

BAB 2

TINJAUAN PUSTAKA

2.1 Penelitian Terdahulu

Keshavamurthy et al. (2024), membangun algoritma *machine learning*, khususnya membangun model *Logistic Regression* (LR) dan *Extreme Gradient Boosting* (XGB) untuk memprediksi kemungkinan hewan memiliki rabies. Penelitian ini menggunakan teknik *Random Oversampling* (ROS) dan *Synthetic Minority Oversampling Technique* (SMOTE) untuk menyeimbangkan data hewan yang memiliki rabies dan data hewan yang tidak memiliki rabies. Data hewan yang digunakan berasal dari Haiti IBCM, yang memiliki fitur-fitur tabular yang menentukan apakah hewan tersebut memiliki rabies atau tidak. Model yang dibangun akan memprediksi seberapa besar kemungkinan sebuah hewan atau kasus hewan berupa kasus rabies, di mulai dari rendah, sedang, dan tinggi. Hasilnya, ditemukan bahwa XGB-ROS mampu mengklasifikasikan kasus terkonfirmasi rabies dengan 85.2% diberi kemungkinan ‘tinggi’, atau sebesar 8.4% diberi kemungkinan ‘sedang’.

Boteju et al. (2020), membangun algoritma *deep learning* untuk membangun sistem untuk menganalisa perilaku serta ras anjing berdasarkan video dari kamera CCTV dan sensor. Penelitian ini juga menggunakan data audio yang juga diperoleh dari kamera CCTV untuk mengidentifikasi perilaku bersuara dari anjing. Algoritma yang dibangun menggunakan *transfer learning* menggunakan ResNet50, Inception V3, dan *support vector machine* (SVM) untuk melihat dan mengklasifikasi aktivitas dari anjing. Model penelitian yang dibangun menemukan bahwa modelnya mampu memprediksi ras dengan akurasi 89%, memprediksi pola berjalan sebesar 99.5%, perilaku beristirahat sebesar 97%, dan perilaku bersuara dengan 60%.

Wang et al. (2023), membangun algoritma *deep learning* untuk dapat mengklasifikasi berbagai jenis perilaku dari hewan untuk dapat memonitori kesehatan hewan. Penelitian ini menggunakan data video hasil sensor IOT dari hewan beruang kutub. Data gambar tersebut dianalisis menggunakan model deteksi objek berupa YOLOv5 untuk mengklasifikasi berbagai jenis perilaku hewan secara berkelanjutan dan efisien. Model yang telah dibangun mampu mendeteksi dengan

akurasi 98.3% akurasi presisi, dan mampu mengklasifikasi perilaku dengan akurasi 89.5%.

Bohnslav et al. (2021), merancang sistem dengan nama DeepEthogram yang dapat secara otomatis mengklasifikasi setiap *frame* dari video menjadi *ethogram* atau kumpulan perilaku dari hewan. Sistem ini dibangun menggunakan model *supervised deep-learning*, berupa *Convolutional Neural Network* (CNN) yang menggunakan *Hidden Two-Stream Network* dan *Temporal Gaussian Mixture* (TGM) *networks* untuk mendeteksi aksi di dalam sebuah video, yang kemudian diadaptasikan pada perilaku hewan tikus dan lalat. Hasil dari model yang dibangun mampu dengan akurat mengklasifikasi tiap *frame* dari video tikus dan lalat dengan akurasi diatas 90%.

Chambers et al. (2021), merancang algoritma untuk mendeteksi perilaku-perilaku dari anjing berdasarkan data akselerometer terletak di kalung anjing. Data akselerometer tersebut akan dianalisa menggunakan *machine learning*, yang kemudian divalidasi menggunakan data video perilaku anjing sebagai salah satu label datanya. Perilaku utama yang akan diklasifikasikan adalah makan dan minum namun, perilaku lain seperti menjilat, menggosok, mencakar, mencium, dan mengelus. Hasil dari algoritma tersebut mampu dengan akurat mengklasifikasi perilaku makan dan minum dengan 95% akurasi, namun juga dapat dengan akurat memprediksi perilaku menjilat, menggosok, mencakar, mencium, dan mengelus.

Tabel 2.1 Perbandingan Studi Literatur

Penulis	Judul	Metode	Hasil	Studi Kasus
Keshavamurthy dkk., 2024	<i>Machine learning to improve the understanding of rabies epidemiology in low surveillance settings</i>	<i>Logistic Regression (LR), Extreme Gradient Boosting (XGBoost), Random Oversampling (ROS), Synthetic Minority Oversampling Technique (SMOTE)</i>	Hasilnya, ditemukan bahwa XGB-ROS mampu mengklasifikasikan kasus terkonfirmasi rabies dengan 85.2% diberi kemungkinan ‘tinggi’, atau sebesar 8.4% diberi kemungkinan ‘sedang’	Kasus Rabies pada hewan di Haiti

