

BAB II

TINJAUAN PUSTAKA

2.1 Penelitian Terkait

Penelitian pertama, Mustopa menggaris bawahi perlunya teknologi pendukung peternak dalam mengenali tanda-tanda penyakit ayam berdasarkan *feses*. Penelitian ini menggunakan dataset dari *Kaggle* yang mencakup 8.046 citra *feses* ayam dari empat kategori: sehat, *Coccidiosis*, *Salmonella*, dan *Newcastle*. Peneliti mengembangkan model CNN *custom* dan menguji berbagai arsitektur transfer *learning* seperti EfficientNetV2L, ResNet50, dan MobileNetV2. Melalui augmentasi dan resampling untuk mengatasi ketidakseimbangan data, model terbaik diperoleh dengan EfficientNetV2L + RMSprop yang menghasilkan akurasi 97%. Pendekatan ini menunjukkan potensi besar dalam penerapan diagnosis berbasis visual di dunia peternakan. [2]

Penelitian kedua oleh Machuve menyoroti tantangan deteksi dini penyakit unggas di Tanzania, khususnya di kalangan peternak kecil yang sulit mengakses layanan laboratorium. Penelitian ini fokus pada deteksi tiga penyakit utama: *Coccidiosis*, *Salmonella*, dan *Newcastle Disease* melalui citra *feses*. Sebanyak 8.067 citra dikumpulkan dan diberi label baik melalui uji PCR maupun oleh ahli *veteriner*. Model CNN yang digunakan meliputi VGG16, InceptionV3, MobileNetV2, dan Xception. Setelah proses fine-tuning, akurasi tertinggi dicapai oleh *Xception* (98,24%), dengan

MobileNetV2 (98,02%) sebagai model yang paling ringan dan layak diterapkan di lapangan. [3]

Penelitian oleh Likitha R membahas pengembangan sistem klasifikasi penyakit ayam berbasis *Convolutional Neural Network (CNN)* dengan tujuan membantu peternak dalam mendeteksi penyakit menular seperti *Newcastle disease*, *Pullorum*, dan *Coccidiosis* melalui analisis citra *feses*. Masalah utama yang ingin diatasi adalah kurangnya pemahaman peternak terhadap penyakit ayam dan mahalunya metode konvensional seperti PCR. Model CNN dikembangkan menggunakan *platform TensorFlow/Keras* dengan arsitektur sederhana. Dataset citra *feses* dikategorikan menjadi empat kelas: sehat, *Newcastle*, *Pullorum*, dan *Coccidiosis*. Melalui pelatihan dengan data tersebut, model berhasil mencapai akurasi klasifikasi sebesar 95,28%, membuktikan bahwa CNN mampu mengidentifikasi pola-pola visual yang khas dari masing-masing penyakit. [4]

Keempat, penelitian oleh Ali Mustopa bertujuan untuk membangun model klasifikasi penyakit ayam melalui citra *feses* dengan pendekatan *transfer learning*. Penelitian ini menyoroti tantangan ketidakseimbangan data dan kurangnya pemahaman peternak terhadap gejala penyakit seperti *Coccidiosis* (coklat), *Newcastle* (putih kehijauan), dan *Salmonella* (putih). Dataset yang digunakan mencakup 8.046 citra dari *Kaggle* dan diproses dengan augmentasi serta resampling. Berbagai model CNN seperti *EfficientNetV2L*, *EfficientNetB7*, *ResNet50*, *MobileNetV2*, dan *VGG16* diuji dalam berbagai skenario. Hasil terbaik diperoleh dari *EfficientNetV2L*

dengan akurasi 97%, lebih tinggi dari model CNN buatan sendiri (93%), menegaskan bahwa transfer *learning* dengan tuning *hyperparameter* mampu meningkatkan kinerja deteksi secara signifikan. [5]

