BAB II
DASAR TEORI

2.1 KAJIAN PUSTAKA

Beberapa penelitian sebelumnya telah mengkaji klasifikasi citra *x-ray* dengan menggunakan CNN sebagai metode utama karena metode ini terbukti optimal untuk pengolahan citra. Selain metode *deep learning* beberapa penelitian juga telah menerapkan metode *machine learning* seperi SVM dan KNN untuk tujuan serupa.

Tim peneliti dari Universitas Qatar di Doha dan Universitas Dhaka di Bangladesh bersama dengan mitra Malaysia bekerja sama dengan Hamad *Medical Corporation* dan dokter Bangladesh membuat database kasus *chest x-ray* positif TB sebanyak 700 foto dan 3500 *chest x-ray* normal [7]. Ada beberapa penelitian berdasarkan data ini yaitu penelitian yang dilakukan oleh Adrian Trueba Espinosa, dkk tahun 2021 tentang klasifikasi penyakit paru-paru menggunakan metode *convolutional neural network* hasil akurasi yang didapat mengenai akurasi matrix untuk ResNet-50 adalah 72% [8].

Salah satu penelitian terdahulu yang relevan adalah penelitian yang dilakukan oleh Qahtan Said dan rekan pada tahun 2021. Penelitian tersebut bertujuan untuk mengidentifikasi tuberkulosis paru menggunakan jaringan syaraf tiruan dengan algoritma *Backpropagation.* Hasil penelitian menunjukkan bahwa model yang dikembangkan memiliki rata-rata akurasi sebesar 84.82%, presisi sebesar 86.13%, dan *recall* sebesar 83.48% [4].

Penelitian oleh Geavanny Elok Fahrusiana, tahun 2019 dengan menggunakan metode SVM digunakan nilai *hyperparameter* yang dioptimalkan, model peramalan *Support Vector Regression* (SVR) dan nilai *Mean Squared Error* (MSE) sebesar 10540,27 nilai *Mean Absolute Percentage Error* (MAPE) sebesar 7.19%. Sedangkan model antisipasi yang tujuan utamanya 1 variabel yaitu jumlah kuantitas kasus tuberkulosis dapat membuat nilai kesalahan MAPE dan MSE lebih sederhana yaitu 2,46% untuk nilai MAPE dan 4579,68 untuk nilai MSE [9]. Berikut Tabel 2.1 merupakan perbandingan penelitian yang telah dilakukan oleh peneliti sebelumnya.

Tabel 2. 1 Perbandingan Penelitian

No	Peneliti	Judul Penelitian	Metode	Hasil Akurasi (%)	Perbedaan Penelitian
1.	Adrian Trueba, dkk	*Classification of Pulmonary Diseases from X-ray Images Using a Convolutional Neural Network*	*Convolutional Neural Network* menggunakan arsitektur ResNet-50	87%	*Convolutional Neural Network* menggunakan tujuh konvolusi layer
2.	Firda Rahmatul Ummah , dkk	*Covid-19 and Tuberculosis Detection in X-Ray of Lung Images with Deep Convolutional Neural Network*	*Convolutional Neural Network* dengan empat konvolusi *layer*	85,4%	*Convolutional Neural Network* menggunakan tujuh *layer* konvolusi
3.	Qahtan Said, dkk	Identifikasi Tuberkulosis Paru Berdasarkan Sinar-X Thorax Menggunakan Jaringan Syaraf Tiruan (JST)	Jaringan Syaraf Tiruan *Backpropagation*	84.82%	*Convolutional Neural Network*
4.	Ade Clinton Sitepu, dkk	Deteksi Penyakit Tuberkulosis Dengan Atensi Ganda CNN Pada Citra X-Ray	*Convolutional Neural Network* menggunakan dasar arsitektur LeNet	80%	*Convolutional Neural Network* menggunakan tujuh *layer* konvolusi
5.	Lina Annisa Widyasari	Sistem Deteksi Dini Penyakit Tuberkulosis Menggunakan (LVQ2)	*Learning Vector Quantization* 2 (LVQ2)	87,5%,	*Convolutional Neural Network*
6.	Favorisen Lumbanraja, dkk	Prediksi jumlah penderita penyakit tuberkulosis menggunakan metode SVM	Metode SVM menggunakan *Linear*, *Gaussian*, dan *Polynomial*	51.43%, 58.53% 36.03%	*Convolutional Neural Network*