Boteju dkk., 2020	<i>Deep Learning Based Dog Behavioural Monitoring System</i>	<i>Transfer Learning Resnet50 dan InceptionV3, Support Vector Machine (SVM)</i>	Modelnya mampu memprediksi ras dengan akurasi 89%, memprediksi pola berjalan sebesar 99.5%, perilaku beristirahat sebesar 97%, dan perilaku bersuara dengan akurasi 60%.	Perilaku dan ras anjing
Wang dkk., 2023	<i>Deep-learning-based multi-behavior classification of animals for efficient health and welfare monitoring</i>	YOLOv5	Model yang telah dibangun mampu mendeteksi dengan akurasi 98.3% akurasi presisi, dan mampu mengklasifikasi perilaku dengan akurasi 89.5%.	Perilaku beruang kutub
Bohnslav dkk., 2020	<i>DeepEthogram, a machine learning pipeline for supervised behavior classification from raw pixels</i>	<i>Convolutional Neural Network (CNN), Hidden Two-Stream Method, Temporal Gaussian Mixture (TGM)</i>	Hasil dari model yang dibangun mampu dengan akurat mengklasifikasi tiap <i>frame</i> dari video tikus dan lalat dengan akurasi diatas 90%.	Perilaku tikus dan lalat
Chambers dkk., 2021	<i>Deep learning classification of canine behavior using a single Collar-Mounted accelerometer</i>	<i>Convolutional Neural Network (CNN)</i>	Hasil dari algoritma tersebut mampu dengan akurat mengklasifikasi perilaku makan dan minum dengan 95% akurasi, namun juga dapat dengan akurat memprediksi perilaku menjilat, menggosok, mencakar, mencium, dan mengelus.	Perilaku anjing

Fernando, B. 2025	Pengembangan sistem klasifikasi perilaku <i>abnormal</i> hewan peliharaan	<i>Convolutional Neural Network</i> (CNN), <i>Generative Adversarial Network</i> (GAN)	Hasil dari model yang dibangun mampu memperoleh akurasi setinggi 80% pada data uji.	Perilaku <i>abnormal</i> anjing dan kucing
-------------------	---	--	---	--

Dari riset yang telah dilakukan berdasarkan penelitian terdahulu di Tabel 2.1, maka penelitian yang akan dilakukan pada proyek akhir ini adalah membangun sistem klasifikasi perilaku abnormal pada hewan berdasarkan gambar dan video hewan menggunakan algoritma *deep learning* berupa CNN, dan *Transfer Learning*. Sistem ini akan dapat mendeteksi perilaku abnormal pada hewan berdasarkan gambar ataupun video yang diambil oleh kamera pengguna. Model yang digunakan adalah model CNN ataupun model *Transfer Learning* yang cocok dalam melakukan masalah klasifikasi menggunakan data gambar. Dikarenakan sedikitnya data gambar dan video hewan tersebut, maka digunakanlah metode GAN atau *Generative Adversarial Network* untuk dapat menghasilkan set data pelatihan yang baru yang dapat digunakan.

2.2 Landasan Teori

Landasan teori yang digunakan sebagai bahan pendukung penelitian ini adalah sebagai berikut:

2.2.1 Perilaku Abnormal Hewan

Perilaku *abnormal* bisa didefinisikan sebagai suatu perilaku yang tidak sesuai dengan pola, kebiasaan, atau dengan konteks saat perilaku tersebut terjadi ketika dibandingkan dengan mayoritas spesiesnya (Broom, 2019). Perilaku ini umumnya muncul ketika kebutuhan dasar hewan, seperti kebutuhan fisiologis, sosial, maupun lingkungan, tidak terpenuhi secara optimal. Ketidaksesuaian antara kondisi lingkungan dengan kebutuhan hewan dapat memicu munculnya respons perilaku yang tidak adaptif sebagai bentuk mekanisme *coping* terhadap tekanan yang dialami.

Perilaku *abnormal* ini bisa bervariasi dari spesies ke-spesies dan seberapa berat penyebab perilaku tersebut. Selain stres, faktor lain yang dapat mempengaruhi perilaku *abnormal* ini bisa dikarenakan genetik, pengalaman, serta trauma dari hewan tersebut. Perilaku abnormal juga bisa diakibatkan penyakit seperti akibat

virus, bakteri ataupun parasit (Abnormal Behaviour Executive Summary, n.d.).

Secara umum, bentuk perilaku abnormal pada hewan dapat diklasifikasikan menjadi beberapa kategori, antara lain perilaku repetitif atau stereotipik (seperti *pacing* dan menjilati secara berlebihan), perilaku agresif yang tidak sesuai konteks, perilaku kecemasan berlebihan, serta perilaku menyakiti diri sendiri. Perilaku-perilaku tersebut sering kali menjadi indikator awal adanya gangguan kesejahteraan atau masalah kesehatan mental pada hewan, khususnya pada hewan peliharaan seperti anjing dan kucing, sehingga deteksi dan penanganan dini sangat diperlukan untuk mencegah dampak yang lebih serius (McMillan, 2020).