Penelitian kelima oleh Hossain dari Bangladesh menawarkan pendekatan berbasis aplikasi *smartphone* yang dapat melakukan deteksi cepat terhadap penyakit ayam hanya melalui foto *feses*. Penelitian ini menangani penyakit *Salmonella*, *Coccidiosis*, dan *Newcastle Disease*. Sistem yang dikembangkan menggunakan *ensemble learning* dari empat CNN terbaik (*VGG16*, *ResNet50*, *Xception*, *InceptionV3*), dengan *Random Forest* sebagai *meta-classifier*. Dataset diambil dari gambar nyata yang diambil melalui kamera ponsel. Hasilnya menunjukkan akurasi luar biasa hingga 99.99%, dan sistem ini berhasil diuji dalam bentuk aplikasi *Android* dengan waktu deteksi kurang dari 5 detik, menjadikannya solusi praktis, murah, dan efektif untuk peternak lokal. [6]

Kelima penelitian di atas secara kolektif memperlihatkan potensi besar pemanfaatan kecerdasan buatan dalam deteksi dini penyakit ayam, dengan pendekatan yang berbasis citra *feses*, penggunaan CNN, dan optimalisasi melalui *transfer learning* atau *ensemble*. Masing-masing pendekatan memiliki keunggulan tersendiri dalam akurasi, kompleksitas model, dan kesiapan untuk diimplementasikan di lapangan

2.3 Computer Vision

Computer vision adalah cabang dari kecerdasan buatan (*Artificial Intelligence/AI*) yang memungkinkan komputer untuk "melihat", mengenali,

dan memproses citra seperti halnya manusia. Dalam konteks identifikasi penyakit pada ayam, *computer vision* digunakan untuk menganalisis gambar atau video dari ayam secara otomatis guna mendeteksi gejala visual penyakit, seperti perubahan warna bulu, bentuk mata, luka di kulit, dan sebagainya. Teknologi ini sangat penting karena dapat meningkatkan efisiensi dan akurasi dalam proses deteksi dini penyakit, serta mengurangi ketergantungan pada pemeriksaan manual oleh peternak atau dokter hewan.

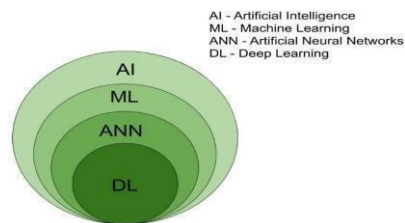
2.4 *Machine Learning*

Machine learning (pembelajaran mesin) adalah pendekatan dalam kecerdasan buatan yang memungkinkan sistem untuk belajar dari data dan membuat prediksi atau keputusan tanpa diprogram secara eksplisit. Dalam studi identifikasi penyakit ayam, *machine learning* berperan dalam membangun model prediksi berdasarkan data citra ayam yang telah diberi label sesuai jenis penyakitnya. Dengan demikian, model dapat mempelajari pola-pola visual yang menunjukkan indikasi penyakit tertentu dan mengklasifikasikan ayam yang sehat atau sakit secara otomatis.

2.5 *Deep Learning*

Deep learning merupakan cabang dari machine learning yang memanfaatkan jaringan saraf tiruan yang memiliki banyak lapisan tersembunyi untuk memahami konteks manusia. Deep learning memainkan peran penting dalam sistem kemudi otomatis untuk kendaraan, termasuk tugas-tugas seperti mengenali ucapan, memproses gambar, mendeteksi

anomali, dan memahami bahasa, semuanya tanpa bergantung pada aturan pemrograman yang telah ditentukan sebelumnya atau keahlian manusia yang dominan



Gambar 2. 1 Venn Deep Learning

Pada *deep learning* tersedia menyeluruh tanpa adanya pembagian pada proses fitue dan proses pengenalan menggunakan metode pembelajaran mesin[5].

Gambar 2.4 merupakan *venn deep learning*.

2.6 Pengolahan Citra

Pengolahan citra (*image processing*) adalah teknik untuk menganalisis dan memanipulasi citra digital guna mengekstrak informasi yang berguna. Tahapan ini merupakan bagian penting dalam sistem identifikasi penyakit berbasis visual karena memungkinkan sistem untuk melakukan pra-pemrosesan seperti peningkatan kualitas gambar (*enhancement*), segmentasi bagian tubuh ayam, serta deteksi area yang menunjukkan gejala penyakit. Pengolahan citra berperan sebagai fondasi dalam sistem *computer vision* karena kualitas input akan sangat memengaruhi performa analisis selanjutnya.