2.2 DASAR TEORI

2.2.1 *Artificial Intellegence*

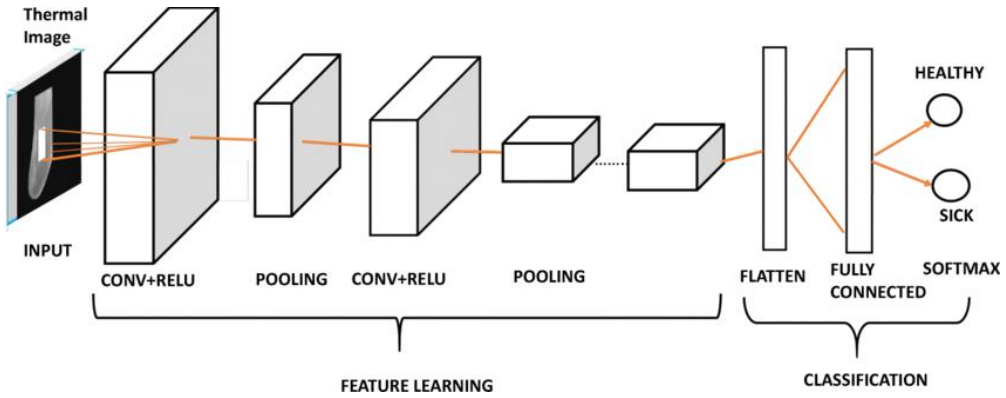
Artificial Intelligence (AI) atau kecerdasan buatan adalah bidang dalam ilmu komputer yang bertujuan untuk menciptakan mesin yang dapat meniru kemampuan otak manusia dalam memecahkan masalah. Fungsi umum AI termasuk analisis dan klasifikasi data mirip dengan cara otak manusia bekerja berdasarkan program tertentu dalam sistem komputer yang dibuat. Algoritma pemrograman digunakan sebagai kerangka berpikir berbasis AI untuk memproses berbagai jenis data. Pemrograman kecerdasan buatan memerlukan data yang besar dan kuat agar komputer dapat mengenali pola dengan baik. Dengan menggunakan data yang melimpah dan algoritma yang kompleks mesin AI dapat berpikir secara mandiri, membuat keputusan, belajar, dan beradaptasi [10].

2.2.2 *Deep Learning*

Deep learning adalah pembelajaran yang terdiri dari banyak lapisan. Lapisan pertama menghasilkan sifat yang sederhana, sedangkan lapisan terakhir menghasilkan sifat yang lebih kompleks. Teknologi pembelajaran mendalam digunakan dalam teknik klasifikasi, pengelompokan, segmentasi atau deteksi. Salah satu penerapan *deep learning* adalah data citra, tujuan klasifikasi citra adalah untuk mengelompokkan citra ke dalam kategori tertentu. Proses klasifikasi membutuhkan metode mulai dari pembelajaran mesin hingga melakukan tugasnya. Selain itu prosedur ekstraksi ciri diperlukan untuk pengenalan citra. Karakteristik gambar yang berbeda dapat menunjukkan kategori yang berbeda [11]. *Deep learning* melakukan pembelajaran mendalam dengan data besar dan dengan jumlah *layer* hingga ratusan. Semakin banyak sistem melakukan pembelajaran terhadap data maka semakin baik sistem melakukan pengenalan. *Deep learning* dapat melakukan pengenalan dengan akurasi tinggi bahkan dapat melebihi pengenalan yang dilakukan oleh manusia. Semakin banyak data *training* yang digunakan semakin tinggi akurasi pengenalan yang didapatkan. *Deep learning* memiliki tingkat akurasi yang lebih tinggi dibandingkan dengan lainnya yang membutuhkan dua syarat utama yaitu data besar dan diproses pada komputer dengan *High Performance Graphical Processing Unit* (GPU) [11].