2.2.2 Artificial Intelligence

Artificial Intelligence (AI) atau kecerdasan buatan dapat didefinisikan sebagai sebuah mesin atau komputer yang memiliki atau menunjukkan kemampuan kognitif, mirip dengan manusia. Kemampuan ini bisa berupa kemampuan memecahkan masalah ataupun kemampuan untuk belajar (Paaß & Hecker, 2024). Untuk menyatakan sebuah sistem komputer sebagai ‘cerdas’, maka seorang ilmuwan matematika bernama Alan Turing mengemukakan *the Turing test* di tahun 1950. Tes ini dilakukan dengan seorang penguji berinteraksi dengan sebuah manusia dan komputer, tanpa mengetahui identitas mereka. Jika si penguji tidak bisa mengidentifikasi siapa komputernya, maka komputer tersebut dapat dikatakan sebagai ‘cerdas’.



Gambar 2.1 Alan Turing

Tes ini merupakan asal mula sejarah AI yang dimulai sejak tahun 1950 dimana Alan Turing di Gambar 2.1 (Jackson, 2024), mengembangkan sebuah metode untuk mengevaluasi kecerdasan sebuah sistem komputer, namun istilah AI baru dikemukakan pada tahun 1956 dimana John McCarthy dan Marvin Minsky

pertama kali mendefinisikan AI. Ini-pun menyebabkan perkembangan pada sebuah sistem pakar DENDRAL pada 1965 yang dapat memecahkan sebuah masalah berbasis aturan-aturan yang telah ditetapkan pada masalah kimia organik.

Namun, pada 1969, tidak lama dari pembuatan sistem pakar tersebut, ditemukan bahwa sebuah *single-layer perceptron* tidak mampu menyelesaikan masalah yang kompleks, namun ini kemudian dipecahkan setelah ditunjukkan bahwa *multilayer neural network* dapat merepresentasikan sebuah hubungan yang kompleks pada masalah. Di tahun 1986, David Rumelhart, Geoffrey Hinton, dan Ronald Williams menemukan algoritma *backpropagation* yang dapat digunakan untuk meningkatkan akurasi ketika melakukan pelatihan, dan di tahun 1997, Sepp Hochreiter dan Jürgen Schmidhuber menemukan potensi menggunakan *Long Short-Term Memory* untuk melakukan pemodelan sebuah data berupa teks, dan suara. Dan akhirnya, pada tahun 2015, *Deep Neural Network* dengan 152 lapisan ditemukan oleh Kaiming He, et al, untuk melakukan pendeteksian gambar lebih baik daripada manusia.

2.2.3 Machine Learning

Machine Learning atau pembelajaran mesin adalah teknik untuk meningkatkan performa sebuah sistem dengan mempelajari dari pengalaman menggunakan suatu metode komputasi. Pengalaman disini mengacu pada data yang dimiliki, dan tugas utama dari pembelajaran mesin adalah untuk membangun sebuah model dari data, dan model disini dapat disimpulkan sebagai hasil dari sebuah pembelajaran mesin dari data (Zhou, 2021).

Vakalopoulou et al. (2023) mendefinisikan *Machine Learning* sebagai salah satu cabang ilmu pengetahuan yang bertujuan untuk memberikan sebuah komputer kemampuan untuk melaksanakan sebuah tugas tanpa harus diprogram secara langsung. Untuk mencapai tujuan tersebut, maka komputer tersebut harus dilatih berdasarkan contoh ataupun pengampunan.

Machine Learning dapat dinyatakan menjadi empat jenis yaitu *supervised learning*, *unsupervised learning*, *semi-supervised learning*, dan *reinforcement learning*.

1) *Supervised Learning*

Supervised Learning dapat didefinisikan sebagai bentuk pembelajaran mesin dimana data *training* telah diberikan sebuah label atau target *output* yang diinginkan untuk mengurangi terjadinya misklasifikasi atau error antara target sebenarnya dengan target dari prediksi.

2) *Unsupervised Learning*

Unsupervised Learning dapat didefinisikan sebagai bentuk pembelajaran mesin dimana data *training* belum diberi label atau target, dan model tersebut akan mengelompokkan sekumpulan data tersebut dalam bentuk *cluster* berdasarkan seberapa mirip suatu data dengan data lainnya.

3) *Semi-supervised Learning*

Semi-supervised Learning merupakan bentuk gabungan dari *supervised* dan *unsupervised learning* dimana, dalam pembelajarannya digunakan data yang telah diberi label dan data yang belum diberi label. Dalam kasus pembelajaran ini, data yang belum diberi label akan dilatih dengan algoritma *unsupervised learning* dimana hasilnya akan merujuk pada data yang telah diberi label. Kemudian, algoritma *supervised learning* akan dilatih dengan data berlabel dan data *cluster* yang belum di label tersebut.

4) *Reinforcement Learning*

Reinforcement Learning dapat didefinisikan sebagai bentuk pembelajaran mesin yang belajar dari lingkungan, dan melakukan suatu aksi berdasarkan *reward* dan *penalty*.