2.7 Convolutional Neural Network (CNN)

Jaringan saraf tiruan sering digunakan dalam ide pembelajaran

mendalam karena kemiripannya dengan pendekatan terstruktur yang ditemukan dalam pembelajaran mendalam. Contoh spesifik dari jenis algoritma ini adalah jaringan saraf konvolusional (CNN), yang berevolusi dari multilayer perceptron (MLP) [5]. CNN berisi lapisan konvolusional yang bertanggung jawab untuk ekstraksi data.

2.8 Konsep Convolutional Neural Network (CNN)

Jaringan Neural Konvolusional, yang umumnya dikenal sebagai CNN, dianggap sebagai jaringan neural dalam karena sering digunakan dalam pemrosesan gambar dan kedalamannya yang substansial [5]. Awalnya, CNN dipasangkan dengan Multilayer Perceptron untuk tugas-tugas yang berkaitan dengan klasifikasi gambar. Namun, stabilitas MLP dipertanyakan karena memperlakukan setiap piksel sebagai fitur yang terisolasi. Hal ini mengakibatkan hasil yang salah dan menghambat retensi detail spasial dalam data gambar. Penggunaan awal CNN diamati dalam studi korteks visual pada kucing, yang dilakukan oleh Hubel dan Wiesel (Hubel dan Wiesel, 1968) [5]. CNN dirancang dengan beberapa tahapan yang memfasilitasi pelatihan. Berbagai tahapan CNN mencakup input dan output. Setiap tahapan ini terdiri dari beberapa array, yang biasanya disebut peta fitur. Setiap tahapan dibangun dari tiga komponen: konvolusi, fungsi aktivasi, dan pooling.

B. Struktur Jaringan CNN

1. Convolutional Layer

Dalam matematika, konvolusi adalah proses penggunaan berulang

output dari fungsi lain. Dalam bidang pemrosesan gambar, konvolusi berarti menerapkan kernel konvolusi (kotak kuning) ke gambar di semua lokasi yang mungkin. Lapisan konvolusi mensimulasikan bagaimana otak mempersepsikan visual dan mempelajari filter khusus untuk gambar masukan. Konvolusi mengidentifikasi fitur dalam gambar masukan dan kemudian menghasilkan transformasi linier data berdasarkan karakteristik spasialnya.

Pada Gambar 2.2, seluruh kotak hijau menggambarkan gambar yang dikonvolusi. Kernel konvolusi yang digambarkan pada Gambar 2.2 bergerak dari sudut kiri atas ke sudut kanan bawah.

1	1	1	0	0
0	1	1	1	0
0	0	1	1	1
0	0	1	1	0
0	1	1	0	0

Image

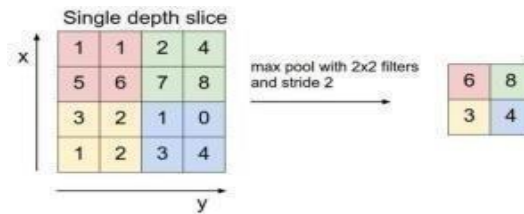
4	3	4
2	4	3
2		

Convolved Feature

Gambar 2. 2 Operasi Konvolusi *Layer*

2. *Pooling Layer*

Pooling mengacu pada teknik meminimalkan data dalam matriks melalui operasi pooling. Lapisan pooling terdiri dari filter yang memiliki dimensi dan stride tertentu, yang bergantian saat bergerak selama fase pemetaan. Pendekatan *Convolutional Neural Network* (CNN) menggunakan max pooling. Metode ini melibatkan pemisahan output dari konvolusi menjadi beberapa bagian kecil, di mana nilai tertinggi dalam setiap bagian diambil untuk membuat gambar yang lebih kecil. Gambar 2.6 menunjukkan operasi max pooling.