2.2.3 *Convolutional Neural Network* (CNN)

CNN merupakan suatu lapisan yang mempunyai susunan neuron 3D (lebar, tinggi dan kedalaman). Lebar dan tinggi mengatasi ukuran lapisan sementara kedalaman mengacu pada jumlah lapisan. Kemampuan dasar dari perhitungan CNN ini adalah untuk mengambil setiap informasi persiapan data latih yang ditentukan untuk mengumpulkan informasi data dengan informasi data yang sebelumnya. *Input, Convolutional Layer, ReLU Layer, Pooling Layer, Flatten Layer, Fully Connected Layer*, dan *Softmax Layer* membentuk arsitektur CNN [12].



Gambar 2. 1 Arsitektur CNN [13]

Berdasarkan Gambar 2.1 ada dua lapisan arsitektur CNN. Lapisan pertama adalah *feature learning* yang terdiri dari *convolution layer* dan *pooling layer*. Masing-masing *layer* tersebut akan menghasilkan fitur *map* berupa angka sehingga dapat merepresentasikan citra untuk diteruskan ke bagian *layer* klasifikasi. Lapisan kedua adalah lapisan klasifikasi yang terdiri dari lapisan yang terhubung sepenuhnya dengan lapisan lainnya. Lapisan ini menerima masukan yang dihasilkan dari lapisan keluaran untuk bagian pembelajaran fitur. Kemudian pada bagian *feature learning* akan diproses secara *flatten* dengan menambahkan *hidden layer* pada koneksi penuh untuk menghasilkan *output* berupa nilai akurasi untuk setiap kelas kalsifikasi. CNN didasarkan pada proses konvolusional yang hanya dapat diterapkan pada dataset dengan struktur dua dimensi seperti objek gambar [13].

2.2.3.1 *Convolutional Layer*

Lapisan pertama arsitektur yang menerima input gambar langsung adalah *Convolutional Layer*. Ukuran kernel, faktor *skipping*, dan tabel koneksi adalah beberapa parameter yang membentuk *convolutional layer.* Faktor lompatan CNN adalah jumlah piksel yang bergeser di kernel yang selalu bergeser pada input [12]. Ilustrasi operasi konvolusi menggunakan Persamaan 2.1 ditunjukkan pada Gambar 2.2 ukuran *output* pada *map* adalah sebagai berikut:

$$M_x^n = \frac{M_x^n - K_x^n}{S_x^n + 1} + 1; \;\; M_x^n = \frac{M_y^n - K_y^n}{S_y^n + 1} + 1; \qquad (2.1)$$

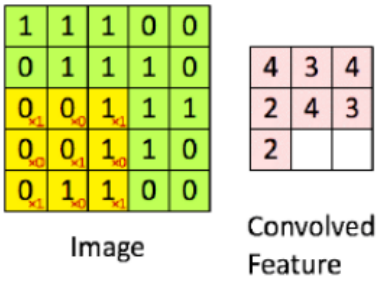
Keterangan

M_x^n, M_y^n = Ukuran fitur *map*

S_x, S_y = *Skipping factor*

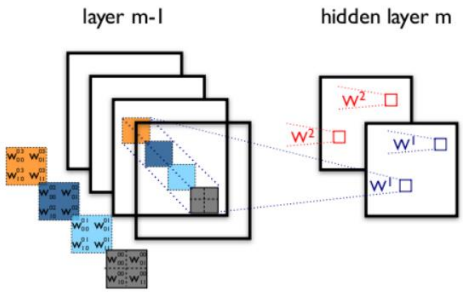
K_x, K_y = Ukuran *karnel*

n= Letak *layer* saat proses.



Gambar 2. 2 Operasi konvolusi [13]

Berdasarkan Gambar 2.2 semua kotak hijau adalah gambar yang akan melakukan konvolusi dari kiri atas ke kanan bawah. Kernel akan melakukan pergeseran kemudian hasilnya ditampilkan pada gambar di sebelah kanan untuk mengekstrak fitur dari masukan citra maka konvolusi digunakan [13].



Gambar 2. 3 Proses Konvolusi [14]

Berdasarkan Gambar 2.3 proses konvolusi hanya dapat digunakan dengan dataset yang memiliki struktur dua dimensi seperti objek gambar [14].