2.2.4 *Deep Learning*

Menurut Ketkar & Moolayil (2021), *Deep Learning* bisa didefinisikan sebagai salah satu kelas atau cabang dari *machine learning* yang menggunakan algoritma yang memiliki struktur mirip dengan otak manusia untuk melaksanakan berbagai tugas di era modern saat ini. *Deep learning* ini bedanya dengan *machine learning* adalah mereka mampu melakukan *feature extraction* yang berarti mempelajari fitur-fitur dari sebuah data tanpa harus ditentukan oleh manusia. Itulah kenapa kata *deep* disini memiliki makna bahwa banyaknya lapisan-lapisan informasi dari sebuah model sehingga memungkinkan bahwa sebuah model untuk mempelajari data dengan tingkat abstraksi yang tinggi.

Goodfellow et al. (2016), mendefinisikan *Deep Learning* sebagai salah satu bentuk pendekatan dari *machine learning* yang memungkinkan sistem komputer untuk berkembang dengan pengalaman dan data dengan cara merepresentasikan sebuah dunia atau data sebagai sekumpulan hirarki atau lapisan-lapisan konsep, dengan tiap konsep tersebut didefinisikan menjadi sebuah konsep yang lebih sederhana dan representasi abstrak direpresentasikan menjadi lebih tidak abstrak, atau dengan kata lain, menyederhanakan sebuah konsep dan abstrak menjadi lebih sederhana.

2.2.5 Convolutional Neural Network

Convolutional Neural Network (CNN) adalah bagian atau salah satu bentuk dari *deep learning* yang menggunakan operasi konvolusi untuk memproses data input. CNN ini cocok untuk memproses data berdimensi atau terorganisasi dalam bentuk *grid*, contohnya: data berbasis waktu, gambar, dan video (Vakalopoulou et al., 2023). Operasi konvolusi ini merupakan operasi yang dilakukan pada dua buah fungsi untuk memperoleh fungsi ketiga, fungsi pertama disini berperan sebagai Input, lalu fungsi kedua berperan sebagai *Kernel*, dan outputnya dikenal sebagai peta fitur (Goodfellow et al., 2016).

$$s(t) = \int x(a)w(t - a)da \quad (2.1)$$

Dimana:

a : *Dummy variable*

$x(a)$: *Input*

$w(t)$: *Kernel/Filter*

$s(t)$: *Feature Map/Output*

Konvolusi dua buah fungsi dapat dirujuk pada persamaan (2.1) seperti diatas dimana terdapat dua fungsi berupa *Input* dan *Kernel/filter*, dan menggunakan persamaan konvolusi, kita dapat mendapatkan fungsi baru yang serupa dengan fungsi Input, tetapi telah diubah atau di *filter* berdasarkan *Kernel*.

$$s(a,b) = \sum_m \sum_n I(m,n)K(a - m, b - n) \quad (2.2)$$

Dimana:

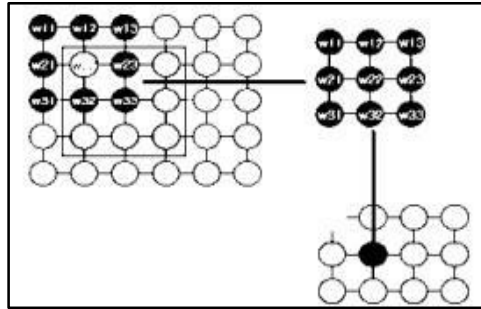
$I(m, n)$: *Input*
 $K(a - m, b - n)$: *Kernel/Filter*
 $s(a, b)$: *Feature Map/Output*

Dalam penerapan *machine learning*, data input nya itu umumnya berupa sebuah *array* atau struktur data yang memiliki banyak dimensi. Bukan hanya itu, kernel juga umumnya berupa array dari parameter yang memiliki banyak dimensi yang diadaptasikan oleh algoritmanya. Bukan hanya itu, data-data yang digunakan akan bersifat diskrit, bukan kontinu, yang berarti data tersebut memiliki informasi yang dapat dihitung dan bersifat terbatas. Untuk mengakomodasi data yang bersifat diskrit dan memiliki sejumlah dimensi, maka digunakan operasi seperti di persamaan (2.2) dengan ketentuan inputan I memiliki dua dimensi (m, n) dan kernel K memiliki dua dimensi (m, n) juga.

