learning* dan *Deep Learning*.*TensorFlow* menggunakan *Python* untuk menyediakan *front-end API* untuk membangun aplikasi dengan kerangka kerja, dengan mengeksekusi aplikasi tersebut dalam bahasa C++.



Gambar 2.12: *Workflow* pada *TensorFlow*. [10]

TensorFlow dapat melatih dan menjalankan *Deep Neural Network* untuk melakukan *Image Recognition and Classification, Natural Language Processing, Text Classification,* dan masih banyak yang lain. Aplikasi *TensorFlow* dapat dijalankan pada sebagian besar *platform*, seperti *local machine, iOS, Android, CPUs,* atau *GPUs*. Model yang dihasilkan dari *TensorFlow* dapat digunakan di hampir semua perangkat yang akan digunakan untuk melakukan prediksi. *Workflow* pada *TensorFlow* ditunjukkan pada gambar 2.12

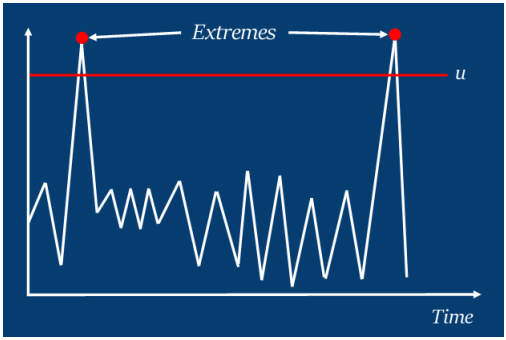
2.8.1 *Keras*

Keras adalah perpustakaan berbasis sumber terbuka yang dirancang untuk menyederhanakan model dari kerangka *Deep Learning*. Keras dapat dijalankan di atas *framework* (kerangka kerja) kecerdasan buatan seperti *TensorFlow*, *Microsoft Cognitive Toolkit*, dan *Theano*. *Keras* merupakan salah satu perpustakaan pembelajaran mesin terbaik di *Python*. Keras juga menyediakan beberapa utilitas terbaik dalam hal menyusun model, memproses dataset, memvisualisasikan grafik, dan hal lainnya. Keras dikembangkan sebagai bagian dari penelitian proyek *Open-ended Neuro-Electronic Intelligent Robot Operating System* (ONEIROS), pengelola utamanya adalah Franois Chollet, seorang insinyur *Google*. Keras berisi

banyak implementasi dari bagian-bagian *neural network* yang biasa digunakan seperti *layer, objectives, activation function, optimizers*, dan sejumlah *tool* untuk memudahkan pengolahan data citra dan teks.

2.9 *Extreme Value Theory*

Extreme Value Theory (EVT) adalah suatu metode untuk menentukan nilai VAR (*Value at Risk*) dengan mencoba menentukan sebaran dari nilai-nilai atau kejadian-kejadian ekstrim. *Extreme Value Theory* banyak digunakan dalam banyak disiplin ilmu, seperti teknik struktural, keuangan, ilmu bumi, prediksi lalu lintas, dan teknik geologi. Nilai - nilai ekstrim merupakan nilai - nilai dari pendistribusian data yang melewati batas sehingga diklasifikasikan sebagai nilai ekstrim. Seiring dengan berkembangnya teknologi, *Extreme Value Theory* jgua digunakan pada *Machine Learning* untuk sebagai *Open-Set Recognition*[29]. Ilustrasi nilai ekstrim dapat dilihat pada gambar 2.13



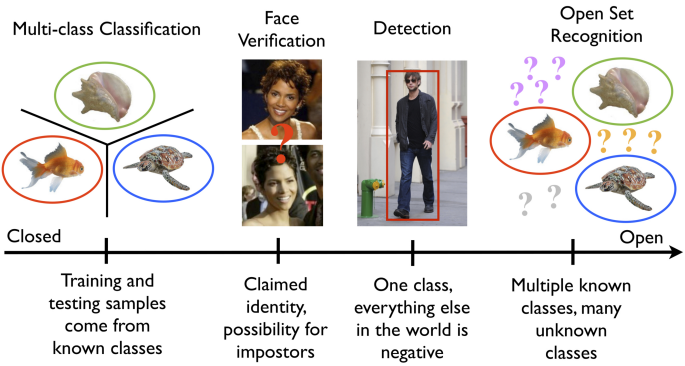
Gambar 2.13: Ilustrasi nilai ekstrim. [11]

2.9.1 *Weibull Distribution*

Distribusi Weibull merupakan salah satu dari *Extreme Value Theory* yang memiliki peranan yang penting terutama pada persoalan keandalan (*reliability*) dan analisis rawatan (*mantainability*). Distribusi Weibull sering dipakai sebagai pendekatan untuk mengetahui karakteristik fungsi kerusakan karena perubahan nilai akan mengakibatkan distribusi Weibull mempunyai sifat tertentu ataupun ekuivalen dengan distribusi tertentu. Distribusi ini adalah

distribusi serbaguna yang dapat mengambil karakteristik dari jenis lain dari distribusi, berdasarkan nilai dari bentuk parameter. Oleh karenanya, Weibull menjadi sangat berguna terutama karena fleksibilitasnya mulai dari data yang sangat tidak simetris sampai data yang mendekati distribusi normal (simetris). *Weibull Fitting* dapat diterapkan pada *Machine Learning* untuk melihat ada tidaknya *Out of Distribution* untuk mengenali data *Open-set*[29].

2.10 *Closed-set Classification* dan *Open-set Recognition*



Gambar 2.14: Ilustrasi Closed-set Classification dan Open-Set Recognition [12]

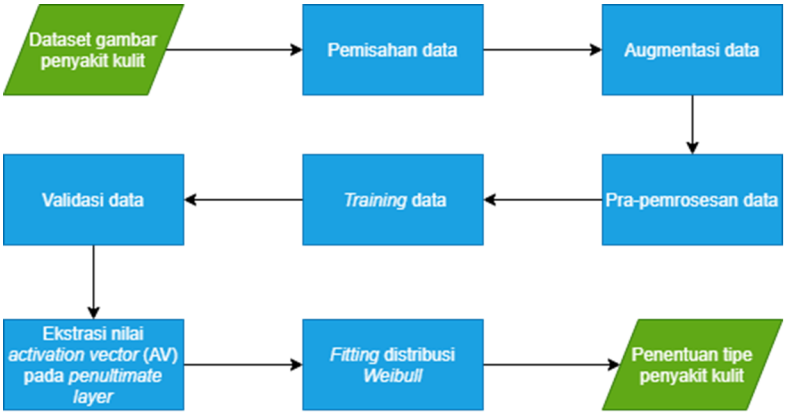
Closed-set Classification adalah sistem klasifikasi yang dimana dapat menentukan suatu objek berdasarkan kelas-kelas data yang sudah ditentukan sebelumnya. Sedangkan *Open-Set Recognition* adalah sistem yang mampu mengklasifikan data sesuai dengan kelas yang ada namun juga mengenali data-data yang tidak termasuk kedalam kelas-kelas klasifikasi[29]. Ilustrasi dari *Closed-set Classification* dan *Open-set Recognition* ada pada gambar 2.14.

BAB 3
DESAIN DAN IMPLEMENTASI SISTEM

Tugas akhir ini bertujuan untuk membuat sistem klasifikasi yang akan diterapkan pada Teledermatology sehingga sistem tersebut dapat mengenali serta mengklasifikasi penyakit kulit dan mengenali data *Open-Set*. Penelitian ini dilkasanakan sesuai dengan desain sistem dan pengimplementasiaanya yang akan dijelaskan pada subbab ini.

3.1 Desain Sistem

Secara garis besar, terdapat tujuh proses yang ada pada bagian ini atau seperti yang digambarkan pada Gambar 3.1



Gambar 3.1: Bagan Umum Metodologi Sistem

1. Pemisahan data
   Pemisahan data bertujuan untuk membagi dataset menjadi tiga jenis data, yaitu data *train*, data validasi, dan data tes.

2. Augmentasi data
   Augmentasi data bertujuan untuk menambah jumlah data train.

3. Pra-pemrosesan data
   Pra-pemrosesan data dilakukan sebelum proses *training* model, yaitu konversi *pixel* to sensor dan konversi nilai RGB.

4. *Training Data*
   Training Data merupakan sebuah proses pelatihan model *CNN* pada dataset yang sudah diolah. Pre-trained model yang digunakan dalam proses training ini adalah *MobileNet* dan *Inception V3.*

5. Validasi data
   Validasi data dari data yang telah melewati proses training kemudian akan dievaluasi untuk menilai performanya. Evaluasi model akan melakukan pengujian performasi terhadap model hasil training.

6. Ekstrasi nilai *Activation Vector* (AV) pada *Penultimate Layer*
   Ekstrasi nilai *Activation Vector* (AV) pada *Penultimate Layer* merupakan proses ekstrasi nilai pada *fully Connected Layer* yang ada pada model *training CNN.*

7. *Fitting* distribusi *Weibull*
   Fitting distribusi *Weibull* merupakan proses *fitting* pada layer yang sudah di ektrak.

Dengan mengacu pada Gambar 3.1 desain sistem ini dimulai dari pembagian dataset menjadi dua bagian yaitu data *train* dan data validasi. Data train terlebih dahulu melewati proses data augmentasi untuk menambah data dan menyeimbangkan jumlah data pada setiap kelas penyakit kulit. Selanjutnya dilakukan proses pra-pemrosesan data untuk seluruh data *train* dan data validasi. Proses berikutnya yaitu *training* menggunakan *pretrained model MobileNet v1* dan *Inception V3.* Proses training tersebut menghasilkan model *Convolutional Neural Network* (CNN) yang kemudian akan dievaluasi terlebih dahulu untuk melihat performa model hasil proses training. Jika evaluasi model hasil training memiliki performa prediksi yang cukup tinggi maka model tersebut akan diimplementasikan *Extereme Value Theory* untuk mengenali data *Open-Set* pada klasifikasi penyakit kulit, sedangkan jika evaluasi model memiliki performa prediksi rendah maka model tersebut harus di-training

ulang (*retrain*). Sebelum melakukan proses-proses yang telah disebutkan pada desain sistem, terlebih dahulu akan dijelaskan tentang dataset yang digunakan pada pembuatan sistem klasifikasi penyakit kulit ini.

3.2 Datasets DERMNET

Dalam menjalakan suatu proses *training Deep Learning* diperlukan kumpulan data - data yang berisikan *database* tunggal, atau matriks data statistik tunggal, di mana setiap kolom tabel mewakili variabel tertentu, dan setiap baris sesuai dengan anggota tertentu dari set data yang dipertanyakan. Kumpulan data mencantumkan nilai untuk masing-masing variabel, seperti tinggi dan berat objek, untuk setiap anggota kumpulan data. kumpulan data tersebut merupakan *Dataset*.

Pada tugas akhir ini, *dataset* yang digunakan adalah Dermnet, yang merupakan dataset publik yang memiliki lebih dari 23000 gambar penyakit kulit yang terdiri lebih dari 600 kelas penyakit kulit. Dari dataset Dermnet, diambil gambar 7 kelas penyakit yang umum terjadi pada negara tropis dengan total 963. Contoh gambar dataset DERMNET bisa dilihat pada gambar 3.2.

