

Dokumentasi: <https://github.com/allunsamudra/artificialintelligence>

Bagian 1: Pengumpulan dan Persiapan Data

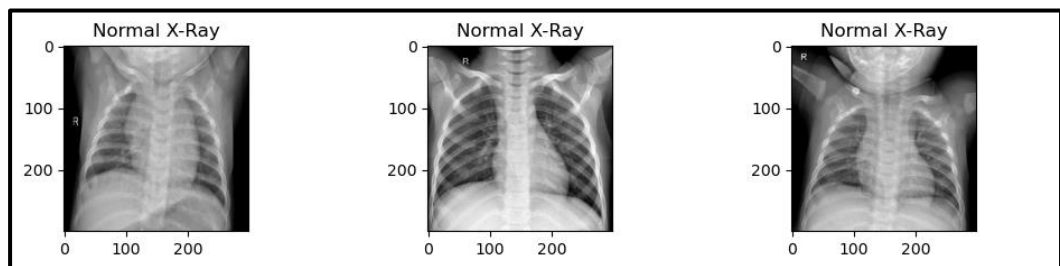
1. Pilih 3 kategori jenis objek yang berbeda

Dataset yang digunakan adalah citra *Chest X-Ray (Rontgen)*. Terdiri dari 3 kategori, yaitu:

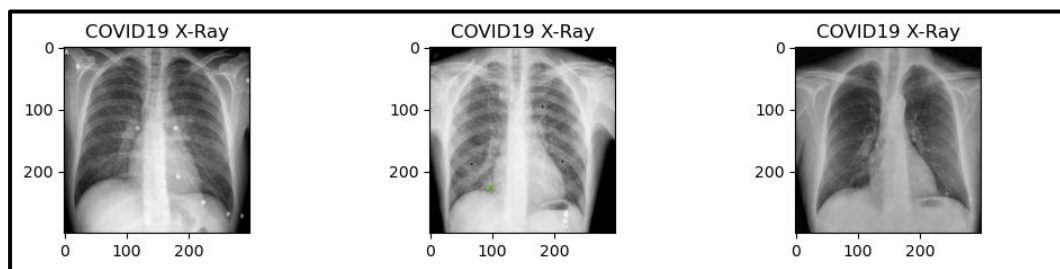
- **Normal:** Citra *Chest X-Ray* pasien dengan paru-paru normal
- **Covid 19:** Citra *Chest X-Ray* pasien penderita covid 19
- **Pneumonia:** Citra *Chest X-Ray* pasien penderita pneumonia

2. Dokumentasikan masing-masing objek tersebut dengan mengambil minimal 100 citra per kategori objek

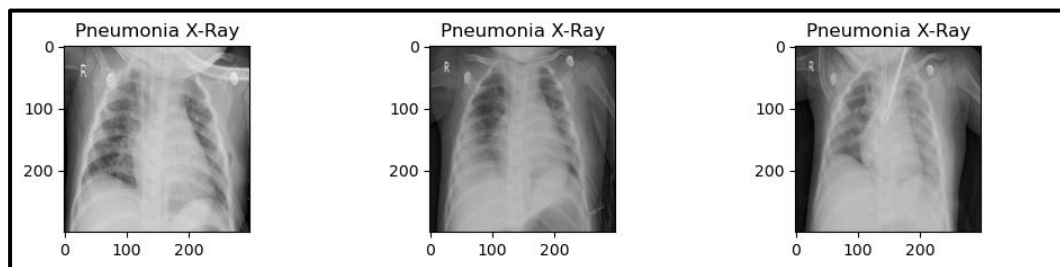
- Normal (3270 citra)



- Covid 19 (1281 citra)



- Pneumonia (4657 citra)



3. Lakukan preprocessing citra: resizing, normalisasi, augmentasi, dan pembagian data menjadi *training* dan *testing*.

Dataset yang digunakan sudah terbagi menjadi *train* dan *validation* dengan rasio *train* adalah 0.8 dan sisanya adalah *validation*.