Sebuah CNN pada umumnya memiliki empat buah komponen, yaitu *convolution layer*, *pooling layer*, *activation function*, dan *fully connected layer* (Indolia et al., 2018)

1) *Convolution Layer*

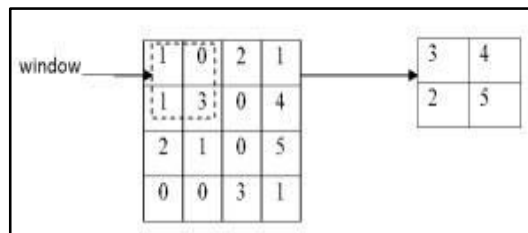
Convolution layer merupakan lapisan dimana operasi konvolusi ini digunakan. Pada lapisan ini, suatu data seperti data gambar akan dilakukan proses prediksi berdasarkan fitur dari gambar tersebut. *Neuron-neuron* yang terkoneksi dari lapisan ke lapisan akan membentuk yang dinamakan *receptive field*, yang akan memperoleh fitur lokal dari sebuah gambar. *Receptive field* dari sebuah neuron pada daerah tertentu di beberapa lapisan akan membentuk yang dinamakan *weight vector*, atau juga dikenal sebagai *filter*, atau *kernel*. *Kernel* ini akan bergeser-geser pada vektor input untuk membuat peta fitur, yang telah dijelaskan pada bagian operasi konvolusi sebelumnya.



Gambar 2.2 Contoh operasi konvolusi di *convolution layer*

Dari Gambar 2.2 (Indolia et al., 2018), diatas, dapat dilihat inputan gambar di bagian kiri-atas memiliki *kernel* 3x3 yang kemudian, akan bergeser dari dari piksel-piksel pada gambar untuk melakukan dilakukan operasi. Hasilnya adalah sebuah peta fitur yang dapat dilihat di bagian kanan bawah gambar.

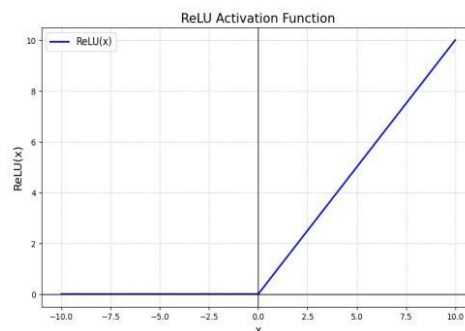
2) *Pooling Layer*



Gambar 2.3 Contoh operasi *pooling* di *convolution layer*

Pooling layer merupakan lapisan yang umum dilakukan setelah *convolution layer*, dimana hasil dari *convolution layer* tersebut akan kemudian diproses dengan mereduksi peta fitur menjadi lebih kecil menggunakan *pooling window* sebagai bentuk filter pada peta fitur tersebut seperti yang dapat di lihat di Gambar 2.3 (Indolia et al., 2018) diatas.

3) *Activation Function*



Gambar 2.4 *Activation Function* ReLU

Agar sebuah model bisa belajar dari data secara non-linear, terutama ketika data yang dimiliki tidak linear seperti gambar atau video. Maka *Rectified Linear Unit* (ReLU) adalah salah satu yang terbukti bagus yang digambarkan di Gambar 2.4 (*ReLU Activation Function in Deep Learning - GeeksforGeeks*, n.d.). Ini dikarenakan beberapa faktor, pertama, fungsi pada ReLU lebih mudah dihitung oleh mesin. Kedua, ketika mempertimbangkan waktu dalam melatih model, ReLU termasuk yang cepat, dan ketiga, ReLU tidak akan membiarkan gradien pada fungsinya untuk menghilang.

4) *Fully Connected Layer*

Fully connected layer pada CNN mirip pada *fully connected layer* pada model umumnya. Disini, output dari bagian *feature extraction* dari kumpulan *convolution* dan *pooling* layer, akan di kemudian menjadi input pada *fully connected layer* akan kemudian diproses dengan tambahan vektor *weight* akan menghasilkan output terakhir.

2.2.6 *Transfer Learning*

Pada *machine learning* tradisional, sebuah model akan memiliki data *training* dan *testing* dengan jumlah ukuran fitur yang sama dan distribusi data yang sama. Jika ada perbedaan distribusi antara data *training* dan *testing* maka performa dari modelnya akan berkurang. Dalam beberapa kasus, memperoleh data dengan jumlah ukuran fitur yang sama dan distribusi yang sama bisa sangat sulit ditemukan dan mahal, maka dikenalkan istilah *transfer learning* (Weiss et al., 2016).

Transfer Learning merupakan metode untuk meningkatkan kemampuan pembelajaran mesin pada satu bidang dengan mengambil atau *transferring* informasi relevan dari bidang lain. Contoh aplikasi dari *transfer learning* ini adalah jika ingin membuat model prediksi sentimen teks pada suatu produk, dimana banyak sekali data *training* dan data *target* berupa ulasan kamera digital, maka metode *machine learning* cocok digunakan untuk prediksi. Namun, jika terdapat kasus dimana data *training* berupa ulasan kamera digital dan data *target* berupa ulasan makanan, maka model prediksi yang dibangun tidak akan memberikan hasil prediksi yang bagus. Namun, ulasan kamera digital dengan ulasan makanan memiliki beberapa kemiripan, dimana kedua data tersebut berupa teks, dengan bahasa yang sama, dan kedua data mengungkapkan sentimen terhadap produk.