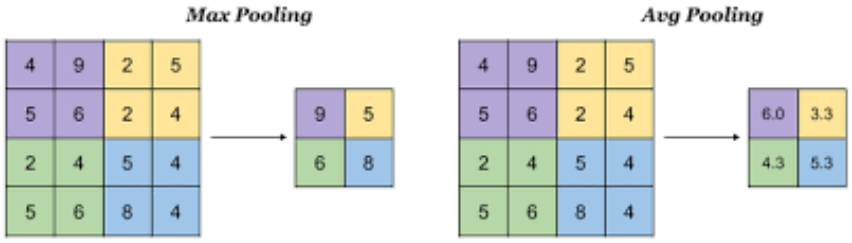
Berdasarkan informasi spesifik dalam Tabel 2.2 merupakan *hyparameter* pada konvolusi yang dapat menghasilkan transformasi linier dari data *input*. Berikut konvolusi kernel yang dimanfaatkan ditentukan oleh bobot lapisan [13].

Tabel 2. 2 *Hyparameter* pada *convolutional layer*

Parameter	Keterangan
Depth	Jumlah *layer* konvolusi atau kedalaman *layer* konvolusi.
Stride	Proses konvolusi terjadi pada jumlah pergeseran filter.
Zero -padding	Jumlah piksel yang ditambahkan di sekitar gambar ketika di proses.

2.2.3.2 *Pooling Layer*

Pooling layer digunakan untuk mengurangi ukuran fitur *map* agar lebih kecil. *Pooling layer* terdiri dari dua jenis yaitu *max pooling* dan *average pooling*. *Max pooling* adalah metode *pooling* yang paling sering digunakan untuk lebih spesifik dengan menggeser jendela ke seluruh permukaan gambar [15].

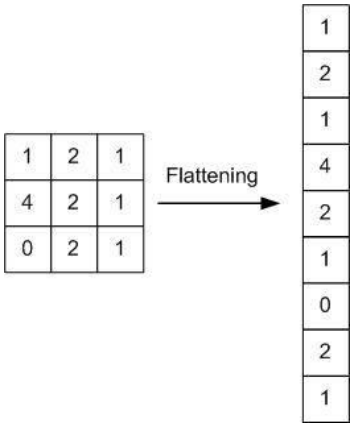


Gambar 2. 4 Operasi *pooling layer* [16]

Berdasarkan Gambar 2.4 penggunaan *pooling layer* diharapkan dapat memperkecil fitur *map* karena lebih sedikit parameter yang perlu diperbarui dan perhitungannya dapat dipercepat serta *overfitting* dapat dihindari.

2.2.3.3 *Flatten*

Flatten adalah lapisan yang akan mengubah fitur *map* sebelumnya yang merupakan matriks menjadi sebuah vektor satu dimensi. Hal ini dilakukan untuk memproses fitur *map* pada lapisan yang terhubung sepenuhnya [17]

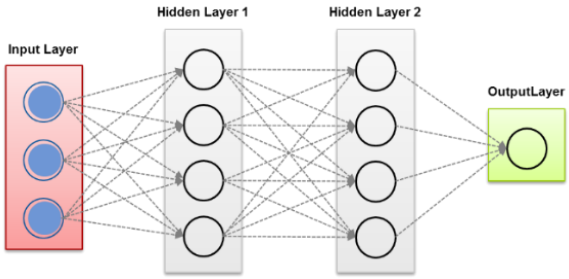


Gambar 2. 5 Proses pada lapisan *flatten* [17]

Berdasarkan Gambar 2.5 merupakan proses dalam membentuk kembali fitur *map* yang akan menjadi vektor untuk digunakan sebagai input dari *fully connected layer*. Pada proses *flattening* hasil nilai *flatten* pada *layer* diperoleh dengan mengalikan *size* terakhir yang didapatkan dari proses *feature learning* dengan *filter* yang digunakan.

2.2.3.4 *Fully Connected Layer*

Fully Connected Layer adalah lapisan yang menghubungkan setiap neuron ke lapisan lain yang saling berhubungan [12]



Gambar 2. 6 *Fully Connected Layer* [18]

Berdasarkan Gambar 2.6 neuron dikelompokkan menjadi beberapa lapisan seperti *input layer*, *hidden layer* dan *output layer*[12].