Resizing

```
IMAGE_SIZE = (128, 128)
```

Citra akan diubah ukurannya menjadi 128x128 pixel dengan 3 chanel warna (RGB)

Data augmentation & Scaling

```
train_datagen = ImageDataGenerator(  
    rescale=1./255,  
    zoom_range=0.2,  
    rotation_range=15,  
    horizontal_flip=True  
)  
val_datagen = ImageDataGenerator(rescale=1./255)
```

Keterangan:

Data *augmentation* dilakukan dengan menormalkan piksel menjadi [0, 1], zoom acak $\pm 20\%$, rotasi acak hingga 15 derajat, dan pembalikan horizontal acak untuk meningkatkan variasi data pelatihan dan mencegah *overfitting*.

Splitting Data and Folder Processing

```
train_generator = train_datagen.flow_from_directory(  
    dir_train,  
    target_size=(128, 128),  
    batch_size=16,  
    class_mode='categorical'  
)  
val_generator = val_datagen.flow_from_directory(  
    dir_valid,  
    target_size=(128, 128),  
    batch_size=16,  
    class_mode='categorical'  
)
```

Keterangan:

Data pelatihan dan validasi dimuat menggunakan metode `flow_from_directory`, dengan ukuran gambar diubah menjadi 128×128 piksel dan batch size sebesar 16. Klasifikasi dilakukan secara multikelas (`class_mode='categorical'`) berdasarkan struktur folder yang merepresentasikan tiga kelas berbeda. Hasil pemuatan menunjukkan bahwa

terdapat 7.367 citra pada data pelatihan dan 1.841 citra pada data validasi, yang masing-masing terbagi ke dalam 3 kelas.

Bagian 2: Implementasi ANN untuk Klasifikasi

1. Rancang dan implementasikan model *Artificial Neural Network* (ANN) untuk mengklasifikasikan ketiga objek tersebut.

Artificial Neural Network (ANN) adalah model komputasi yang meniru cara kerja jaringan saraf manusia. ANN terdiri dari lapisan *input*, *hidden layer*, dan *output*, di mana setiap neuron saling terhubung melalui bobot. Model ini mampu mempelajari hubungan kompleks antar data melalui proses pelatihan

Define Model

```
model = Sequential([
    Input(shape=(128, 128, 3)),
    Rescaling(1./255),
    Flatten(),
    Dense(1024, activation='relu'),
    Dropout(0.5),
    Dense(512, activation='relu'),
    Dropout(0.4),
    Dense(256, activation='relu'),
    Dropout(0.3),
    Dense(3, activation='softmax') # 3 kelas output
])

model.compile(
    optimizer=Adam(learning_rate=0.0001),
    loss='categorical_crossentropy',
    metrics=['accuracy']
)

Model.summary()
history = model.fit(
    train_generator,
    epochs=15,
    validation_data=val_generator,
    callbacks=[]
)
```

2. Dokumentasikan secara jelas arsitektur ANN yang digunakan, mencakup.

Layer (type)	Output Shape	Param #
rescaling_1 (Rescaling)	(None, 128, 128, 3)	0
flatten_1 (Flatten)	(None, 49152)	0
dense_4 (Dense)	(None, 1024)	50,332,672
dropout_3 (Dropout)	(None, 1024)	0
dense_5 (Dense)	(None, 512)	524,800
dropout_4 (Dropout)	(None, 512)	0
dense_6 (Dense)	(None, 256)	131,328
dropout_5 (Dropout)	(None, 256)	0
dense_7 (Dense)	(None, 3)	771

a. Jumlah layer tersembunyi

Model ANN yang dibangun memiliki 3 *hidden layer*, yaitu:

- Dense (1024) – lapisan pertama
- Dense (512) – lapisan kedua
- Dense (256) – lapisan ketiga

b. Jumlah *neuron* per *layer*

- *Input layer*: 49152 *neuron*
- *Hidden layer 1*: 1024 *neuron*
- *Hidden layer 2*: 512 *neuron*
- *Hidden layer 3*: 256 *neuron*
- *Output layer*: 3 *neuron* (sesuai jumlah kelas)

c. Fungsi aktivasi dalam tiap layer

- Hidden Layer 1 (Dense 1024): ReLu
- Hidden Layer 2 (Dense 512): ReLU
- Hidden Layer 3 (Dense 256) : ReLu
- Output Layer (Dense 3): Softmax

d. Metode inisialisasi bobot awal

Secara *default*, *Dense layer Keras* menggunakan Glorot Uniform (Xvier Uniform) *initializer* untuk bobot