2.2.7 Data Augmentation

Data Augmentation bisa didefinisikan sebagai teknik yang dapat menghasilkan suatu data baru dengan bentuk atau orientasi yang bervariasi (Maharana et al., 2022). Dengan *Data Augmentation*, dapat dipecahkannya dua masalah, yaitu menghasilkan beberapa data baru dari jumlah data yang sedikit, dan meminimalisir *overfitting*. *Data augmentation* tersebut memiliki beberapa teknik yang dapat digunakan berdasarkan konteks dan jenis datanya.

1) *Symbolic augmentation*

Symbolic augmentation merupakan bentuk augmentasi dimana sebuah data baru diproduksi dimana data tersebut dapat dipahami dan diinterpretasikan oleh manusia. Contoh dari interpretasikan oleh manusia. Contoh dari *symbolic augmentation* ini adalah dimana sebuah data baru teks berupa kalimat akan diubah dengan mengubah salah satu kata atau frase untuk menghasilkan data kalimat baru yang masih bisa dipahami manusia.

2) *Rule-based augmentation*

Rule-based augmentation merupakan bentuk augmentasi dimana data tersebut diproduksi melalui aturan-aturan yang telah ditentukan programmer. Penghapusan, penambahan, mengubah sinonim, pergantian kata merupakan contoh dari *rule-based augmentation*.

3) *Graph-structured augmentation*

Graph-structured augmentation akan membuat sebuah bentuk representasi teks data dengan struktur graf. Ini juga termasuk *metadata* seperti sitasi, encoding, struktur bahasa, dll.

4) *Mixup augmentation*

Mixup augmentation dilakukan dengan membuat data baru dengan mengkombinasikan data lama dengan labelnya. Contoh dari augmentasi ini bisa berupa menggabungkan setengah data teks kalimat dengan setengah data teks kalimat yang berbeda di sebuah *dataset*.

5) *Feature space augmentation*

Feature space augmentation menggunakan *noise* untuk membuat sebuah fitur pada data yang baru. *Noise* ini bisa dibuat menggunakan *adversarial controller* atau diambil dari dataset, ataupun *gaussian distribution*.

6) *Neural augmentation*

Neural augmentation menggunakan sebuah model lain untuk menghasilkan sebuah data baru.

Adapun teknik-teknik yang dilakukan untuk augmentasi sebuah data gambar yakni sebagai berikut:

1) *Flipping*

Flipping berarti membalikkan gambar, baik itu membalikkan gambar secara vertikal atau horizontal.

2) *Color Space*

Color space transformation merupakan teknik untuk mengubah warna pada gambar.

3) *Cropping*

Cropping berarti mengambil bagian kecil dari gambar dan membesarkannya sehingga ukurannya sesuai dengan dimensi awalnya. Jika diperlukan, *cropping* bisa dilakukan untuk mengubah dimensi gambar awal.

4) *Rotation*

Sebuah gambar dapat dilakukan rotasi baik secara 90 derajat atau pada sudut derajat yang bervariasi.

5) *Translation*

Translation atau translasi berarti menggerakkan objek pada gambarnya baik di dimensi x atau y.

6) *Noise Injection*

Noise injection merupakan teknik untuk menambahkan *noise* pada suatu gambar. Umumnya, sebuah gambar akan menggunakan *salt and pepper noise*.

2.2.8 *MixUp dan CutMix*

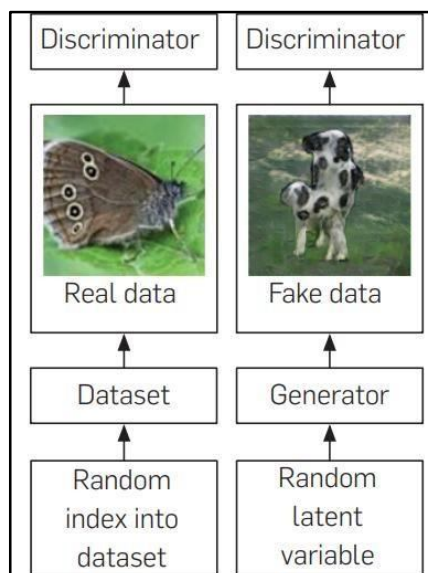
MixUp merupakan teknik augmentasi yang menghasilkan data baru dengan menggabungkan dua gambar beserta labelnya menggunakan kombinasi linear (Zhang et al., 2018). Pada metode ini, nilai piksel dari dua gambar dicampurkan dengan proporsi tertentu, sedangkan label baru dibentuk berdasarkan proporsi pencampuran tersebut. Teknik ini membantu model untuk membentuk batas keputusan yang lebih halus, mengurangi *overfitting*, serta meningkatkan kemampuan generalisasi model terhadap data yang belum pernah dilihat

sebelumnya.