1) *Input layer*

   Lapisan ini digunakan untuk menggabungkan seluruh matriks fitur *map* yang diperoleh saat proses *pooling*. Kemudian *vector* dengan panjang sejumlah *pixels* dibuat dari matriks yang diperoleh melalui prosedur *pooling* [15].

2) *Hidden layer*

Perhitungan akan dilakukan oleh l*ayer* ini dengan menambahkan nilai bias dan mengalikan nilai *input layer* dengan bobot yang sudah ditetapkan. Untuk proses perhitungan dapat menggunakan Persamaan 2.2 berikut ini:

$$z\_in_i = \sum_{j=1}^{n} X_j * V_{j,i} + V_{o,i} \quad (2.2)$$

Keterangan

Z_in_i = *Input node hidden layer* ke-I dengan jumlah *node* n

X_j = *nod*e X ke j

$V_{j,i}$ = Bobot V untuk X_j dengan *node* Z_i

$V_{o,i}$ = Bias V untuk z_in_i

Selain itu dengan memasukkan fungsi aktivasi di semua hasil perhitungan maka nilai *output* Z akan didapatkan. *Output layer* kemudian memanfaatkannya selama proses perhitungan [15].

3) *Output layer*

Perhitungan hasil dari nilai *hidden layer* dikalikan dengan bobot yang diinisialisasi pada *layer* ini dan kemudian nilai bias ditambahkan ke hasil [15]. Untuk proses perhitungan dapat memanfaatkan Persamaan 2.3 sebagai berikut:

$$y_{in_i} = \sum_{j=1}^{m} Z_j * W + W_{o,i} \quad (2.3)$$

Keterangan

y_in_i = Masukan *node hidden layer* Z ke I dengan jumlah *node* m

$W_{j,i}$ = Bobot W untuk Z_J dengan *node* Y_i

$W_{o,i}$ = Bias W untuk y_in_i

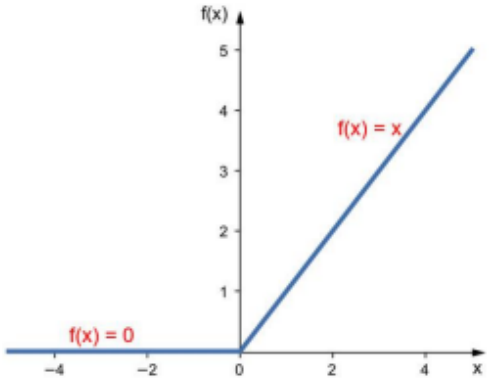
Z_j = *Node* Z ke j

2.2.3.5 Fungsi Aktivasi

Fungsi aktivasi merupakan fungsi non-linear yang memungkinkan jaringan saraf untuk melakukan transformasi data ke dimensi yang lebih tinggi. Dengan demikian jaringan saraf dapat mengekstrak pola dan fitur yang lebih kompleks. Hal ini memungkinkan jaringan saraf untuk membuat keputusan klasifikasi dengan menggunakan jalur sederhana dan efektif dalam memproses data yang kompleks [13].

a) *Rectified Linear Unit (ReLU)*

ReLU adalah salah satu fungsi aktivasi yang memiliki keunggulan dalam memproses data dalam jumlah besar dengan cepat. Fungsi ini melakukan pengecekan terhadap nilai nol pada data masukan, khususnya pada nilai piksel dalam citra [13]-[15].



Gambar 2. 7 Fungsi Aktivasi ReLU [19]

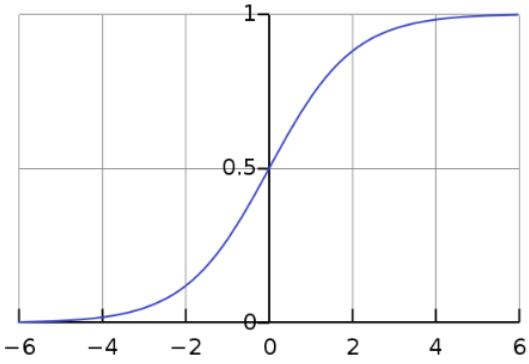
Berdasarkan Gambar 2.7 menunjukkan bahwa nilai piksel *input* gambar dibatasi ke nol oleh fungsi ini. Berikut Persamaan 2.4 merupakan matematis untuk fungsi aktivasi ReLu:

$$f(x)=\max(0,x) \qquad (2.4)$$

Dalam proses ini jika nilai piksel pada citra kurang dari nol maka nilai tersebut akan diubah menjadi nol [13] .

b) Sigmoid

Sigmoid adalah salah satu fungsi aktivasi yang sering digunakan dalam metode *convolutional neural network*. Fungsi ini memiliki karakteristik kontinu dan tidak menurun secara monoton. [20].