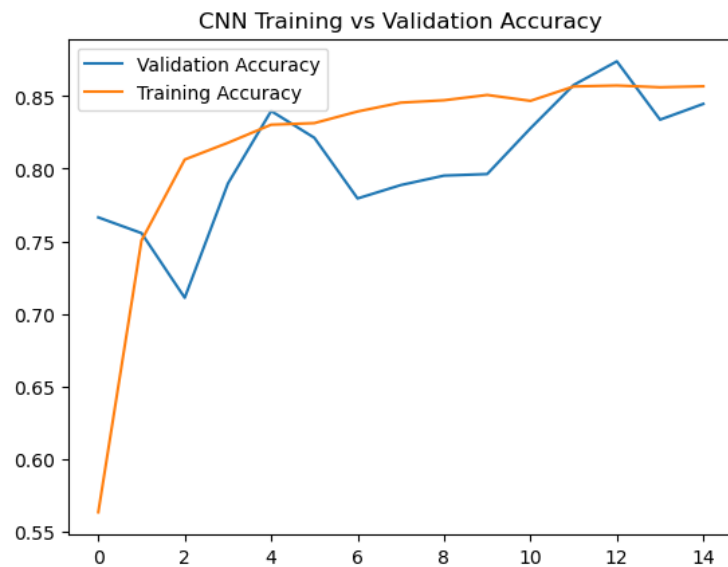
e. Metode Training dan Optimasi

- *Optimizer*: Adam()
- *Learning rate*: 0.0001
- *Loss Function*: categorical_crossentropy
- *Metrics*: accuracy

f. Rasio *Training* dan *Testing*

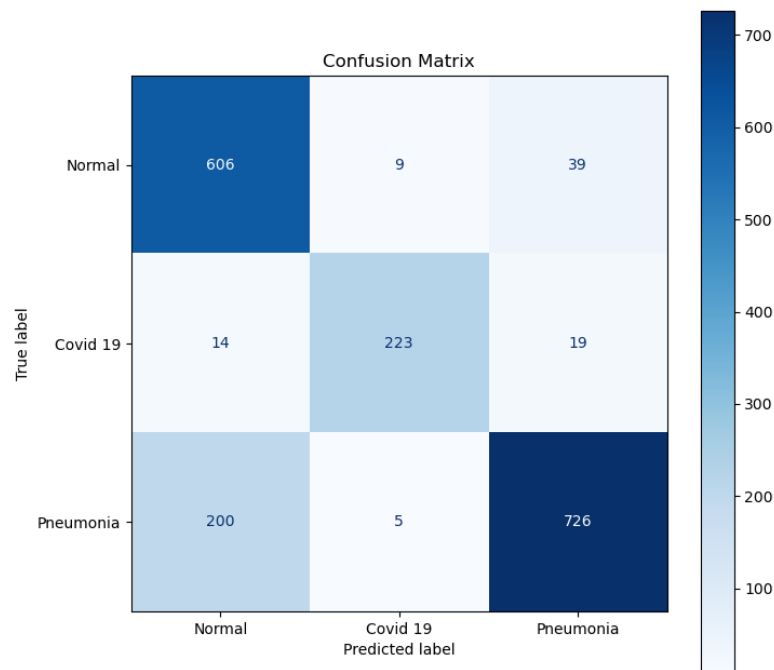
- *Training*: 80%
- *Testing*: 20%

g. Evaluasi Akurasi



- *Training Accuracy* (garis kuning) pada *epoch* pertama adalah 0.5038 dan terus meningkat hingga akurasi tertinggi ada pada *epoch* 14 yaitu 0.8598 dan di *epoch* 15 turun menjadi 0.8549.
- *Validation Accuracy* (garis biru) pada *epoch* pertama adalah 0.7664 dan terus meningkat hingga akurasi tertinggi ada pada *epoch* 13 yaitu 0.8740 dan di *epoch* 15 turun menjadi 0.8446.

Berdasarkan nilai akurasi dari *training* dan *validation*, model ANN sudah cukup baik dalam mengklasifikasikan citra *Chest X-Ray* menjadi 3 klasifikasi yaitu Normal, Covid 19 dan Pneumonia.