Dataset DERMNET yang digunakan dibuat dengan cara manual, dengan mengumpulkan gambar penyakit kulit yang pada website resmi DERMNET. Setelah keseluruhan dataset terkumpul, langkah selanjutnya adalah melakukan pengecekan ulang terhadap gambar yang akan digunakan untuk proses selanjutnya. Setelah memastikan bahwasannya gambar tersebut sudah sesuai dengan labelnya, maka dataset gambar tersebut bisa diproses di tahap selanjutnya.

Pada Dataset ini juga ditambahkan dataset yang dikumpulkan secara manual menggunakan alat *Dermoscope*. Pengambilan data ini dilakukan dengan mencari relawan yang mengidap penyakit yang merupakan dari 7 kelas penyakit yang akan diklasifikasikan. Kemudian relawan diminta kesedian untuk diambil gambar dari bagian kulit yang terkena penyakit menggunakan dermoscope. Data yang sudah diambil kemudian dilabel kan sesuai dengan nama penyakitnya.

Gambar 3.2: Dataset DERMNET [13]

3.2.1 *Class Dataset*

Delapan kelas penyakit kulit yang ada pada dataset ini yaitu :

1. *Acne* (acne)
   Acne merupakan gangguan pada kulit yang berhubungan dengan produksi minyak (sebum) berlebih. Jerawat terjadi ketika folikel rambut atau tempat tumbuhnya rambut tersumbat oleh minyak dan sel kulit mati. Hal tersebut menyebabkan peradangan serta penyumbatan pada pori-pori kulit. Peradangan ini ditandai dengan munculnya benjolan kecil (yang terkadang berisi nanah) di atas kulit. Gangguan kulit ini dapat terjadi di bagian tubuh dengan kelenjar minyak terbanyak,

yaitu di wajah, leher, bagian atas dada, dan punggung.

2. *Discoid eczema* (disez)
   Discoid eczema atau dermatitis numularis adalah kelainan kulit kronis yang menyebabkan munculnya bercak-bercak lingkaran seperti koin. Bercak ini terasa gatal dan memiliki permukaan kasar, namun tidak menular. Dermatitis numularis (*nummular dermatitis*) biasa muncul setelah terjadi kerusakaan pada permukaan kulit, seperti luka bakar, gesekan, atau gigitan serangga. Gangguan ini dapat berlangsung selama beberapa minggu, bulan, bahkan tahun, dan bisa kambuh kembali. *Discoid eczema* bisa terjadi di bagian tubuh mana pun. Dermatitis numularis lebih sering dialami oleh pria berusia 55-65 tahun.

3. *Atopic Dermatitis* (dmta)
   Atopic Dermatitis merupakan salah satu jenis dermatitis (eksim) yang terjadi akibat adanya peradangan pada kulit. Kondisi ini bisa disertai dengan kulit yang memerah, kering, dan pecah-pecah. Peradangan biasanya berlangsung lama, bahkan hingga bertahun-tahun. Setiap pengidap dapat merasakan gejala yang berbeda. Pada balita, gejala *Atopic Dermatitis* berupa kulit bersisik, memerah, dan berkerak di area pipi, kulit kepala, tangan dan kaki. Sedangkan pada anak-anak dan orang dewasa, gejala eksim atopik yang sering muncul adalah ruam merah dan terasa sangat gatal di area belakang leher, lutut, dan siku.

4. *Furuncle* (frncl)
   Furuncle atau bisul merupakan benjolan merah pada kulit yang terasa sakit dan berisi nanah. Benjolan ini muncul akibat infeksi bakteri yang memicu peradangan pada folikel rambut, yaitu lubang tempat tumbuhnya rambut. Bagian tubuh yang paling sering terkena bisul adalah wajah, leher, ketiak, bahu, bokong, dan paha. Ini terjadi karena bagian-bagian tersebut sering mengalami gesekan dan berkeringat. Selain itu, bisul juga bisa tumbuh pada kelopak mata. Bisul bisa terjadi pada semua golongan usia, namun kondisi ini lebih sering terjadi pada remaja dan orang dewasa muda. Jika bisul timbul da-

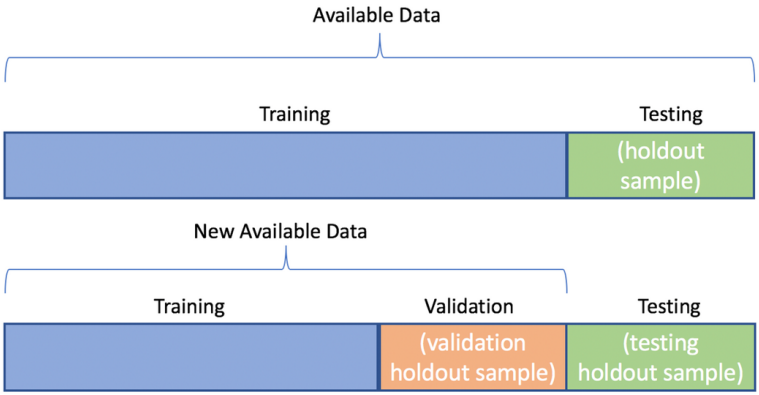
lam jumlah banyak dan berkumpul menjadi satu, kondisi ini disebut bisul sabut atau karbunkel.

5. *Herpes* (herp)
   Herpes merupakan nama kelompok virus *herpesviridae* yang dapat menginfeksi manusia. Infeksi virus *herpes* dapat ditandai dengan munculnya lepuhan kulit dan kulit kering. Pada dasarnya, ada dua macam penyakit herpes, yaitu Herpes Simplex dan Herpes Zoster. Keduanya sama-sama disebabkan oleh virus herpes, namun gejala herpes yang ditunjukkan oleh kedua jenis ini tidak sepenuhnya sama. Jenis virus *herpes* yang paling terkenal adalah *herpes simplex* virus atau HSV. *Herpes simplex* dapat menyebabkan infeksi pada daerah mulut, wajah, dan kelamin (herpes genitalia).

6. *Psoriasis* (psr)
   Psoriasis adalah peradangan pada kulit yang ditandai dengan ruam merah, kulit kering, tebal, bersisik, dan mudah terkelupas. Terkadang, *psoriasis* juga disertai dengan gatal dan nyeri. *Psoriasis* lebih sering muncul di daerah lutut, siku, punggung bagian bawah, dan kulit kepala. *Psoriasis* merupakan jenis penyakit yang dapat dialami oleh siapa saja, namun lebih sering terjadi pada orang berusia 15-35 tahun. Penyakit ini tidak menular, sehingga kontak langsung dengan ruam di kulit penderita tidak akan menyebabkan seseorang terkena penyakit ini.

7. *Ringworm* (ring)
   Ringworm atau jamur kulit adalah kelainan pada kulit yang bisa terjadi di badan, kulit kepala, kaki, juga selangkangan. Penyakit kulit ini disebabkan oleh infeksi jamur yang dikenal dengan nama *dermatofita*. *Ringworm* menyebabkan bercak atau ruam melingkar yang gatal, berwarna kemerahan, dengan bagian tepi lebih merah dan bagian tengahnya bersisik. Selain itu, infeksi jamur kulit pada manusia bisa menyebabkan area kulit kepala atau janggut menjadi botak. *Ringworm* bisa menular dari hewan ke manusia ataupun sebaliknya dan tumbuh pada bagian tubuh yang sering berkeringat karena jamur tumbuh pada area yang lembab dan hangat.

3.3 Pemisahan dan Augmentasi Data



Gambar 3.3: Visualisasi dari Pemisahan Data [14]

Dalam proses *training*, *dataset* perlu dibagi menjadi 3 set seperti yang ada pada gambar 3.3 dimana penjelasan dari masing - masing set yaitu :

1. *Training Sets*
   Training set adalah bagian *dataset* yang akan dilatih untuk membuat prediksi atau menjalankan fungsi dari sebuah algoritma *Machine learning*. ada umumnya, training sets menggunakan 80% dari jumlah keseluruhan dataset.

2. *Validation Sets*
   Validation Sets merupakan bagian dari dataset untuk mengevaluasi hasil dari *training sets* yang kemudian, akan digunakan *test sets* untuk memeriksa kembali evaluasi tersebut setelah model telah lulus atau melalui tahapan validasi.

3. *Test Sets*
   Test Sets merupakan bagian *dataset* yang diunakan untuk melakukan *testing* untuk melihat keakuratan dari hasil *training*.

Seluruh citra penyakit kulit yang terdapat dalam dataset DERMNET memiliki informasi yang dapat diolah. Informasi yang terkan-

dung dalam setiap citra memiliki kategori dan ciri khusus. Untuk mendapat informasi tersebut citra terlebih dahulu harus melalui berbagai proses ektraksi fitur (pra-pemrosesan data). Semua fitur yang telah didapatkan kemudian dijadikan satu dalam sebuah file berformat *Comma Separated Values*(.csv) yang disebut dengan metadata. Fitur-fitur yang terdapat pada metadata dataset DERMNET ditunjukan pada Tabel 3.1.

Tabel 3.1: Beberapa contoh metadata dataset DERMNET

lesion_id	*image_id*	*dx*	*dx_type*	*age*	*sex*	*localization*
SD_000340	DTA_1000044	dmta	histo	40	male	body
SD_000341	DTA_1000045	dmta	histo	40	male	body
SD_000341	DTA_1000046	dmta	histo	40	male	body
SD_000342	DTA_1000047	dmta	histo	40	male	body
SD_000343	DTA_1000048	dmta	histo	40	male	body

Proses pemisahan data ini akan memberikan fitur baru yaitu *duplicate* pada tabel metadata seperti pada Tabel 3.2.

Tabel 3.2: Beberapa contoh metadata dataset DERMNETsetelah proses pemisahan data

lesion_id	*image_id*	*dx*	*dx_type*	*age*	*sex*	*localization*	*duplicate*
SD_000340	DTA_1000044	dmta	histo	40	male	body	no duplicate
SD_000341	DTA_1000045	dmta	histo	40	male	body	has duplicate
SD_000341	DTA_1000046	dmta	histo	40	male	body	has duplicate
SD_000342	DTA_1000047	dmta	histo	40	male	body	no duplicate
SD_000343	DTA_1000048	dmta	histo	40	male	body	noduplicate

Metadata dataset DERMNET memiliki tujuh fitur data yakni *lession id, image id, dx, dx type, age, sex, localization.* Fitur *lession id* merupakan citra penyakit kulit yang mencirikan citra tersebut berasal dari satu orang sedangkan *image id* merupakan nomor identitas yang mencirikan citra penyakit kulit tersebut. Data citra penyakit kulit yang memiliki fitur *lession id* yang sama dengan datapenyakit kulit lainnya disebut citra *duplicate.* Citra *duplicate* menandakan bahwa citra penyakit kulit tersebut diambil dari satu orang yang sama tetapi memiliki cara pengambilan citra yang berbeda. Proses pemisahan data (Data Split) akan mengacu pada citra penyakit kulit yang memiliki *duplicate* (*has duplicate*) dan citra penyakit kulit yang tidak memiliki *duplicate* (*no duplicate*).