Gambar 2. 8 Fungsi aktivasi sigmoid [21]

Berdasarkan Gambar 2.8 Fungsi ini mengambil nilai riil sebagai *input* dan menghasilkan *output* dalam kisaran antara 0 hingga 1. Berikut Persamaan 2.5 matematis untuk fungsi aktivasi sigmoid:

$$\emptyset(z) = \frac{1}{1+e^{-z}} \qquad (2.5)$$

Semakin besar nilai *input* (lebih positif) semakin mendekati nilai *output* ke 1,0 sementara semakin kecil nilai *input* (lebih negatif) semakin mendekati nilai *output* ke 0,0 [20].

2.2.3.6 *Optimizer*

Optimizer adalah algoritma yang berfungsi untuk mencari bobot nilai terbaik dalam suatu model dengan tujuan meminimalkan kesalahan dan memaksimalkan hasil akurasi. Selama proses pelatihan *optimizer* akan melakukan penyesuaian pada bobot atau parameter model agar dapat mengurangi nilai fungsi kerugian (*loss function*) dan memprediksi data dengan tingkat akurasi yang tinggi [22].

a) *Adaptive Moment Estimation* (ADAM)

ADAM adalah suatu optimisasi yang menggunakan gradien orde pertama yang efisien secara komputasi dan membutuhkan penggunaan memori yang rendah. Algoritma ini merupakan salah satu metode optimasi *learning rate* [22]. Berikut Persamaan 2.6 merupakan perhitungan optimisasi ADAM:

$$\theta_{t+1} = \theta_t - \alpha \frac{n}{\sqrt{v_t}} m_t \qquad (2.6)$$

Dimana: α = *Learning rate*

m_t = estimasi momentum

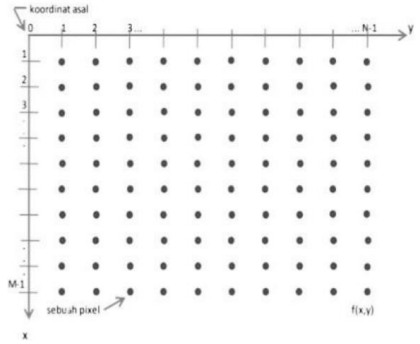
v_t = estimasi sub-gradien

2.2.4 Tuberkulosis

Bakteri *Mycobacterium tuberculosis* atau tuberculosis merupakan penyakit menular. Mikroba ini biasanya menyerang paru-paru. Ada beberapa cara untuk tertular tuberkulosis yaitu sumber penularannya melalui penderita tuberkulosis dengan Bakteri Tahan Asam (BTA) positif. Infeksi ini dapat disebabkan oleh penyebaran bakteri karena percikan dahak (inti tetesan) terlempar ke udara pada saat batuk [23].

2.2.5 Citra

Citra merupakan kumpulan beberapa *pixle* yang disusun dalam larik dua dimensi (matriks) yang berisi nilai nyata dan direpresentasikan dalam urutan bit tertentu. Proses *sampling* citra analog dibagi dengan proses pengambilan *sampel* menjadi baris M dan kolom N, di mana x dan y adalah koordinat spesifik dan f adalah intensitas atau skala abu-abu gambar pada titik koordinat tersebut (x,y) [24].