Berdasarkan *confusion matrix* dari model ANN, kesalahan klasifikasi paling banyak terjadi pada kelas pneumonia, di mana sekitar 200 citra pneumonia justru diprediksi sebagai citra normal.

3. Uji model dengan data baru dan lakukan prediksi terhadap gambar tersebut.
 - a. Prediksi data yang pertama dilakukan dengan menginputkan citra *Chest X-Ray* paru paru normal, klasifikasi menggunakan model ANN menunjukkan bahwa citra tersebut masuk ke dalam klasifikasi normal dengan akurasi sebesar 0.62.

Prediksi: Normal (0.62)

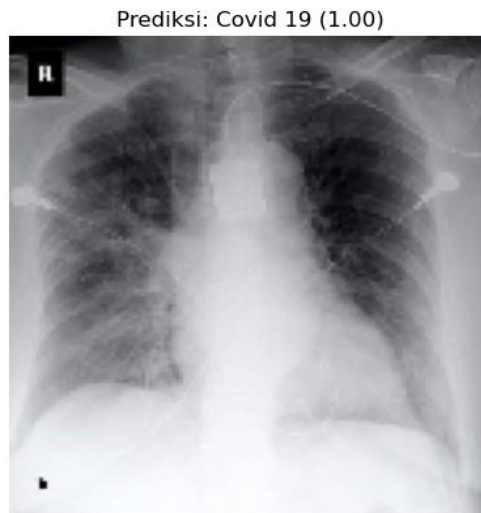


- b. Prediksi data yang kedua dilakukan dengan menginputkan citra *Chest X-Ray* penderita Pneumonia, klasifikasi menggunakan ANN menunjukkan bahwa citra tersebut masuk ke dalam klasifikasi Pneumonia dengan akurasi sebesar 0.95.

Prediksi: Pneumonia (0.95)



- c. Prediksi data yang kedua dilakukan dengan menginputkan citra *Chest X-Ray* penderita Covid 19, klasifikasi menggunakan ANN menunjukkan bahwa citra tersebut masuk ke dalam klasifikasi Covid 19 dengan akurasi sebesar 1.00.



4. Tulis analisis tentang performa ANN dan bagaimana proses tuning dilakukan untuk mendapatkan akurasi optimal

Model *Artificial Neural Network* (ANN) yang dibangun menggunakan arsitektur berlapis, terdiri dari tiga hidden layer dengan jumlah neuron masing-masing 1024, 512, dan 256, serta satu output layer dengan 3 *neuron* yang merepresentasikan tiga kelas: Normal, Covid-19, dan Pneumonia. Fungsi aktivasi yang digunakan di setiap *hidden layer* adalah ReLU, sedangkan output layer menggunakan *softmax* untuk klasifikasi multikelas. Untuk menghindari *overfitting*, diterapkan teknik regularisasi berupa *dropout* dengan rasio 0.5, 0.4, dan 0.3 secara bertahap pada setiap *layer*. Model dilatih menggunakan algoritma optimisasi Adam dengan *learning rate* sebesar 0.0001 dan *loss function categorical crossentropy*.

Proses *tuning* dilakukan melalui eksperimen bertahap terhadap jumlah *neuron*, kombinasi *dropout*, dan penyesuaian *learning rate*. Dari hasil pelatihan selama 15 *epoch*, diperoleh akurasi yang cukup stabil baik pada data pelatihan maupun validasi. Evaluasi model menggunakan *confusion matrix* dan *classification report* menunjukkan bahwa model mampu mengklasifikasikan *citra X-ray* dengan cukup baik, meskipun masih ditemukan beberapa kesalahan klasifikasi antara kelas Covid-19 dan Pneumonia yang memiliki karakteristik visual yang mirip. Data augmentation juga diterapkan untuk meningkatkan kemampuan generalisasi model dengan menggunakan rotasi, *zoom*, dan *horizontal flip*. Hasil akhir menunjukkan bahwa model ANN ini memiliki performa yang cukup baik untuk klasifikasi citra medis sederhana, namun dapat ditingkatkan lebih lanjut melalui penerapan teknik tuning lanjutan seperti *early stopping*, *hyperparameter optimization*, atau penggunaan arsitektur CNN yang lebih sesuai untuk data citra.