Tabel 3.3: *Tabel hasil pembagian data*

Class Data	Data Training	Data Validation
Psoriosis (psr)	191	26
Acne (acne)	162	45
Ringworm (ring)	169	25
Atopic Dermatitis (dmta)	92	28
Discoid Eczema (disez)	90	15
Herpes (herp)	61	9
Furuncle (frncl)	41	9
TOTAL	**808**	**157**

Tabel 3.4: *Tabel hasil penambahan data*

Class Data	Data Training	Data Generator
Psoriosis (psr)	191	5720
Acne (acne)	162	5240
Ringworm (ring)	169	5150
Atopic Dermatitis (dmta)	92	4210
Discoid Eczema (disez)	90	5510
Herpes (herp)	61	3650
Furuncle (frncl)	41	5203
TOTAL	**808**	**34683**

Dari hasil data split diperoleh hasil bahwa citra penyakit yang memiliki *duplicate* (*has duplicate*) berjumlah 190 citra dan citra penyakit kulit yang tidak memiliki *duplicate* (*no duplicate*) berjumlah 190 citra. Dari hasil data split tersebut ,citra yang tidak memiliki duplicate akan dibagi secara acak dengan proporsi pembagian 17% yang akan dijadikan data validation sekaligus data test dan sisa citra akan digabung dengan citra yang memiliki duplicate untuk dijadikan data train. Sehingga secara keseluruhan diperoleh hasil seperti pada Tabel 3.3.

Dari hasil data split diperoleh untuk data train adalah 808 dan untuk data validasi sekaligus data tes berjumlah 157. Data train kemudian dilakukan proses penambahan data data (augmentation) menggunakan metode data generator. Hasil penambahan data setiap class akan ditunjukan pada Tabel 3.4

3.4 Pra-pemrosesan Data

Setelah data train yang akan digunakan untuk proses training siap, tahap berikutnya adalah melakukan pra-pemrosesan train dan data validasi. Hal pertama yang dilakukan dalam pra-pemrosesan data ini adalah melakukan *image-resize*. Citra penyakit pada dataset DERMNET diubah menjadi 224x224 pixel. Seluruh citra penyakit kulit yang telah di resize kemudian harus melewati tahap *rescale* nilai RGB, skala nilai RGB pada citra semulanya adalah 0-244 kemudian skala tersebut dirubah menjadi skala nilai -1, 0 dan 1. Perubahan nilai skala RGB dilakukan dengan cara membagi nilai tensor asli dengan setengah dari nilai maksimal RGB yaitu 127,5 hal ini akan merubah semua nilai tensor menjadi skala -1 hingga 1. Selanjutnya dimensi tensor akan diperluas kemudian dikembalikan ke tensor yang dihasilkan dari operasi resclae. Proses resize dan rescale nilai RGB ini juga akan diterapkan pada citra penyakit kulit input.

3.5 *Training Data*

Pada proses *training* data ini digunakan data train yang telah dilakukan proses data *augmantation*. Proses data *augmantation* akan menambah jumlah data train yang semula hanya berjumlah 808 gambar menjadi 34.683 gambar. Data hasil *agumentation* akan digunakan dalam proses *training* sedangkan jumlah data awal masih digunakan dalam penentuan ukuran *train steps*. Pada proses *training* model perlu dilakukan penentuan ukuran *batch size*, *train steps* dan *epoch* yang akan dijelaskan sebagai berikut:

1. *Batch Size*
   Penentuan batch size berdasarkan jumlah sampel data yang ingin disebarkan ke seluruh neural network dalam setiap iterasi (train steps). Pada proses training kali ini ukuran batch size ditentukan sejumlah 10, yang berarti setiap satu step akan disebarkan 10 data training ke neural network.

2. *Train Steps*
   Jumlah data dalam persamaan tersebut adalah jumlah awal data train sebelum proses data augmantation. Dari operasi persamaan tersebut dihasilkan ukuran train steps pada proses training model kali ini adalah 3468. Ukuran train steps

tersebut ditentukan dari persamaan 3.5.1.

$$TrainSteps = \frac{Jumlahdata}{BatchSize} \tag{3.5.1}$$

$$TrainSteps = \frac{34683}{10} = 3468 \tag{3.5.2}$$

3. *Epoch*
   Epoch merupakan satu set putaran training steps. Beberapa epoch diperlukan untuk pelatihan sebuah neural network sehingga didapatkan kesalahan (loss) mendekati nol. Epoch pada proses training model ini didefinisikan sejumlah 30 kali.

3.5.1 Load Pretrained Model

Setelah dilakukan penentuan ukuran batch size, train steps dan epoch tahap selanjutnya dalam proses training adalah melakukan proses load pretrained model. Pretrained model yang digunakan pada penelitian ini adalah MobileNet v1 dan Inception V3. Proses load pretrained model ini dapat dilakukan dengan cara memanggil class yang ada pada Keras API. Class tersebut akan memuat semua layer yang terdapat pada arsitektur MobileNet v1 dan Inception V3. Berikut code untuk memuat layer MobileNet v1 dan Inception V3.

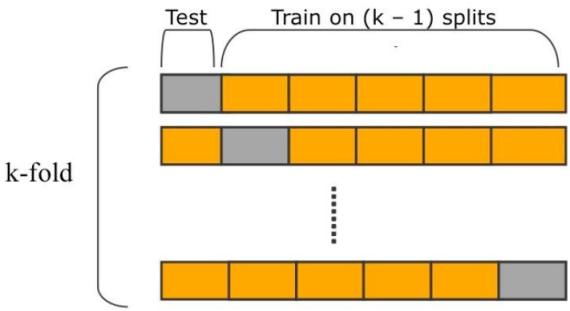
```
from tensorflow.keras.applications.mobilenet import MobileNet


from tensorflow.python.keras.applications import InceptionV3
```

3.6 Validasi Data

Setelah proses *training* data selesai, proses selanjutnya adalah melakukan validasi terhadap model hasil *training* tersebut. validasi ini bertujuan untuk melakukan pengecekan terhadap proses training model yang telah dilakukan. Untuk memvalidasi model, digunakan metode *Cross Validation*. Metode ini merupakan metode statistik untuk mengevaluasi dan membandingkan algoritma learning. Dalam *Cross Validation*, data *train* dan data validasi disilangkan dalam putaran berturut-turut sehingga setiap node data memiliki peluang untuk divalidasi. Dalam setiap satu iterasi akan menggunakan lipatan pertama (k-1) data untuk mempelajari satu model, dan

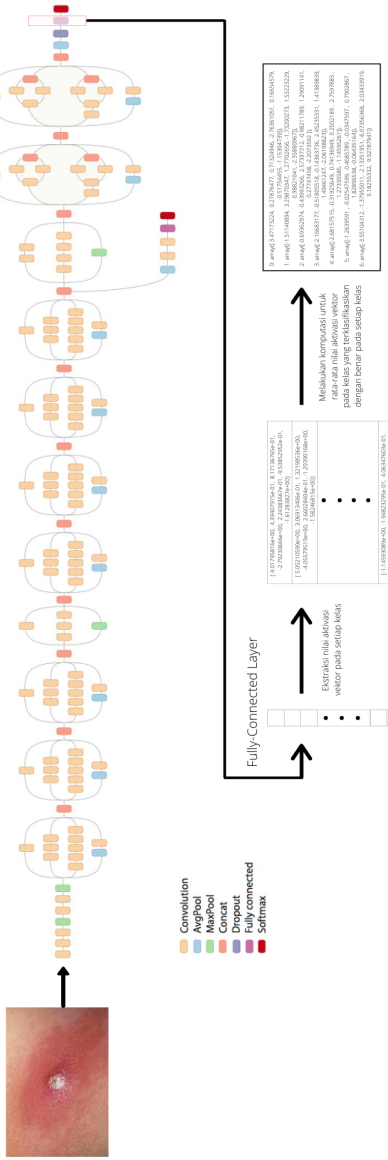
selanjutnya model yang dipelajari diminta untuk membuat prediksi tentang data tersebut. Performa setiap algoritma learning pada setiap lipatan akan dapat diketahui akurasinya menggunakan *confusion matrix*. Jika model tersebut memiliki tingkat akurasi rendah maka akan dilakukan konfigurasi ulang paramater yang digunakan dan kemudian akan dilakukan training ulang. Ilustrasi dari *cross validation* dapat dilihat pada gambar 3.4



Gambar 3.4: ilustrasi *Cross Validation* [15]

3.7 Ekstrasi nilai *Activation Vector* (AV) pada *Penultimate Layer*

Proses selanjutnya adalah mengekstrak fitur dari *Fully Connected Layer*(FC) pada model *training* yang sudah divalidasi.FC layer yang di ekstrak adalah layer yang sebelum memasuki layer aktivasi fungsi softmax. Ekstrakrasi fitur dari layer yang sebelum memasuki layer aktivasi fungsi *softmax* karena ketika memasuki layer aktivasi fungsi *softmax*, *softmax* menjalakan komputasi yang mengubah seluruh kelas pada data input bernilai 1. Yang berarti nantinya seluruh gambar yang dijadikan input akan dimasukan dalam salah satu kelas yang ada walaupun gambar tersebut bisa jadi bukanlah gambar yang sesaui dengan gambar pendistribusian data training.



Gambar 3.5: Ilustrasi Ekstrasi nilai Aactivation Vector

Berdasarkan arsitektur pada gambar 3.5, nilai pada *fully Connected Layer* atau yang disebut dengan *activation vector* dari setiap kelas diambil. Kemudian dicari nilai AV dari setiap hasil *training* yang terklasifikasi dengan benar pada penultimate layer untuk setiap kelas data. Selanjutnya melakukan komputasi untuk mendapatkan nilai rata-rata dari AV yang sudah terklasifikasi dengan benar, dimana nilai ini nantinya akan digunakan pada *fitting* distribusi *weibull*. Algoritma pada ekstrasi nilai AV ada pada algoritma 1 [**?**].

Algorithm 1: Algoritma ekstrasi nilai AV

Require: Nilai aktivasi vector pada penultimate layer $V(X) = v_1(x)...v_N(x)$

Require: Untuk setiap kelas j maka $S_{i,j} = v_j(x_{i,j})$ dimana setiap kelas yang terklasifikasi benar $x_{i,j}$

1: for $j = 1...N$ do

2: Komputasi *mean* AV $\mu = mean_i(S_{i,j})$

3.8 *Fitting* distribusi *Weibull*

Selanjutnya adalah pengimplementasian dari *Weibul Fitting* pada model layer yang sudah diekstrak sebelumnya. Pada implementasi ini digunakan fungsi *FitHigh* dari libMR untuk memanggil fungsi *Weibul Fitting*[30]. libMR menyediakan fungsionalitas inti *MetaRecognition* dan distribusi *Weibull*. Ini digunakan untuk menghitung skor-w yang digunakan untuk multi-modal fusion, merenormalkan data SVM dan secara umum mendukung algoritma *open-set* dengan deteksi outlier.

Pada tahap ini, *Weibull Fitting* akan digunakan pada nilai rata-rata AV dan nilai AV pada kelas yang sudah terklasifikasi dengan benar. Pada *Weibull Fitting* kedua nilai tersebut, ditentukan jarak terbesar dimana nilai dari jarak terbesar ini akan menjadi *treshold* nilai dari gambar yang diklasifikasikan. Apabila gambar yang nantinya akan diinputkan memiliki nilai AV dengan nilai yang tidak termasuk dalam threshold yang ditentukan maka gambar tersebut akan diklasifikasikan sebagai gambar yang tidak diketahui (unknown). Algoritma dari implementasi *Weibull Fitting* ada pada algoritma 2 [31].

Algorithm 2: Algoritma implementasi *Weibull Fitting*, dengan η merupakan jarak terbesar dari *Weibull Fitting* pada rata-rata AV. Model LibMR ρ_j dengan parameter τ_i untuk pergeseran data, juga *Weibull shape* dan parameter skala:κ_i, λ_i.