CutMix merupakan pengembangan dari MixUp yang dilakukan dengan memotong sebagian area dari suatu gambar dan menempelkannya pada gambar lain (Yun et al., 2019). Label gambar hasil augmentasi kemudian disesuaikan berdasarkan proporsi luas area yang dipotong dan ditempelkan. Berbeda dengan MixUp yang mencampurkan seluruh piksel gambar, CutMix tetap mempertahankan informasi visual yang lebih jelas sehingga model dapat mempelajari fitur objek secara lebih efektif. Teknik ini terbukti mampu meningkatkan akurasi klasifikasi, kemampuan lokalisasi objek, serta ketahanan model terhadap gangguan pada gambar.

2.2.9 Generative Adversarial Network

Generative Adversarial Network (GAN) adalah bentuk model *machine learning* yang berdasarkan *game theory* atau teori permainan antara dua jenis *neural network* (Goodfellow et al., 2020). Satu jaringan dikenal sebagai *generator* dan satu jaringan lain dikenal sebagai *discriminator*. *Generator* berperan untuk belajar dari distribusi *dataset* dan mencoba menghasilkan suatu data baru yang realistis, sedangkan *Discriminator* akan belajar dari gabungan data asli dan data yang dihasilkan *Generator* untuk mengestimasi kemungkinan data tersebut adalah data asli atau data palsu.



Gambar 2.5 Ilustrasi Melatih Model GAN

Gambar 2.5 (Goodfellow et al., 2020) diatas mengilustrasikan bagaimana model GAN dilatih dimana model GAN itu sendiri terdiri atas melatih jaringan *generator* dan jaringan *discriminator*. Proses ini membutuhkan beberapa data asli dari sebuah *dataset*, dan *generator* akan menghasilkan data baru berdasarkan *dataset* tersebut. Jaringan *Discriminator* dilatih mirip dengan model klasifikasi lainnya, dimana jaringan tersebut akan dilatih menggunakan dua jenis label yaitu ‘asli’ atau ‘palsu’ menggunakan gabungan *dataset* asli dengan data baru dari *generator*. *Generator* dilatih untuk menipu *discriminator* atau dalam arti membuat *discriminator* memberikan label ‘asli’ pada data baru yang dibuat *generator*.

2.2.10 Aplikasi Mobile

Mobile dapat didefinisikan sebagai mudahnya melakukan perpindahan dari satu tempat ke tempat yang berbeda, dalam kasus ini yang melakukan perpindahan tersebut merupakan *smartphone* yang kita gunakan dan bawa dalam kegiatan sehari-hari. Sedangkan aplikasi dapat didefinisikan sebagai suatu program atau perangkat lunak dalam suatu sistem komputer yang dapat digunakan oleh orang-orang untuk membantu pekerjaannya. Maka, dari dua definisi tersebut dapat disimpulkan bahwa, aplikasi *mobile* adalah suatu perangkat lunak yang terletak dalam sebuah *smartphone* sehingga dapat diakses dan digunakan dimana saja, kapan saja (Voutama & Novalia, 2021)

2.2.11 Flutter

Flutter merupakan sebuah *framework* atau *Software Development Kit* (SDK) untuk pengembangan aplikasi mobile yang dapat digunakan lintas *platform* yang dikembangkan oleh google menggunakan bahasa pemrograman Dart (Windmill, 2020). Flutter memungkinkan kita untuk mengembangkan aplikasi mobile dengan mudah dan sederhana, yang dapat digunakan oleh Android dan iOS. Flutter pun juga dapat digunakan oleh web atau aplikasi *desktop*.

2.2.12 Confusion Matrix

Confusion matrix merupakan salah satu bentuk cara untuk mengevaluasi atau memvisualisasi kemampuan model dalam konteks model klasifikasi atau *supervised learning*. *Confusion matrix* berupa matriks persegi dimana sebuah baris merepresentasikan kelas, label atau target sebenarnya dari sebuah data, sedangkan

kolom merupakan kelas, label, atau target yang diprediksi oleh model (Caelen, 2017). Dalam kasus *binary classification*, maka bentuk dari matriks tersebut adalah 2 x 2 yang akan menunjukkan jumlah *True Positives* (TP), *True Negatives* (TN), *False Positive* (FP), dan *False Negatives* (FN) seperti digambarkan pada Tabel 2.2.

Tabel 2.2 *Confusion Matrix*

		Predicted Label	
		True	False
Actual Label	True	TP	FN
	False	FP	TN

Pada *confusion matrix*, sebuah model dapat dievaluasi dalam berbagai jenis *metrics* perhitungan untuk pertimbangan dan akurasi model (Sathyanarayanan, 2024).

1) *Accuracy*

Accuracy merupakan nilai perbandingan jumlah prediksi yang benar dengan jumlah semua prediksi yang dilakukan. Metrik ini cocok untuk mengevaluasi kemampuan model untuk memprediksi label dari suatu data secara umum berdasarkan total jumlah akurasinya. Namun, *metrics* ini kurang cocok jika *dataset* yang dimiliki tidak seimbang, dimana salah satu labelnya jauh mendominasi label lainnya. Dalam kasus ini, nilai *accuracy* tidak terlalu bagus dikarenakan suatu model bisa mendapatkan nilai *accuracy* cukup dengan memprediksi data mayoritas.