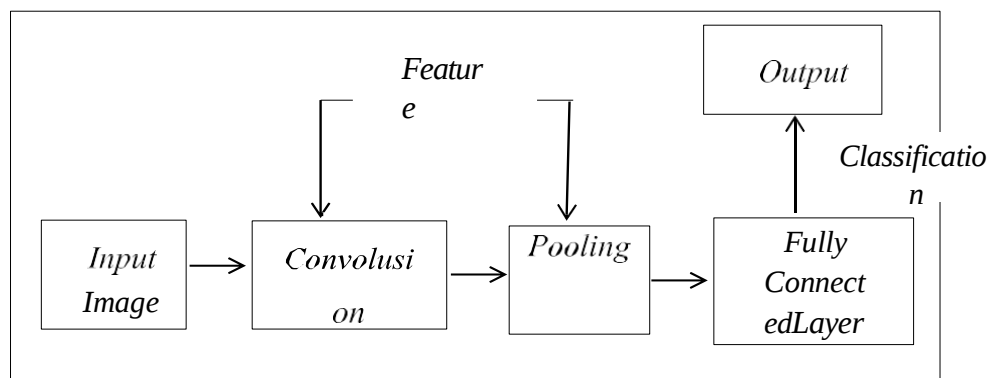
Gambar 2. 3 *Pooling Layer*

3. *Fully Connected Layer*

Lapisan dense sering digunakan dalam desain MLP untuk mengubah data dan kemudian mengaturnya secara linier. Setiap jaringan saraf dari lapisan konvolusional harus diubah terlebih dahulu menjadi format satu dimensi sebelum memasuki lapisan dense. Karena proses ini menyebabkan hilangnya sebagian data dan tidak dapat dibatalkan, lapisan dense hanya digunakan di bagian akhir jaringan.

B. Langkah-Langkah *Convolutional Neural Network (CNN)*

Langkah-langkah proses CNN yaitu :



- a) Awalnya, gambar dibagi menjadi beberapa bagian yang lebih kecil dan saling tumpang tindih. Hasil penerapan konvolusi pada setiap bagian digunakan sebagai input untuk menghasilkan representasi fitur yang memungkinkan CNN untuk mengidentifikasi suatu objek.

Akibatnya, invariansi translasi didefinisikan sebagai tidak bergantung pada lokasi objek dalam gambar.

- b) Setiap gambar yang lebih kecil diproses melalui jaringan saraf kecil. Semua segmen gambar menjalani konvolusi dengan filter yang identik. Akibatnya, setiap bagian gambar diberi pengali yang sama, sebuah teknik yang disebut pembagian bobot (weight sharing) dalam jaringan saraf. Jika sebuah gambar mengandung informasi penting, gambar tersebut diidentifikasi sebagai objek yang menarik. Hasil dari pembagian bobot menghasilkan peta aktivasi. Untuk meningkatkan penggambaran gambar, proses konvolusi dilakukan lagi menggunakan berbagai filter untuk menghasilkan peta aktivasi tambahan.
- c) Proses pembelajaran fitur dibagi menjadi tiga komponen: konvolusi, lapisan aturan (juga dikenal sebagai fungsi aktivitas), dan lapisan pooling. Urutan ini diulangi hingga peta fitur dibuat yang memenuhi persyaratan untuk tahap klasifikasi. Lapisan konvolusi terdiri dari jaringan saraf yang membuat filter dengan dimensi yang ditentukan dalam tinggi (dalam piksel) dan panjang. Operasi titik terjadi dengan setiap pergerakan antara input dan nilai filter untuk menghasilkan hasil yang biasa disebut peta fitur atau peta aktivasi. Peta fitur dapat ditentukan menggunakan rumus ini.

$$\text{Output size} = \frac{W - F + 2P}{S} + 1$$

dengan keterangan:

- W = ukuran panjang/lebar input (dalam pixel)
- F = ukuran filter/kernel (dalam pixel)
- P = padding (jumlah pixel yang ditambahkan di tepi input)
- S = stride (langkah pergeseran filter pada input)
- Output size = ukuran panjang/lebar output feature map.

Dari persamaan rumus tersebut, maka ukuran spasial dari volume *output* dapat dihitung dimana *hyperparameter* yang digunakan ialah ukuran *filter* (F), *Stride* yang diterapkan (S), [8] jumlah *padding* nol yang digunakan (P) dan volume (V)[4]. *Stride* yaitu nilai yang berfungsi untuk menggeser filter melalui input citra dan *Zero padding* yaitu nilai untuk mendapatkan angka nol di sekitar border citra.

- Fungsi Aktivasi (*Activation Function*)

Fungsi aktivasi merupakan komponen penting dalam jaringan saraf tiruan, termasuk CNN, yang berperan untuk memperkenalkan *non-linearitas* ke dalam model. Tanpa fungsi aktivasi, jaringan saraf akan bertindak sebagai model linear sederhana dan tidak mampu mempelajari pola-pola kompleks dalam data. Fungsi ini diterapkan pada output dari setiap neuron setelah proses komputasi (*dot product*) untuk menentukan apakah suatu informasi harus diteruskan ke neuron berikutnya atau tidak.