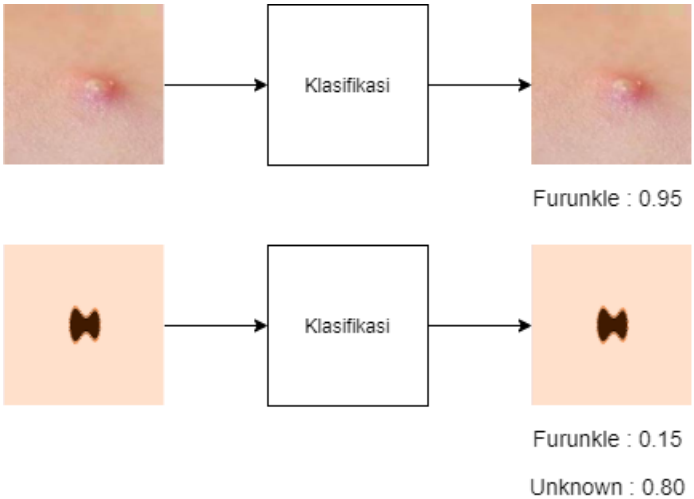
Require: FitHigh function dari libMR.

1: for $j = 1...N$ do

2: *Weibull Fitting* $\rho = (\tau_i, \kappa_i, \lambda_i) = \text{FitHigh}(\|S_j - \mu_j\|, \eta)$

3.9 Penentuan Tipe Penyakit Kulit

Tahap selanjut adalah pembuatan penentuan tipe penyakit kulit. Sistem ini akan berperan sebagai sistem *classifier* terhadap gambar penyakit kulit yang akan diklasifikasikan. Pada sistem ini juga dapat mengenali gambar yang open-set yang tidak berkaitan dengan penyakit kulit ketika di-inputkan seperti pada gambar 3.6.



Gambar 3.6: Ilustrasi Image Classification

Halaman ini sengaja dikosongkan

BAB 4
PENGUJIAN DAN ANALISA

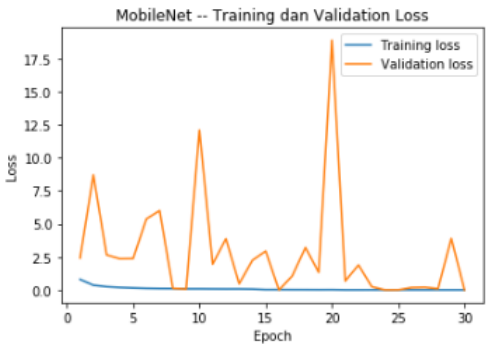
Pada bab ini dipaparkan hasil pengujian serta analisa dari desain sistem dan implementasi. Pengujian dilakukan guna mengetahui performa dari model yang telah dibuat, dan mengetahui performa dari implementasi *Extreme Value Theory* dalam mengenali data Open-Set.

4.1 Pengujian Jenis Model

Pengujian pada jenis model bertujuan untuk mengetahui performa dan akurasi dari setiap model yang dihasilkan.

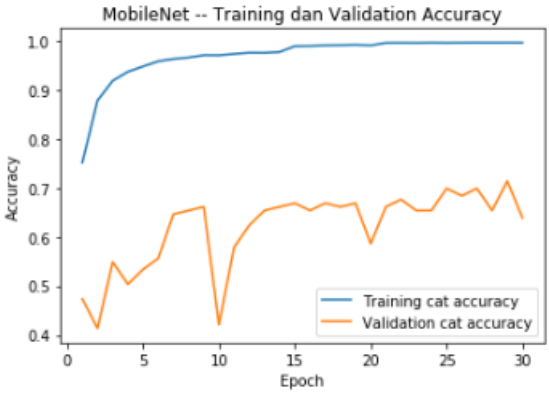
4.1.1 Pengujian Model *MobileNet* v1

Pada pengujian model *MobileNet* v1 hasil dari proses *training* dilakukan untuk evaluasi performa model dalam melakukan klasifikasi citra medis penyakit kulit. Jika performa model memiliki akurasi tinggi maka model akan digunakan dalam sistem klasifikasi sedangkan jika performa model memiliki akurasi rendah maka perlu dilakukan proses training ulang. Langkah pertama untuk melakukan pengujian terhadap model *MobileNet* v1 ini yaitu dengan melakukan analisa *cross-validation* history saat proses *training*. Dari grafik Gambar 4.1 menunjukan bahwa nilai loss pada proses *training MobileNet* v1 terus mengalami penurunan walaupun tidak signifikan.



Gambar 4.1: Grafik nilai *loss* training Mobilenet V1

Dari grafik Gambar 4.2 menunjukan bahwa nilai akurasi pada proses *training MobileNet* v1 tidak mengalami kenaikan yang signifikan. Dapat disimpulkan bahwa model MobileNet v1 pada proses training mengalami *overfitting* sehingga model *MobileNet* v1 mempunyai nilai performansi yang kurang optimal.



Gambar 4.2: Grafik nilai *accuracy* training Mobilenet V1

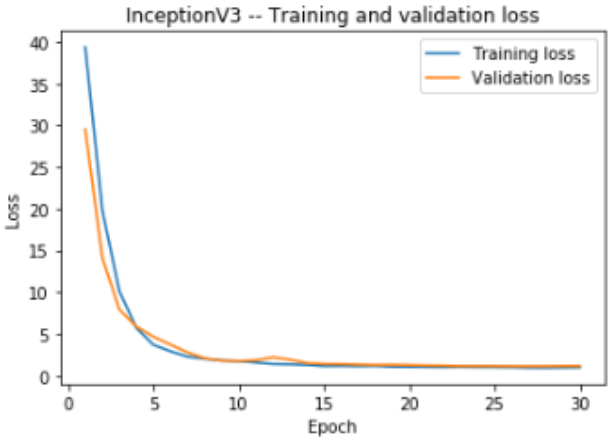
Tabel 4.1 menunjukan nilai *Precision, Recall, F1-Score,* nilai rata-rata mikro dan makro serta nilai rata-rata *weight* pada setiap kelas penyakit kulit.

Tabel 4.1: Tabel nilai *confusion matrix* Mobilenet V1

	precision	*Recall*	*F1-Score*	*Support*
Furuncle (frncl)	0.80	0.57	0.67	9
Atopik Dermatitis (dmta)	0.71	0.75	0.73	28
Discoid Eczema (disez)	0.45	0.36	0.40	15
Herpes (herp)	1.00	0.70	0.82	9
Psoriasis (psr)	0.60	0.64	0.62	26
Ringworm (ring)	0.64	0.75	0.69	25
Acne (acne)	0.89	0.91	0.90	45
accuracy			0.71	157
Macro avg	0.73	0.67	0.69	157
Weighted avg	0.72	0.71	0.71	157

4.1.2 Pengujian Model *Inception* V3

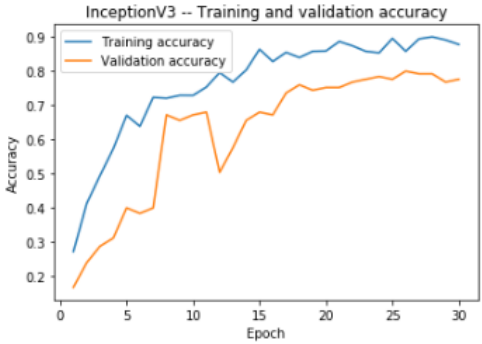
Pada pengujian model *Inception* V3 hasil dari proses training dilakukan untuk evaluasi performa model dalam melakukan klasifikasi citra medis penyakit kulit. Jika performa model memiliki akurasi tinggi maka model akan digunakan dalam sistem klasifikasi sedangkan jika performa model memiliki akurasi rendah maka perlu dilakukan proses training ulang. Langkah pertama untuk melakukan pengujian terhadap model *Inception* V3 ini yaitu dengan melakukan analisa *cross-validation* history saat proses *training*. Dari grafik Gambar 4.3 menunjukan bahwa nilai loss pada proses *training Inception* V3 terus mengalami penurunan.



Gambar 4.3: Grafik nilai *loss* training Inception V3

Dari grafik Gambar 4.4 menunjukan bahwa nilai akurasi pada proses *training Inception* V3 terus mengalami kenaikan. Dapat disimpulkan bahwa model *Inception* V3 pada proses training tidak mengalami *underfitting* namun maupun *overfitting*, sehingga model *Inception* V3 mempunyai nilai performansi yang optimal.

Tabel 4.2 menunjukan nilai *Precision, Recall, F1-Score*, nilai rata-rata mikro dan makro serta nilai rata-rata *weight* pada setiap kelas penyakit kulit.



Gambar 4.4: Grafik nilai *accuracy* training Inception V3

Tabel 4.2: Tabel nilai *confusion matrix* Inception V3

	precision	*Recall*	*F1-Score*	*Support*
Furuncle (frncl)	0.90	1.00	0.95	9
Atopik Dermatitis (dmta)	0.88	0.79	0.83	28
Discoid Eczema (disez)	0.71	0.67	0.69	15
Herpes (herp)	0.89	0.89	0.89	9
Psoriasis (psr)	0.69	0.85	0.76	26
Ringworm (ring)	0.87	0.80	0.83	25
Acne (acne)	0.98	0.96	0.97	45
accuracy			0.85	157
Macro avg	0.85	0.85	0.84	157
Weighted avg	0.86	0.85	0.85	157

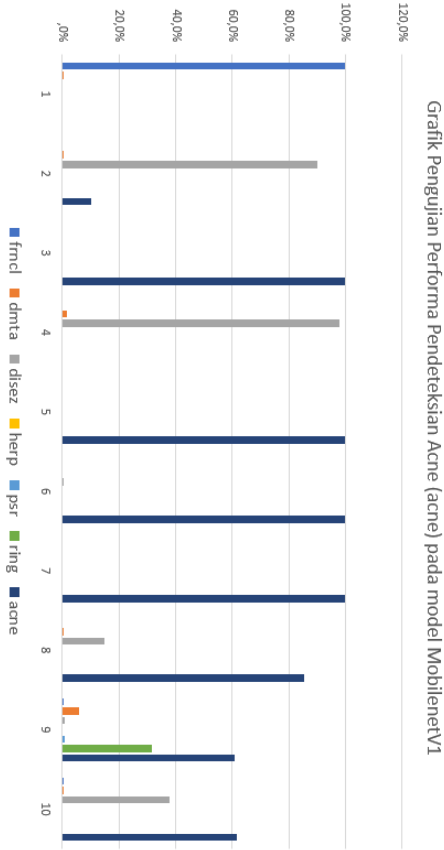
4.2 Pengujian Hasil Klasifikasi

Pengujian ini akan mengukur performansi model *MobileNet* v1 dan *Inception* V3 berdasarkan nilai klasifikasi yang didapatkan dari citra *dermoscopic input*. Data tes yang akan di-input merupakan *citra dermoscopic* dari tujuh kelas penyakit kulit yang berasal dari dataset DERMNET. Pengujian pada kelas penyakit kulit Acne (acne), Discoid Eczema (disez), Atopik Dermatitis (dmta), Ringworm (ring), Herpes (herp), Psoriasis (psr), dilakukan sebanyak sepuluh kali menggunakan sepuluh citra dermoscopic yang berbeda. Pada kelas penyakit kulit Furuncle (frncl) dilakukan pengujian sebanyak tujuh kali menggunakan 7 citra dermoscopic berbeda.