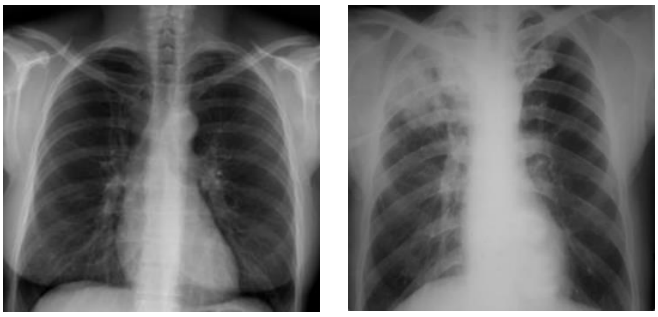
Gambar 2. 9 Titik Koordinat Citra [24]

Titik koordinat pada citra ditunjukan seperti pada Gambar 2.9 dimana pengolahan citra (*image processing*) merupakan tahapan untuk dapat mengolah *pixels* di dalam citra digital untuk maksud dan tujuan tertentu yang dimanfaatkan untuk dapat memperbaiki kualitas pada sebuah citra, pengenalan pada pola (*pattern recognition*), pengenalan identifikasi pada manusia (*biometric*), mendapatkan citra gambar/foto atau video (*content based image and video retrieval),* video editing, dan lain-lain[24]. Pengolahan citra membentuk proses suatu *input* menjadi *output*, maka pengolahan citra memiliki *input* berupa citra serta *output* berupa citra.

2.2.6 Citra *X-Ray*

Citra *X-Ray* adalah gambaran bagian dalam tubuh manusia yang dihasilkan melalui pemeriksaan menggunakan radiasi sinar-X dengan dosis yang rendah. Hasil pemeriksaan ini berperan penting sebagai pendukung bagi dokter dalam melakukan diagnosis terhadap keluhan pasien. Oleh karena itu, penting untuk membaca hasil pemeriksaan dengan tepat agar dapat menghindari hal-hal yang tidak diinginkan dan memastikan keakuratan dalam memberikan penilaian dan

tindakan medis [25]. Berikut contoh hasil pemeriksaan *x-ray* dengan kondisi normal dan kondisi tuberkulosis dapat dilihat pada Gambar 2.10 di bawah ini.



(a) Normal (b) Tuberkulosis

Gambar 2. 10 (a) Paru-paru normal (b) paru-paru positif TB [7]

Berdasarkan Gambar 2.10 memiliki perbedaan antara paru-paru sehat dan positif tuberkulosis. Perbedaan ciri yang dapat diamati adalah berdasarkan tingkat kekeruhan atau kegelapan pada citra. Pada citra normal paru-paru terlihat bersih tanpa adanya kekeruhan. Sementara pada citra tuberkulosis terdapat tingkat kekeruhan yang lebih tinggi pada kedua paru-paru dengan ciri-ciri sebagai berikut [26].

1. Bayangan mendung atau nodular di *apicopsterior* atas dan fragmen kurva bawah yang lazim.
2. Sebagian besar waktu ada lebih dari satu rongga dan bayangan keruh atau nodular buram mengelilinginya.
3. Bercak yang lebih besar dan Efusi pleura di kedua arah.

Oleh karena itu, perbedaan ciri-ciri tersebut menjadi fitur atau pola penting yang harus dipelajari oleh sistem agar dapat memprediksi input gambar dengan akurat sesuai dengan kondisi atau kelas yang sebenarnya [27].

2.2.7 Augmentasi Data

Augmentasi data adalah metode pemrosesan data visual yang memerlukan pengubahan atau modifikasi gambar sedemikian rupa sehingga komputer dapat mengenalinya sebagai gambar baru dengan tetap mengenali gambar yang diubah itu sebagai gambar yang sama [28]. Keakuratan model CNN dapat ditingkatkan dengan augmentasi namun hal ini sulit karena augmentasi mengumpulkan informasi tambahan pada pemembuatan model yang lebih dapat digeneralisasikan [29].

Gambar 2. 11 Augmentasi dengan perputaran gambar [29]

Berdasarkan Gambar 2.11 contoh augmentasi rotasi atau perputaran arah gambar dengan 90 derajat. Selain perputaran arah proses augmentasi bisa di lakukan dengan cara *zoom-in* secara acak, *resize*, *crop* secara acak dan *rotate*.