Dalam CNN, fungsi aktivasi yang paling umum digunakan adalah

ReLU (*Rectified Linear Unit*) karena kemampuannya dalam mempercepat proses pelatihan dan mengurangi masalah *vanishing gradient*. Selain ReLU, terdapat juga fungsi-fungsi lain seperti *Sigmoid*, *Tanh*, dan *Leaky ReLU*, yang digunakan tergantung pada kebutuhan dan karakteristik data.

- *Learning Rate*

Learning rate adalah parameter penting dalam proses pelatihan jaringan saraf yang menentukan seberapa besar langkah pembaruan bobot yang dilakukan oleh algoritma optimisasi setiap kali terjadi perhitungan error. Nilai learning rate yang terlalu besar dapat menyebabkan model melewati nilai minimum global (*overshooting*), sementara nilai yang terlalu kecil akan membuat proses pelatihan sangat lambat dan berpotensi terjebak pada local minimum.

Dalam CNN, pemilihan learning rate yang tepat sangat berpengaruh terhadap kinerja model. Untuk mengatasinya, beberapa pendekatan seperti *learning rate decay*, *adaptive learning rate* (seperti pada Adam), atau *cyclical learning rate* digunakan agar proses pembelajaran menjadi lebih efisien dan stabil.

- *Optimizer* (Algoritma Optimasi)

Optimizer adalah algoritma yang digunakan untuk meminimalkan fungsi loss dengan menyesuaikan bobot jaringan saraf berdasarkan hasil dari *backpropagation*. Tujuan utamanya adalah

menemukan kombinasi bobot yang menghasilkan *error* paling rendah terhadap data pelatihan.

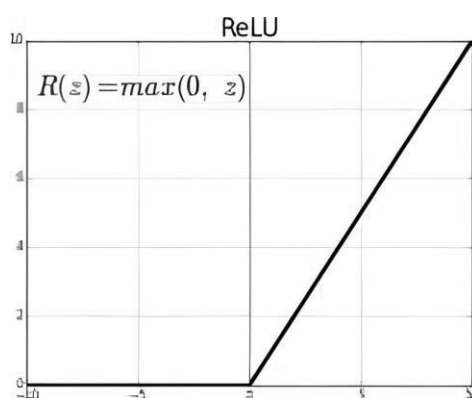
Beberapa *optimizer* yang umum digunakan dalam CNN antara lain:

- **Stochastic Gradient Descent (SGD):** Optimizer dasar yang bekerja berdasarkan perhitungan gradien terhadap batch kecil data.
- **Adam (Adaptive Moment Estimation):** Optimizer yang menggabungkan kelebihan dari momentum dan RMSProp, serta menyesuaikan learning rate untuk setiap parameter secara adaptif.
- **RMSProp:** Cocok untuk data dengan gradien yang berubah-ubah secara dinamis, seperti pada data visual atau citra.
-
- **Adagrad dan Adadelta:** Optimizer yang menyesuaikan learning rate secara otomatis berdasarkan frekuensi parameter.

Pemilihan optimizer yang sesuai akan berdampak besar pada konvergensi model serta kecepatan dan akurasi proses pelatihan.

Fungsi Rule, yang dikenal sebagai Rectified Linear Unit (ReLU), mewakili output dari jaringan saraf, menghasilkan nilai nol ketika input kurang dari nol. Lapisan pooling biasanya mengikuti lapisan konvolusi dalam arsitektur. Operasinya melibatkan filter dengan ukuran dan stride tertentu yang bergerak melintasi seluruh peta fitur. Proses ini mengurangi