4.2.1 Hasil Klasifikasi *MobileNet* V1

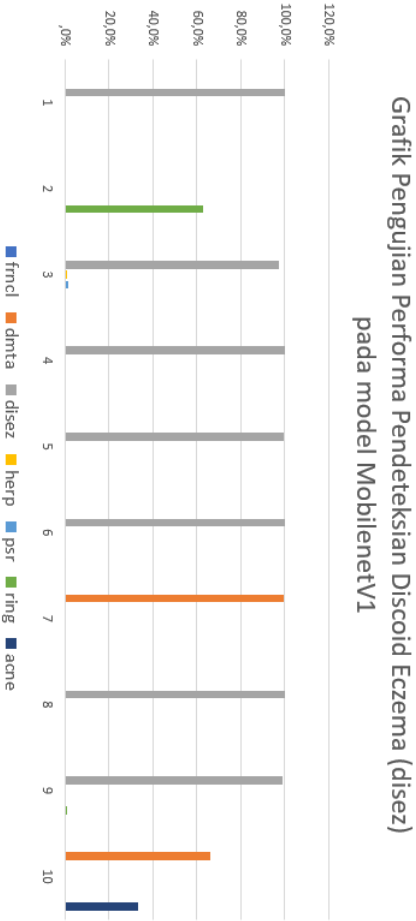
berikut hasil prediksi model *MobileNet* v1 terhadap skenario pengujian:

Dari grafik Gambar 4.5 menunjukan pengujian klasifikasi penyakit kulit kelas Acne (acne). Dari sepuluh kali percobaan menggunakan data tes, model *MobileNet* v1 berhasil melakukan **klasifikasi benar sebanyak tujuh kali** pada citra uji 3, 5 , 6, 7, 8, 9, 10 dan **klasifikasi salah sebanyak tiga kali** pada citra uji 1, 2, 4.



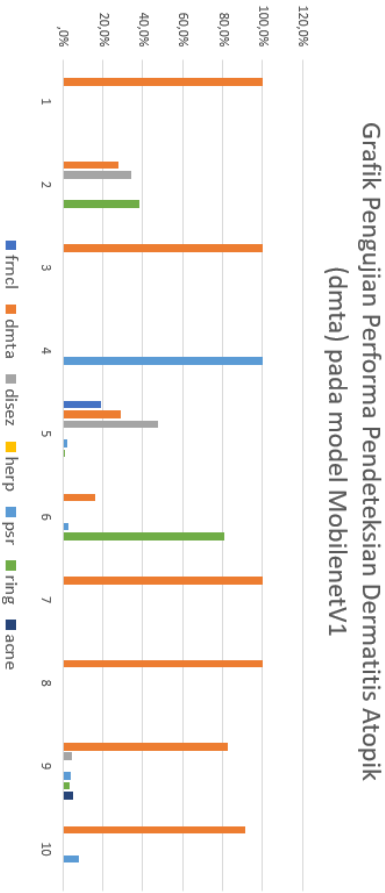
Gambar 4.5: Grafik nilai klasifikasi *acne* MobileNet V1

Dari grafik Gambar 4.6 menunjukan pengujian klasifikasi penyakit kulit kelas Discoid Eczema (disez). Dari sepuluh kali percobaan menggunakan data tes, model *MobileNet* v1 berhasil melakukan **klasifikasi benar sebanyak tujuh kali** pada citra uji 1, 3 , 4, 5, 6, 8, 9 dan **klasifikasi salah sebanyak tiga kali** pada citra uji 2, 7, 10.



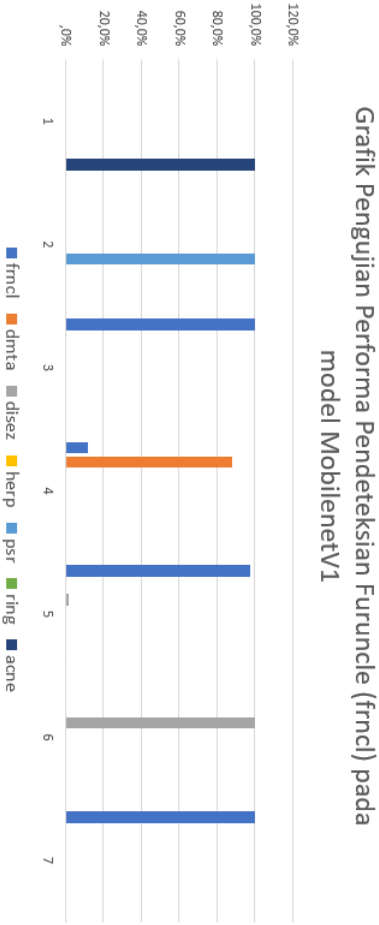
Gambar 4.6: Grafik nilai klasifikasi *disez* MobileNet V1

Dari grafik Gambar 4.7 menunjukan pengujian klasifikasi penyakit kulit kelas Atopik Dermatitis (dmta). Dari sepuluh kali percobaan menggunakan data tes, model *MobileNet* v1 berhasil melakukan **klasifikasi benar sebanyak enam kali** pada citra uji 1, 3 , 7, 8, 9, 10 dan **klasifikasi salah sebanyak empat kali** pada citra uji 2, 4, 5, 6.



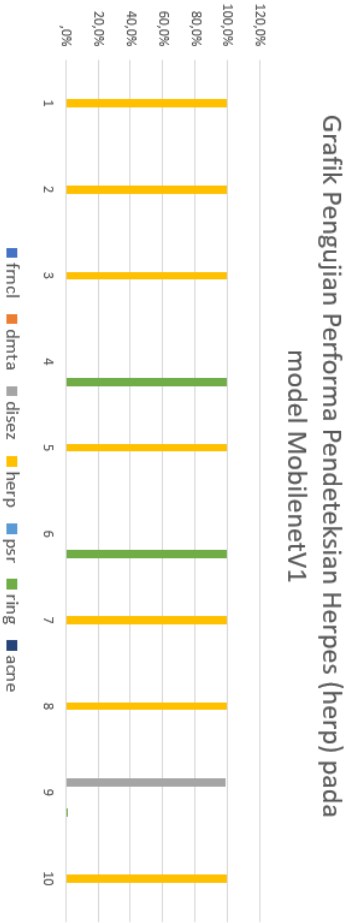
Gambar 4.7: Grafik nilai klasifikasi *dmta* MobileNet V1

Dari grafik Gambar 4.8 menunjukan pengujian klasifikasi penyakit kulit kelas Furuncle (frncl). Dari sepuluh kali percobaan menggunakan data tes, model *MobileNet* v1 berhasil melakukan **klasifikasi benar sebanyak tiga kali** pada citra uji 3, 5, 7 dan **klasifikasi salah sebanyak tiga kali** pada citra uji 1, 2, 4.



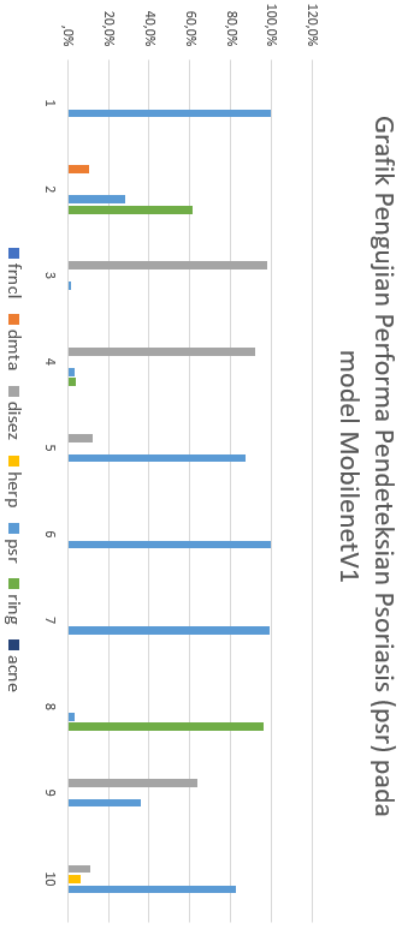
Gambar 4.8: Grafik nilai klasifikasi *frncl* MobileNet V1

Dari grafik Gambar 4.9 menunjukan pengujian klasifikasi penyakit kulit kelas Herpes (herp). Dari sepuluh kali percobaan menggunakan data tes, model *MobileNet* v1 berhasil melakukan **klasifikasi benar sebanyak tujuh kali** pada citra uji 1, 3 , 5, 7, 8, 10 dan **klasifikasi salah sebanyak tiga kali** pada citra uji 4, 6, 9.



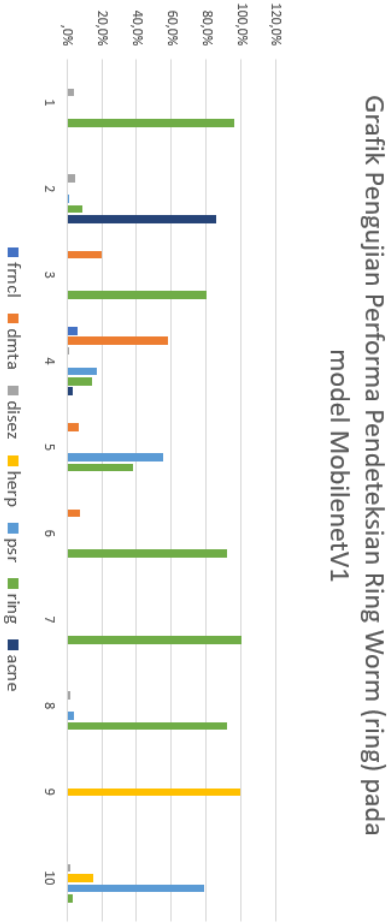
Gambar 4.9: Grafik nilai klasifikasi *herp* MobileNet V1

Dari grafik Gambar 4.10 menunjukan pengujian klasifikasi penyakit kulit kelas Psoriasis (psr). Dari sepuluh kali percobaan menggunakan data tes, model *MobileNet* v1 berhasil melakukan **klasifikasi benar sebanyak lima kali** pada citra uji 1, 5 , 6, 7, 10 dan **klasifikasi salah sebanyak lima kali** pada citra uji 2, 3, 4, 8, 9.



Gambar 4.10: Grafik nilai klasifikasi *psr* MobileNet V1

Dari grafik Gambar 4.11 menunjukan pengujian klasifikasi penyakit kulit kelas Ringworm (ring). Dari sepuluh kali percobaan menggunakan data tes, model *MobileNet* v1 berhasil melakukan **klasifikasi benar sebanyak lima kali** pada citra uji 1, 3, 6, 7, 8 dan **klasifikasi salah sebanyak lima kali** pada citra uji 2, 4, 5, 9, 10.