2.2.8 *Confusion Matrix*

Confusion matrix adalah suatu metode untuk mengukur kinerja atau tingkat kebenaran dari metode klasifikasi dengan menggunakan tabel untuk membandingkan hasil total data yang benar dan data yang salah dalam pengujian. Pengukuran kinerja menggunakan *confusion matrix* melibatkan perbandingan antara nilai prediksi dan nilai aktual dari kelas-kelas yang

Tabel 2. 3 Kondisi *Confusion Matrix*

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

Berdasarkan Tabel 2.3 *confusion matrix* memiliki empat kondisi yaitu *True Positive* (TP), *False Negative* (FN), *False Positive* (FP), dan *True Negative* (TN). TP adalah kondisi di mana model melakukan prediksi yang benar dan sesuai dengan kondisi aktual. FN adalah kondisi di mana model melakukan prediksi yang salah atau negatif, padahal kondisi aktualnya seharusnya positif. Kondisi ini dianggap sebagai kesalahan atau *error*. FP adalah kondisi di mana model melakukan prediksi yang positif atau benar, namun kondisi aktualnya seharusnya negatif. Kondisi ini juga dianggap sebagai kesalahan. Dapat diartikan bahwa antara FN dan FP, lebih baik jika terjadi FP. TN adalah kondisi di mana model

melakukan prediksi negatif dan sesuai dengan kondisi aktual negatif, atau dapat diartikan bahwa model memprediksi dengan benar.

Dengan adanya kondisi-kondisi tersebut, dapat dilakukan evaluasi dengan menghitung nilai Akurasi, Presisi, *Recall*, dan F1-*Score*. Nilai akurasi adalah perbandingan antara data yang diklasifikasikan dengan benar TP dan TN dengan keseluruhan data [30]. Persamaan 2.7 dapat digunakan untuk menentukan nilai akurasi sebagai berikut:

$$\text{Akurasi} = \frac{\text{TP+TN}}{\text{TP+TN+FP+FN}} \times 100\% \qquad (2.7)$$

Nilai presisi menggambarkan seberapa banyak jumlah data yang akurat diklasifikasi secara benar dibagi dengan total data yang diklasifikasi positif. Presisi merupakan rasio prediksi benar positif dibandingkan dengan keseluruhan hasil yang diprediksi positf. Persamaan 2.7 - 2.10 dapat digunakan untuk menentukan nilai presisi positif , presisi negatif dan total presisi.

$$\text{Presisi Positif} = \text{PPV} = \frac{\text{TP}}{\text{TP+FP}} \times 100\% \qquad (2.8)$$

$$\text{Presisi Negatif} = \text{NPV} = \frac{\text{TN}}{\text{TN+FN}} \times 100\% \qquad (2.9)$$

$$\text{Total Presisi} = \frac{\text{PPV+NPV}}{2} \times 100\% \qquad (2.10)$$

Recall menunjukkan beberapa persen data yang terklasifikasi dengan benar oleh sistem. Persamaan 2.11 – 2.13 dapat digunakan untuk menentukan nilai presentase *recall* positif, *recall* negatif dan total *recall*.

$$Recall \text{ Positif} = \text{TPR} = \frac{\text{TP}}{\text{P}} = \frac{\text{TP}}{\text{TP+FN}} \times 100\% \qquad (2.11)$$

$$RecalL \text{ Negatif} = \text{TNR} = \frac{\text{TN}}{\text{N}} = \frac{\text{TN}}{\text{TN+FP}} \times 100\% \qquad (2.12)$$

$$Total\ Recall = \frac{\text{TPR+TNR}}{2} \times 100\% \qquad (2.13)$$

F1- Score merupakan perbandingan rata-rata dari hasil presisi atau rasio prediksi benar positif dan *recall* yang dibobotkan. Untuk menentukan nilai presentasi F1-*Score* positif, F1-*Score Negatif* dan Total F1-*Score* dapat digunakan Persamaan 2.14 – 2.16 berikut:

$$F1\ Positif = 2 \times \frac{(\text{Total TPR} \times \text{Total PPV})}{(\text{Total TPR} + \text{Total PPV})} \qquad (2.14)$$

$$F1\ Negatif = 2 \times \frac{(\text{Total TNR} \times \text{Total NPV})}{(\text{Total TNR} + \text{Total NPV})} \qquad (2.15)$$

$$Total\ F1 = 2 \times \frac{(\text{Total Recall} \times \text{Total Precision})}{(\text{Total Recall} + \text{Total Precision})} \qquad (2.16)$$