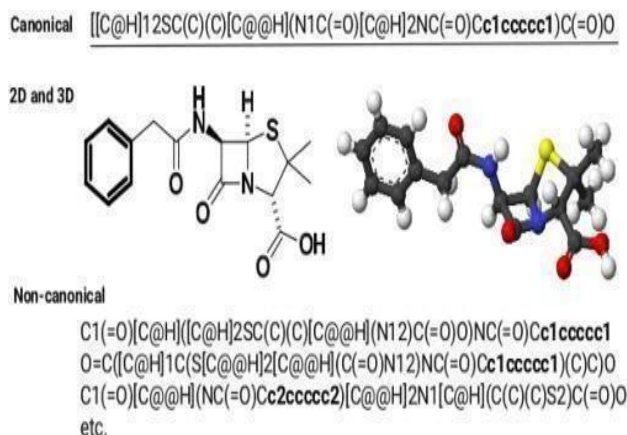
ukuran peta fitur (*downsampling*), yang mempercepat perhitungan karena jumlah parameter yang perlu disesuaikan lebih sedikit. Akibatnya, ini membantu mencegah overfitting. Pooling dapat dikategorikan menjadi dua jenis: Average Pooling dan Max Pooling. Gambar 2.4 mengilustrasikan fungsi aktivitas Rule.



Gambar 2. 4 Fungsi Aktivitas *Rule*

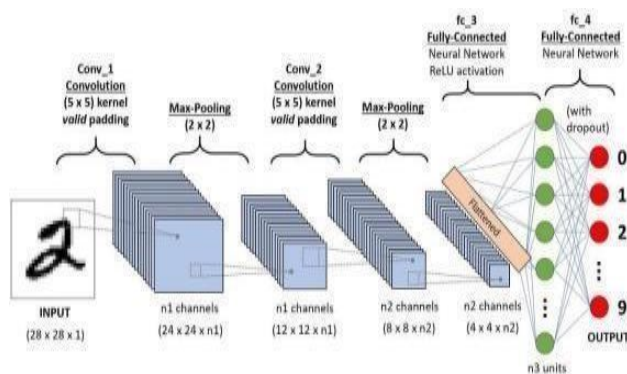
SWIS adalah metode yang menggabungkan teknik Transformer dan *Convolutional Neural Networks* (CNN) untuk pemodelan QSAR (*Quantitative Structure-Activity Relationship*), yang sering digunakan dalam prediksi aktivitas biologis senyawa kimia. SWIS menggunakan CNN untuk mengekstrak fitur-fitur lokal, sementara Transformer menangkap dependensi global dalam data. Ini memberikan fleksibilitas dan akurasi lebih tinggi dalam analisis kimia dan biologi.

Diagram Arsitektur SWIS: Diagram arsitektur SWIS akan menunjukkan integrasi antara CNN dan Transformer. Biasanya, ini melibatkan lapisan konvolusi diikuti dengan lapisan Transformer untuk menangkap informasi kontekstual lebih luas.



Gambar 2. 5 SWIS (Swiss Knife for QSAR Modeling)

Rulex CNN merupakan pengembangan dari Rule CNN yang otomatis mengekstrak aturan dari data.



Gambar 2. 6 Rulex Matrix

Teknik ini memanfaatkan ekstraksi fitur otomatis untuk belajar aturan yang relevan langsung dari dataset, tanpa harus menyusun aturan secara manual. Ini sangat berguna untuk masalah yang dinamis atau kompleks.

Diagram Arsitektur Rulex CNN: Diagram ini biasanya menunjukkan integrasi lapisan CNN dengan mekanisme ekstraksi aturan otomatis, yang membantu model belajar langsung dari data.

d) Hasil dari *feature learning* ialah *feature map* yang masih berbentuk

multidimensional *array*, sehingga harus melakukan “*flatten*” atau “*reshape feature map*” menjadi sebuah *vector* sebagai bisa digunakan sebagai input dari *fully-connected* layer[5]. Lapisan *fully-connected* merupakan suatu lapisan dimana seluruh *neural* aktivitas dari lapisan sebelumnya semua terhubung dengan *neural* dilapisan selanjutnya seperti pada jaringan syaraf tiruan yang sering digunakan pada metode Multi Lapisan Perceptron yang berfungsi untuk mengolah data sehingga bisa diklasifikasikan. Perbedaan lapisan konvolusi dan lapisan *fully-connected* ialah pada lapisan konvolusi *neural* dilapisan konvolusi hanya terhubung ke daerah tertentu pada input. Sedangkan *fully-connected* mempunyai *neural* secara keseluruhan terhubung. Tetapi, kedua lapisan tersebut masih mengoperasikan produk *dot*, sehingga fungsinya hampir sama.