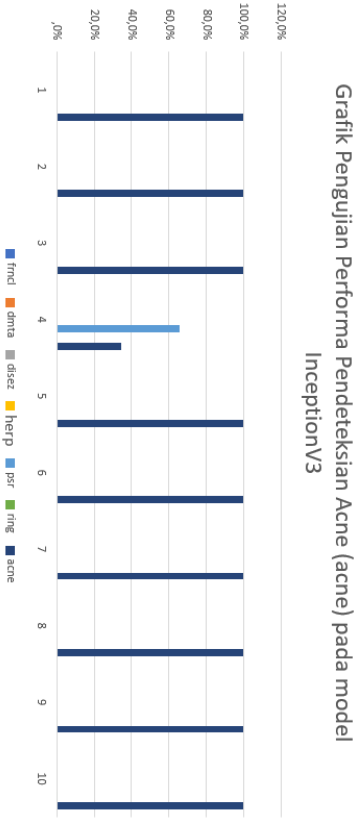


Gambar 4.11: Grafik nilai klasifikasi *ring* MobileNet V1

4.2.2 Hasil Prediksi *Inception* V3

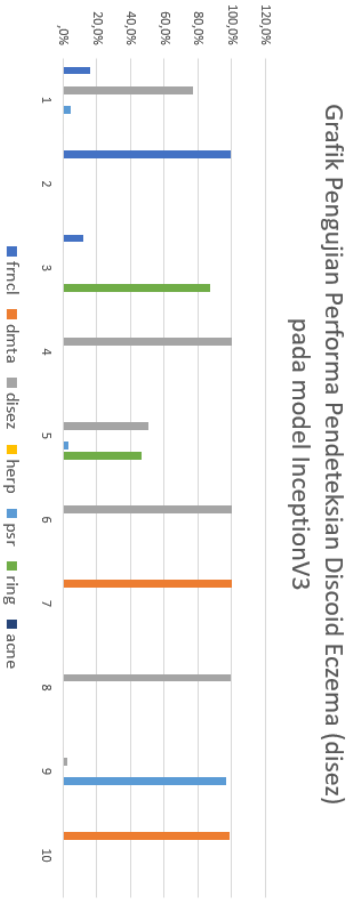
berikut hasil klasifikasi model Inception V3 terhadap skenario pengujian:

Dari grafik Gambar 4.12 menunjukan pengujian klasifikasi penyakit kulit kelas Acne (acne). Dari sepuluh kali percobaan menggunakan data tes, model MobileNet v1 berhasil melakukan **klasifikasi benar sebanyak sembilan kali** pada citra uji 1, 2, 3, 5, 6, 7, 8, 9, 10 dan **klasifikasi salah sebanyak satu kali** pada citra uji 4.



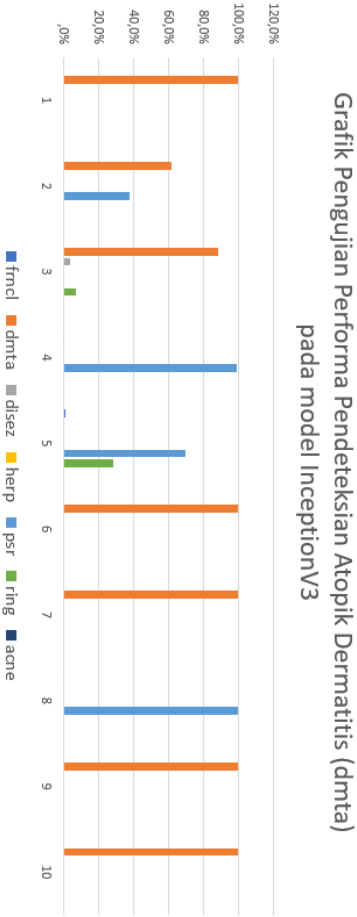
Gambar 4.12: Grafik nilai klasifikasi *acne* Inception V3

Dari grafik Gambar 4.13 menunjukan pengujian klasifikasi penyakit kulit kelas Discoid Eczema (disez). Dari sepuluh kali percobaan menggunakan data tes, model MobileNet v1 berhasil melakukan **klasifikasi benar sebanyak lima kali** pada citra uji 1, 4, 5, 6, 8 dan **klasifikasi salah sebanyak lima kali** pada citra uji 2, 3, 7, 9, 10.



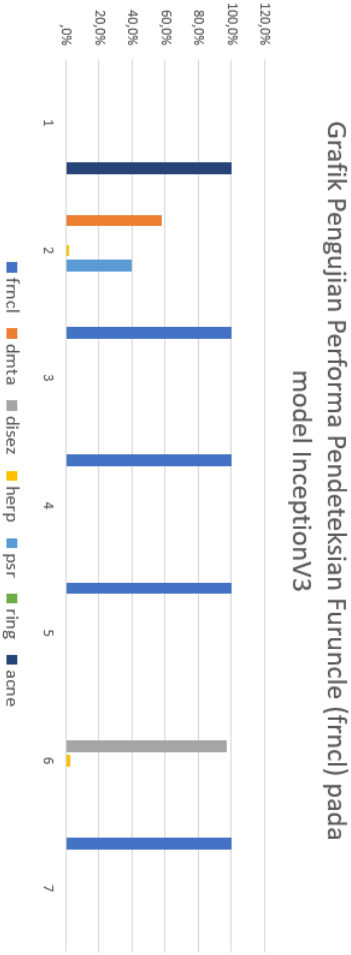
Gambar 4.13: Grafik nilai klasifikasi *disez* Inception V3

Dari grafik Gambar 4.14 menunjukan pengujian klasifikasi penyakit kulit kelas Atopik Dermatitis (dmta). Dari sepuluh kali percobaan menggunakan data tes, model MobileNet v1 berhasil melakukan **klasifikasi benar sebanyak tujuh kali** pada citra uji 1, 2, 3, 6, 7, 9, 10 dan **klasifikasi salah sebanyak tiga kali** pada citra uji 4, 5, 8.



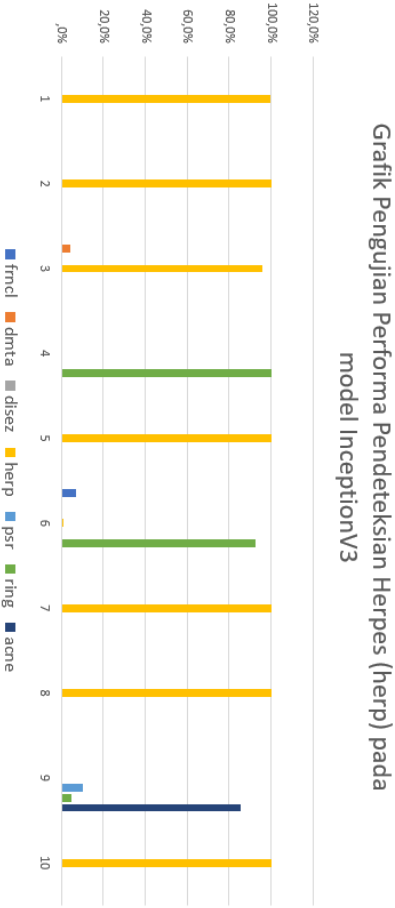
Gambar 4.14: Grafik nilai klasifikasi *dmta* Inception V3

Dari grafik Gambar 4.15 menunjukan pengujian klasifikasi penyakit kulit kelas Furuncle (frncl). Dari sepuluh kali percobaan menggunakan data tes, model MobileNet v1 berhasil melakukan **klasifikasi benar sebanyak empat kali** pada citra uji 3, 4, 5, 7 dan **klasifikasi salah sebanyak tiga kali** pada citra uji 1, 2, 6.



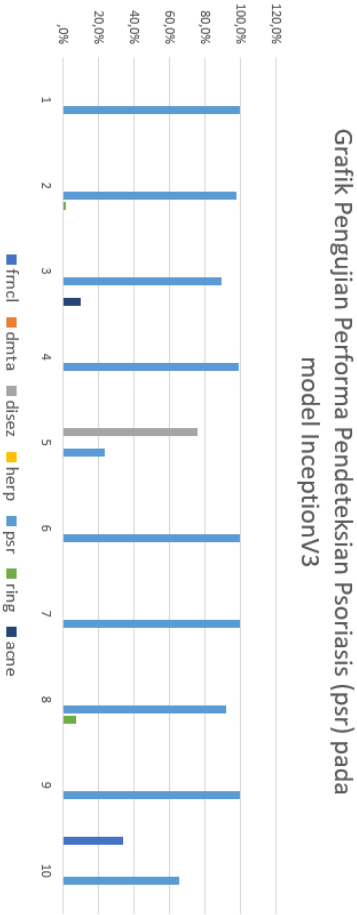
Gambar 4.15: Grafik nilai klasifikasi *frncl* Inception V3

Dari grafik Gambar 4.16 menunjukan pengujian klasifikasi penyakit kulit kelas Herpes (herp). Dari sepuluh kali percobaan menggunakan data tes, model MobileNet v1 berhasil melakukan **klasifikasi benar sebanyak tujuh kali** pada citra uji 1, 2, 3, 5, 7, 8, 10 dan **klasifikasi salah sebanyak tiga kali** pada citra uji 4, 6, 9.



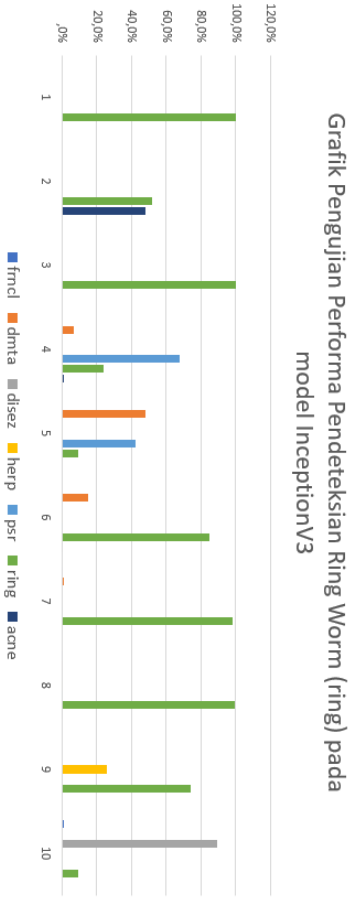
Gambar 4.16: Grafik nilai klasifikasi *herp* Inception V3

Dari grafik Gambar 4.17 menunjukan pengujian klasifikasi penyakit kulit kelas Psoriasis (psr). Dari sepuluh kali percobaan menggunakan data tes, model MobileNet v1 berhasil melakukan **klasifikasi benar sebanyak sembilan kali** pada citra uji 1, 2, 3, 4, 6, 7, 8, 9, 10 dan **klasifikasi salah sebanyak satu kali** pada citra uji 5.



Gambar 4.17: Grafik nilai klasifikasi *psr* Inception V3

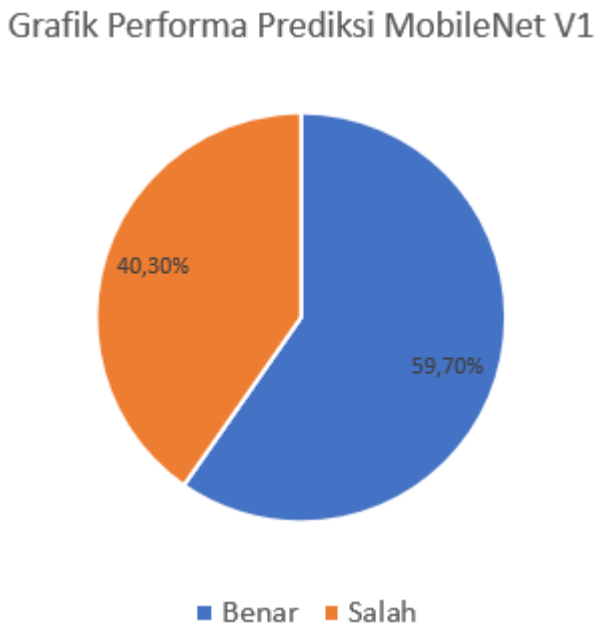
Dari grafik Gambar 4.18 menunjukan pengujian klasifikasi penyakit kulit kelas Ringworm (ring). Dari sepuluh kali percobaan menggunakan data tes, model MobileNet v1 berhasil melakukan **klasifikasi benar sebanyak tujuh kali** pada citra uji 1, 2, 3, 6, 7, 8, 9 dan **klasifikasi salah sebanyak tiga kali** pada citra uji 4, 5, 10.