2) *Precision*

Precision merupakan nilai perbandingan dimana jumlah prediksi *True Positive* dibandingkan total jumlah prediksi *True Positive* dan *False Positive*. *Precision* ini bagus untuk melihat dari sekian prediksi pada label prediksi yang dinyatakan true, seberapa banyak prediksi tersebut dinyatakan benar. Namun, *Precision* tidak mempertimbangkan *False Negative*, yang merupakan data yang seharusnya diprediksi true, tetapi diprediksi sebagai negatif. *Metrics* ini kurang cocok jika kita ingin meminimilasi *False Negative* dikarenakan beberapa kasus, contohnya kasus prediksi penyakit kanker pada pasien, tidak ingin agar pasien yang memiliki kanker, diprediksi sebagai tidak memiliki kanker.

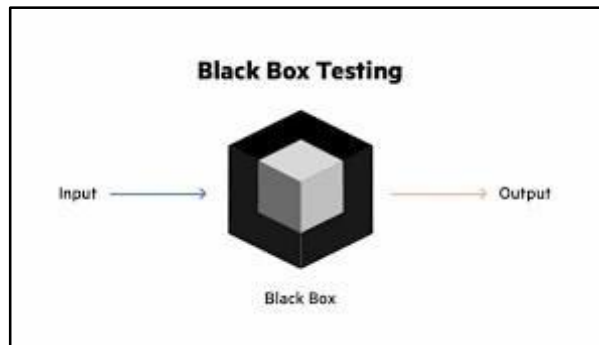
3) Recall

Recall merupakan nilai perbandingan seberapa akurat model memprediksi label true itu sendiri, bukan hanya *True Positive* saja. *Recall* cocok jika kita ingin memastikan model kita mampu memprediksi semua data dengan label true dengan benar. Jika *recall* kita tinggi, maka *false negative* kita dinyatakan rendah. *Recall* terkadang juga dikenal sebagai *sensitivity* atau *True Positive Rate*.

4) Specificity

Specificity bisa dikatakan sebagai kebalikan dari *recall* dimana *specificity* menghitung seberapa akurat model dalam memprediksi label negatif. *Specificity* juga dikenal sebagai *total negative rate* (TNR).

2.2.13 Black Box Testing



Gambar 2.6 Black Box Testing

Black Box Testing dapat didefinisikan sebagai pengujian yang dilakukan dengan mengamati hasil eksekusi dan kebutuhan fungsional dari perangkat lunak. Bentuk pengujian dapat dianalogikan seperti di Gambar 2.6 dimana kita melihat suatu kotak hitam, yang dimana kita hanya bisa melihat penampilan luarnya, bukan yang dibalik bungkus hitamnya. Sama disini, kita hanya bisa melihat penampilan dari perangkat lunak, serta kebutuhan fungsionalnya (Astuti, 2018).

Menurut Hartono & Sugiarti (2022) terdapat beberapa tahapan yang perlu dilaksanakan dalam black box testing yakni:

- 1) Merancang test case untuk menguji tiap kebutuhan fungsional aplikasi.
- 2) Merancang test case untuk menguji alur kerja dari fungsi sebuah sistem, apakah sudah sesuai dengan kebutuhan.
- 3) Menemukan bugs/error berdasarkan penampilan interface perangkat lunak.

2.2.14 System Usability Scale

System Usability Scale (SUS) merupakan kuesioner standar yang digunakan untuk menilai fungsionalitas dari sebuah sistem. SUS dalam bentuk paling umumnya memiliki 10 pertanyaan yang secara selang seling mengevaluasi aspek positif dan negatif dari pengalaman pengguna. SUS ini pertama kali dibentuk oleh John Brooke pada tahun 1996 dikarenakan adanya kebutuhan untuk mengaplikasikan faktor psikologis manusia dalam mengevaluasi sebuah sistem komputer baik di bidang perkantoran ataupun pribadi (Lewis, 2018).

$$SUS = 2.5 \left(20 + \sum (SUS01, SUS03, SUS05, SUS07, SUS09) - \sum (SUS02, SUS04, SUS06, SUS08, SUS10) \right) \quad (2.3)$$

Dalam menghitung nilai dari hasil evaluasi SUS, dapat dirujuk dengan persamaan (2.3). Langkah pertama yang diambil adalah untuk mengubah nilai awal menjadi nilai kontribusi yang berkisar dari 0-4. Untuk pertanyaan yang dikosongin atau berada ditengah, maka nilai awalnya adalah 3. Untuk pertanyaan ganjil, maka kurangi 1 dari nilai awal atau Nilai Awal – 1, dan untuk pertanyaan genap, kurangi nilai awal dari 5 atau 5 – Nilai awal. Setelah itu, jumlahkan semua nilai tersebut dan dikalikan 2,5 untuk memperoleh nilai SUS (Brooke & others, 1996).