2.9 Python

Python adalah bahasa pemrograman yang tersedia secara gratis dan dirancang untuk meningkatkan kompatibilitas, integrasi, efisiensi pengembang, keunggulan perangkat lunak, dan kemampuan untuk mentransfer program. *Python* banyak digunakan dalam berbagai inisiatif pengembangan perangkat lunak, seperti pembuatan antarmuka pengguna, perhitungan numerik, penyesuaian produk, pemrograman tingkat sistem, dan pembuatan skrip berbasis web

2.10 TensorFlow

TensorFlow adalah sebuah *framework open-source* untuk *machine learning* (ML) dan *deep learning* (DL) yang dikembangkan oleh Google

Brain. *TensorFlow* menyediakan berbagai alat untuk membangun dan melatih model ML, khususnya untuk aplikasi yang melibatkan data besar dan perhitungan yang kompleks, seperti *computer vision*, *natural language processing*, dan *speech recognition*.

2.11 Confusion Matrix

Dalam penelitian ini, matriks kesalahan, yang umumnya dikenal sebagai matriks kebingungan, digunakan untuk menilai presisi. Matriks kebingungan adalah grafik yang menunjukkan berapa banyak titik data uji yang diklasifikasikan dengan benar atau salah. Matriks ini mencakup empat metrik utama untuk menggambarkan hasil klasifikasi:

1. *True negative (TN (TN): Data negatif yang diidentifikasi secara akurat.*
2. *True positive (TP): Data positif yang diidentifikasi secara akurat.*
3. *False negatif (FN) Data positif yang salah diidentifikasi. Ini berarti bahwa semua kejadian kemungkinan besar positif (P) dan semua kejadian kemungkinan besar negatif (N)*
4. *False positif (FP) Data negatif yang diidentifikasi secara akurat.*

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

Gambar 2. 7 Convolution Matriks

Nilai yang ada pada gambar 2.7 diatas bisa digunakan untuk menghitung akurasi dengan persamaan.

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}}$$

Presisi mengacu pada tingkat keberhasilan model dalam mengidentifikasi kejadian positif dalam skenario prediksi. Sebuah rumus digunakan untuk mengukur akurasi.

$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{P} + \text{N}}$$

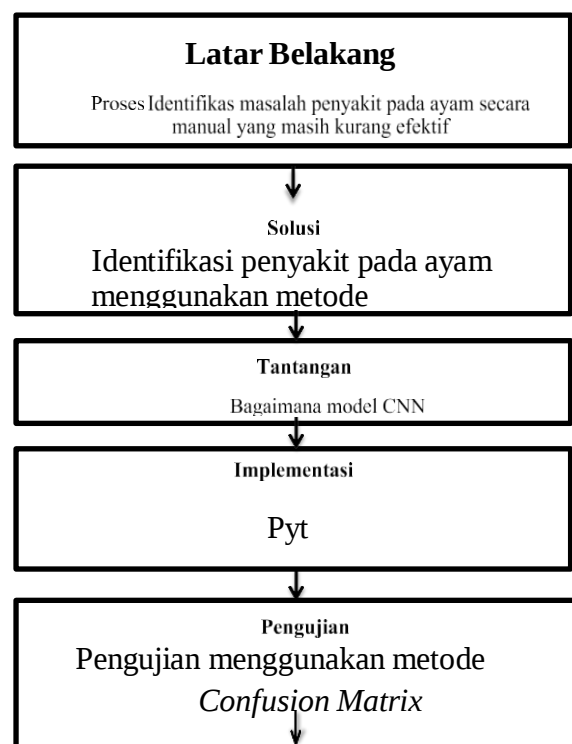
Akurasi berfungsi sebagai metrik seberapa baik model mengklasifikasikan hasil. Recall dapat ditentukan dengan menggunakan rumus tertentu.

$$\text{Recall} = \frac{TP}{TP + FN}$$

F1 Score merupakan rata-rata dari *precision* dan *recall* yang dapat dihitung dengan persamaan

$$F1 \text{ Score} = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}$$

4.3 Kerangka Pikir



Kerangka pikir dalam penelitian ini mengenai identifikasi penyakit ayam menggunakan metode CNN dapat di lihat pada gambar