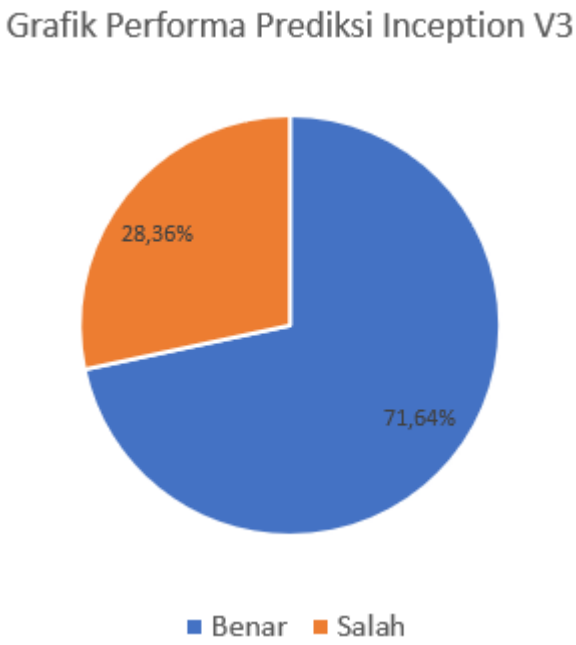
Gambar 4.18: Grafik nilai klasifikasi *ring* Inception V3

4.3 Analisa Perbandingan Model

Setelah dilakukan uji performa klasifikasi didapatkan nilai rata-rata performa model yang dapat dilihat pada Gambar 4.19 dan Gambar 4.20. klasifikasi benar pada model *Inception* V3 mencapai 71,64% sedangkan nilai klasifikasi benar pada *MobileNet* v1 hanya 59,70%. Nilai klasifikasi tersebut juga berbanding lurus dengan nilai klasifikasi salah, pada model *Inception* V3 nilai klasifikasi salah hanya sekitar 28,36% sedangkan pada model *MobileNet* v1 nilai klasifikasi salah mencapai 40,30%. Sehingga dapat disimpulkan bahwa hasil model *Inception* V3 memiliki performa dalam prediksi citra input yang lebih baik jika dibandingkan performa klasifikasi model *MobileNet* v1.



Gambar 4.19: Grafik rata-rata performa klasifikasi *MobileNet* v1(%)



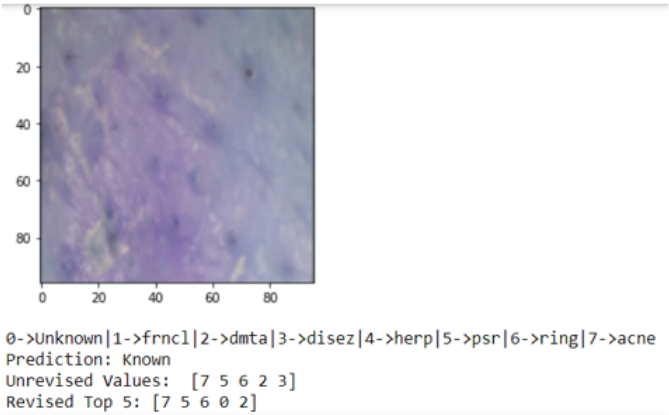
Gambar 4.20: Grafik rata-rata performa klasifikasi *Inception* v3(%)

4.4 Pengujian Hasil Implementasi *Extreme Value Theory* Pada Proses Klasifikasi (*Open-set Recognition*)

Pengujian ini akan mengukur performansi hasil implementasi *Extreme Value Theory* pada proses klasifikasi dalam menentukan gambar penyakit kulit dan gambar *open-set*. Data tes yang akan di-input adalah Dataset DERMNET sebagai data *closed-set* yang akan diklasifikasikan, dataset STL-10 [32] sebagai data *open-set* dan dataset HAM10000[33] sebagai data *open-set* penyakit kulit. Pengujian pada masing-masing dataset dilakukan sebanyak sepuluh kali menggunakan sepuluh gambar yang berbeda.

Dari Gambar 4.21 menunjukan pengujian klasifikasi penyakit kulit dengan implementasi *Extreme Value Theory* dengan dataset DERMNET. Dari gambar dapat dilihat sistem dapat mengenali

gambar penyakit kulit. dari 10 gambar dataset DERMNET yang di klasifikasikan, 7 gambar diklasifikasikan sebagai data 7 kelas penyakit kulit dan 3 gambar diklasifikasikan sebagai gambar *open-set*.



Gambar 4.21: Hasil pengujian proses klasifikasi dataset DERMNET dengan input gambar 7 kelas penyakit kulit

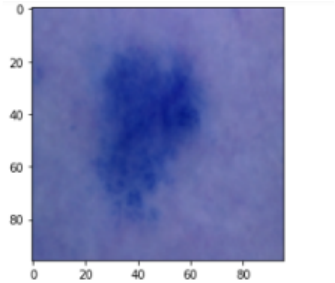
Dari Gambar 4.22 menunjukan pengujian klasifikasi penyakit kulit dengan implementasi *Extreme Value Theory* dengan dataset

STL-10. Dari gambar dapat dilihat sistem dapat menolak gambar *open-set*. Dari 10 gambar dataset STL-10 yang di klasifikasikan, 10 gambar diklasifikasikan sebagai gambar *open-set*.

Gambar 4.22: Hasil pengujian proses klasifikasi dataset STL-10 dengan input gambar *open-set*

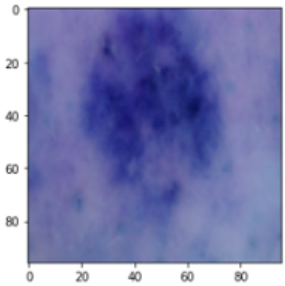
Dari Gambar 4.23 menunjukan pengujian klasifikasi penyakit

kulit dengan implementasi *Extreme Value Theory* dengan dataset HAM10000. Dari gambar dapat dilihat sistem dapat menolak gambar *open-set* penyakit kulit, namun masih ada beberapa gambar yang diketahui oleh sistem sebagai gambar 7 kelas penyakit kulit. Dari 10 gambar dataset HAM10000 yang di klasifikasikan, 2 gambar diklasifikasikan sebagai data 7 kelas penyakit kulit dan 8 gambar diklasifikasikan sebagai gambar *open-set*.




```
0->Unknown|1->frncl|2->dmta|3->disez|4->herp|5->psr|6->ring|7->acne
Prediction: Unknown

Unrevised Values:  [1 3 2 5 6]
Revised Top 5: [0 3 1 2 5]
```

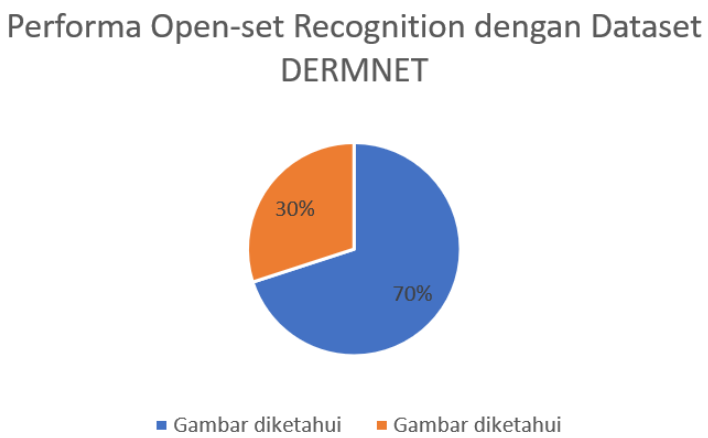



```
0->Unknown|1->frncl|2->dmta|3->disez|4->herp|5->psr|6->ring|7->acne
Prediction: Known
Unrevised Values:  [3 1 6 2 5]
Revised Top 5: [3 0 6 1 2]
```

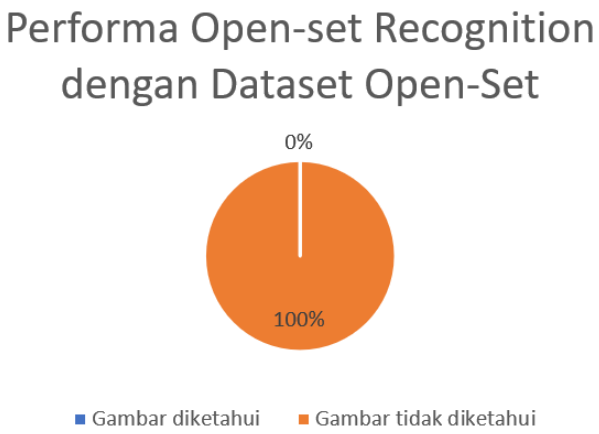
Gambar 4.23: Hasil pengujian proses klasifikasi dataset HAM10000 dengan input gambar *open-set* penyakit kulit

4.5 Analisa Perbandingan Hasil Pengujian Implementasi *Extreme Value Theory* (*Open-set Recognition*)

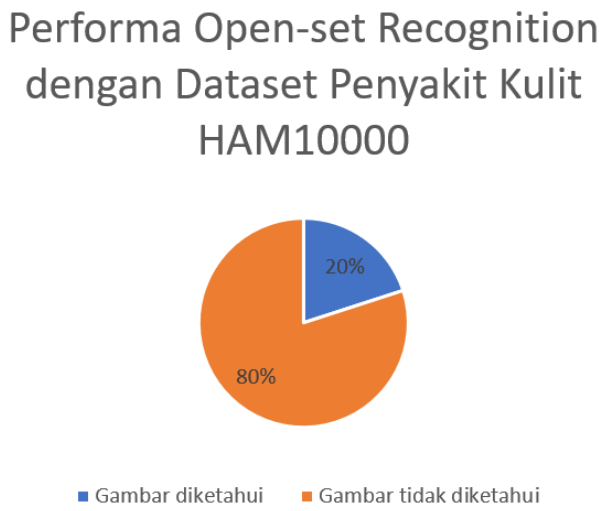
Setelah dilakukan uji Implementasi *Extreme Value Theory* (*Open-set Recognition*) pada ketiga dataset DERMNET, dataset STL-10, dan dataset HAM10000, didapatkan nilai yang dapat dilihat pada Gambar 4.24, gambar 4.25, dan gambar 4.25. Dari hasil klasifikasi dengan *open-set recognition* dapat dilihat jika sistem sudah dapat membedakan antara data yang merupakan *closed-set* dengan data *open-set* dimana pada pengujian dengan dataset DERMNET sudah dapat mengenali dengan benar 7 dari 10 gambar dataset. Pada dataset STL-10 sudah dapat sepenuhnya mengenali gambar *open-set* yaitu mengenali 10 dari 10 gambar dataset. Dan pada dataset HAM1000 sudah dapat mengenali 8 dari 10 data *open-set* penyakit kulit. Sehingga dapat disimpulkan bahwa hasil Implementasi *Extreme Value Theory* pada sistem klasifikasi penyakit kulit sudah dapat mengenali antara gambar *closed-set* dengan gambar *open-set* namun masih ada beberapa gambar yang tidak dikenali dengan baik sehingga data yang benar diklasifikasikan sebagai data *open-set* dan data yang salah diklasifikasikan sebagai data *closed-set*.



Gambar 4.24: Grafik performa *Open-set Recognition* pada dataset DERMNET



Gambar 4.25: Grafik performa *Open-set Recognition* dengan dataset STL-10



Gambar 4.26: Grafik performa *Open-set Recognition* dengan dataset HAM10000

Halaman ini sengaja dikosongkan

BAB 5
PENUTUP

5.1 Kesimpulan

Pada penelitian kali ini, Deteksi *Out of Distribution* (OOD) menggunakan *Extreme Value Theory* (EVT) pada Klasifikasi Penyakit Kulit dengan model CNN *MobileNet* v1 dan *Inception* V3 mendapatkan kesimpulan sebagai berikut :

1. Dari hasil pengujian implementasi *Extreme Value Theory* pada proses training dengan dataset Dermnet dan gambar Open-set. Sistem sudah dapat membedakan data open-set dan data penyakit kulit(*closed-set*) dengan mengenali dengan benar 7 dari 10 gambar dataset DERMNET, 10 dari 10 gambar dataset STL-10 dan 8 dari 10 dataset HAM10000.
2. Berdasarkan hasil pengujian, model yang memiliki performansi prediksi optimal ada pada penyakit Acne (acne) dengan data citra yaitu 207 citra dan menjadi kelas penyakit kulit yang memiliki persentase prediksi benar paling tinggi yaitu 90% pada model *Inception* V3 dan 70% pada model *MobileNet* V1.
3. Berdasarkan hasil pengujian, model yang memiliki performansi prediksi rendah dipengaruhi oleh sedikitnya data train asli. Furuncle (frncl) memiliki data citra paling sedikit yaitu 50 citra dan menjadi kelas penyakit kulit yang memiliki persentase prediksi salah paling tinggi yaitu 43% pada model *Inception* V3 dan 57% pada model *MobileNet* V1.

5.2 Saran

Demi pengembangan lebih lanjut mengenai tugas akhir ini, terdapat beberapa saran sebagai langkah lanjutan yaitu sebagai berikut :

1. Memperbanyak *dataset* citra penyakit kulit. Karena pada tugas akhir ini, *dataset* citra penyakit kulit sangat terbatas, sehingga nilai akurasi dari hasil training tidak cukup tinggi. Permasalahan yang dihadapi oleh penulis adalah adanya pandemi yang mengakibatkan tidak dapat mendapat data dari Rumah Sakit.

Halaman ini sengaja dikosongkan

DAFTAR PUSTAKA

[1] S. Greengard, "What is artificial intelligence?." https://www.datamation.com/artificial-intelligence/what-is-artificial-intelligence.html, 2019. Diakses: 2020-07-21. (Dikutip pada halaman xi, 5).

[2] "Cognitive computing and machine learning." http://www.cognub.com/index.php/cognitive-platform/, 2017. Diakses: 2020-07-21. (Dikutip pada halaman xi, 7).

[3] Mattab, "Artificial intelligence, enough of the hype! what is it?." https://community.hpe.com/t5/hpe-blog-uk-ireland/artificial-intelligence-enough-of-the-hype-what-is-it/ba-p/7046672#, 2019. Diakses: 2020-07-21. (Dikutip pada halaman xi, 10).

[4] A. Deshpande, "A beginner's guide to understanding convolutional neural networks." https://adeshpande3.github.io/adeshpande3.github.io/A-Beginner's-Guide-To-Understanding-Convolutional-Neural-Networks/, 2016. Diakses: 2020-07-21. (Dikutip pada halaman xi, 11).

[5] J. P. J. K. Nameer Hirschkind, Saruque Mollick, "Convolutional neural network." https://brilliant.org/wiki/convolutional-neural-network/, 2019. Diakses: 2020-07-21. (Dikutip pada halaman xi, 12, 13).

[6] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, "Rethinking the inception architecture for computer vision," in 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), IEEE, June 2016. (Dikutip pada halaman xi, 14).

[7] A. Milton-Barker, "Inception v3 deep convolutional architecture for classifying acute myeloid/lymphoblastic leukemia." https://software.intel.com/content/www/us/en/develop/articles/inception-v3-deep-convolutional-architecture-for-classifying-

acute-myeloidlymphoblastic.html, 2019. Diakses: 2020-07-21. (Dikutip pada halaman xi, 15).

[8] I. Mansouri, "Computer vision part 4: An overview of image classification architectures." https://medium.com/@ilias_mansouri/part-4-image-classification-9a8bc9310891, 2019. Diakses: 2020-07-21. (Dikutip pada halaman xi, 15).

[9] J. chaturvedi, "Confusion matrix in machine learning." https://medium.com/@jaiminichaturvedi_44311/confusion-matrix-in-machine-learning-93de6f29eb32, 2019. Diakses: 2020-07-21. (Dikutip pada halaman xi, 16).

[10] T. Team, "What's coming in tensorflow 2.0." https://blog.tensorflow.org/2019/01/whats-coming-in-tensorflow-2-0.html, 2019. Diakses: 2020-07-21. (Dikutip pada halaman xi, 18).

[11] "Extreme value theory: A better way to analyse environmental data." https://www.lancaster.ac.uk/~blackb/extremes.html, 2016. Diakses: 2020-07-21. (Dikutip pada halaman xi, 19).

[12] W. J. Scheirer, "Open set recognition." https://www.wjscheirer.com/projects/openset-recognition/, 2011. Diakses: 2020-07-21. (Dikutip pada halaman xi, 20).

[13] "Dermnet : dermatology pictures skin disease pictures." http://www.dermnet.com/dermatology-pictures-skin-disease-pictures. Diakses: 10 April 2020. (Dikutip pada halaman xi, 24).

[14] K. Matthew Mayo, "Dataset splitting best practices in python." https://www.kdnuggets.com/2020/05/dataset-splitting-best-practices-python.html, 2020. Diakses: 2020-07-21. (Dikutip pada halaman xi, 27).

[15] S. Sossi Alaoui, Y. Farhaoui, and B. Aksasse, "Classification algorithms in data mining," International Journal of Tomography and Simulation, vol. 31, pp. 34–44, 08 2018. (Dikutip pada halaman xi, 32).

[16] R. Departemen Kesehatan, "Profil kesehatan indonesia tahun 2015," 2016. (Dikutip pada halaman 1).

[17] R. Kementrian Kesehatan, "Data dan informasi profil kesehatan indonesia 2017," 2018. (Dikutip pada halaman 1).

[18] K. Hung and Y.-T. Zhang, "Implementation of a wap-based telemedicine system for patient monitoring," vol. 7, no. 2, pp. 101–107, 2003. (Dikutip pada halaman 1).

[19] A. K. Hernanda, Klasifikasi Penyakit pada Citra Dermoscopic Kulit menggunakan Convolutional Neural Network (CNN). Teknik Komputer ITS, 2019. (Dikutip pada halaman 2).

[20] M. Alsedrah, "Artificial intelligence," 12 2017. (Dikutip pada halaman 5).

[21] D. Sharma and N. Kumar, "A review on machine learning algorithms, tasks and applications," International Journal of Advanced Research in Computer Engineering & Technology (IJARCET), vol. 6, no. 10, 2017. (Dikutip pada halaman 6).

[22] A. Ahmad, "Mengenal artificial intelligence, machine learning, neural network, dan deep learning," no. October, 2017. (Dikutip pada halaman 6, 7, 9).

[23] J. DeMuro, "What is a neural network?." `https://www.techradar.com/news/what-is-a-neural-network`, 2018. Diakses: 2020-24-04. (Dikutip pada halaman 6).

[24] S. S. Haykin, S. S. Haykin, S. S. Haykin, K. Elektroingenieur, and S. S. Haykin, Neural networks and learning machines, vol. 3. Pearson education Upper Saddle River, 2009. (Dikutip pada halaman 7, 8).

[25] I. Goodfellow, Y. Bengio, and A. Courville, "Deep learning," 2016. (Dikutip pada halaman 10).

[26] S. R. Dewi et al., "Deep learning object detection pada video menggunakan tensorflow dan convolutional neural network," 2018. (Dikutip pada halaman 10).

[27] S. Sena, ""pengenalan deep learning part 7 : Convolutional neural network (cnn)." https://medium.com/@samuelsena/pengenalan-deep-learning-part-7-convolutional-neural-network-cnn-b003b477dc94, 2017. Diakses: 2020-24-04. (Dikutip pada halaman 13, 14, 15).

[28] A. G. Howard, M. Zhu, B. Chen, D. Kalenichenko, W. Wang, T. Weyand, M. Andreetto, and H. Adam, "Mobilenets: Efficient convolutional neural networks for mobile vision applications," CoRR, vol. abs/1704.04861, 2017. (Dikutip pada halaman 16).

[29] E. M. Rudd, L. P. Jain, W. J. Scheirer, and T. E. Boult, "The extreme value machine," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 40, no. 3, pp. 762–768, 2018. (Dikutip pada halaman 19, 20).

[30] W. J. Scheirer, A. Rocha, R. Michaels, and T. E. Boult, "Meta-recognition: The theory and practice of recognition score analysis," IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI), vol. 33, p. 1689–1695, 2011. (Dikutip pada halaman 34).

[31] A. Bendal and T. E. Boult, "Towards open set deep networks," 2016. (Dikutip pada halaman 34).

[32] A. Coates, A. Ng, and H. Lee, "An Analysis of Single Layer Networks in Unsupervised Feature Learning," in AISTATS, 2011. https://cs.stanford.edu/~acoates/papers/coatesleeng_aistats_2011.pdf. (Dikutip pada halaman 56).

[33] P. Tschandl, C. Rosendahl, and H. Kittler, "The ham10000 dataset, a large collection of multi-source dermatoscopic images of common pigmented skin lesions," 2018. (Dikutip pada halaman 56).

Halaman ini sengaja dikosongkan

BIOGRAFI PENULIS

Yordan Yasin, lahir pada 22 Juni 1998 di Kota Bandar Lampung, Provinsi Lampung. Penulis merupakan anak ketiga dari tiga bersaudara. Penulis menyelesaikan pendidikan dasar pada SDN Ciputat 1 pada tahun 2010. Pada tahun 2013, penulis menyelesaikan tingkat menengah pada SMP Negeri 3 Tangerang Selatan. Penulis menyelesaikan tingkat atas pada SMA Negeri 1 Tangerang Selatan pada tahun 2016. Penulis kemudian melanjutkan pendidikan Strata Satu pada Departemen Teknik Komputer Fakultas Teknologi Elektro dan Informatika Cerdas Institut Teknologi Sepuluh Nopember Surabaya. Selama masa kuliah, penulis aktif dalam beberapa kepanitiaan maupun organisasi. Pada tahun 2018, penulis mendapatkan amanah sebagai wakil ketua MAGE (*Multimedia And Game Event*) yang diselenggarakan oleh Departemen Teknik Komputer FTE ITS. Di tahun yang sama, penulis mendapatkan amanah sebagai Kepala Bidang Keprofesian departemen PROKESMA HIMATEKTRO. Selain itu, penulis juga aktif menjadi asisten laboraturium B401 Komputasi Multimedia dan pernah menjabat sebagai koordinator dalam bidang *Internet of Things Development*. Penulis memiliki hobby *Gaming* dan *Touring*. Untuk kepentingan pada penulis terkait kritik, saran, atau pertanyaan mengenai tugas akhir ini dapat menghubungi melalui *E-Mail* : yordanyasin9@gmail.com

Halaman ini sengaja dikosongkan