Bagian 3: Implementasi CNN untuk Klasifikasi

1. Rancang dan implementasikan model *Convolutional Neural Network* (CNN) untuk klasifikasi objek yang sama

Convolutional Neural Network (CNN) adalah jenis jaringan saraf tiruan yang dirancang khusus untuk pengolahan citra. CNN menggunakan lapisan konvolusi untuk mengekstraksi fitur visual seperti bentuk dan tekstur, serta lapisan *pooling* untuk

mengurangi dimensi data. Berbeda dengan ANN biasa, CNN mempertahankan informasi spasial dalam gambar sehingga lebih efektif untuk tugas klasifikasi citra. Struktur CNN umumnya terdiri dari beberapa lapisan konvolusi, pooling, dan diakhiri dengan *fully connected layer* untuk menghasilkan prediksi.

```
model = Sequential([
    Input(shape=(128, 128, 3)),
    Conv2D(32, (3, 3), activation='relu'),
    MaxPooling2D(2,2),
    Conv2D(64, (3,3), activation='relu'),
    MaxPooling2D(2,2),
    Conv2D(128, (3,3), activation='relu'),
    MaxPooling2D(2,2),
    Flatten(),
    Dense(128, activation='relu'),
    Dropout(0.5),
    Dense(3, activation='softmax') # 3 kelas
])

model.compile(optimizer='adam',
loss='categorical_crossentropy', metrics=['accuracy'])
model.summary()
history = model.fit(
    train_generator,
    epochs=15,
    validation_data=val_generator,
    callbacks=[]
)
```

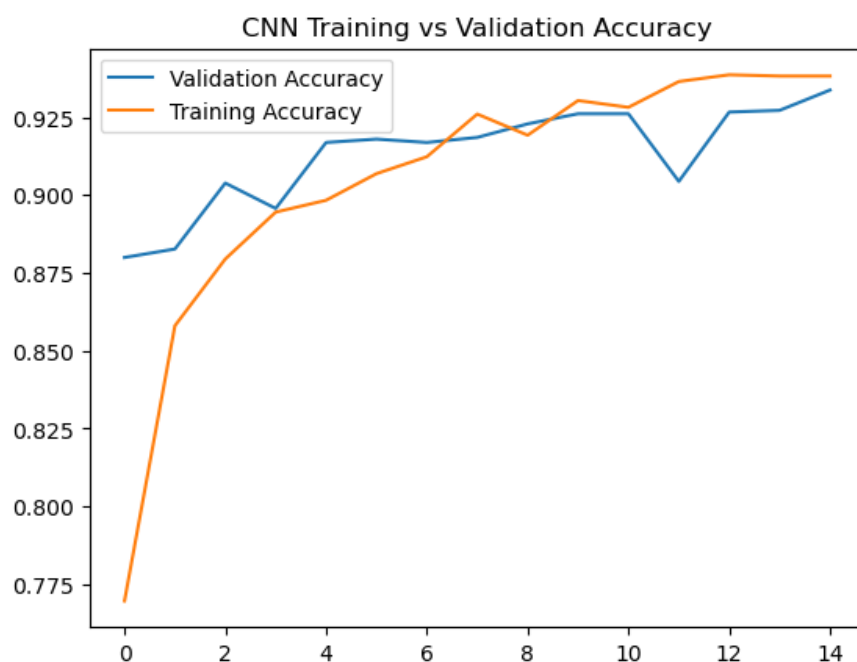
2. Dokumentasikan arsitektur CNN yang digunakan, mencakup

Layer (type)	Output Shape	Param #
conv2d_3 (Conv2D)	(None, 126, 126, 32)	896
max_pooling2d_3 (MaxPooling2D)	(None, 63, 63, 32)	0
conv2d_4 (Conv2D)	(None, 61, 61, 64)	18,496
max_pooling2d_4 (MaxPooling2D)	(None, 30, 30, 64)	0
conv2d_5 (Conv2D)	(None, 28, 28, 128)	73,856
max_pooling2d_5 (MaxPooling2D)	(None, 14, 14, 128)	0
flatten_1 (Flatten)	(None, 25088)	0
dense_2 (Dense)	(None, 128)	3,211,392
dropout_1 (Dropout)	(None, 128)	0
dense_3 (Dense)	(None, 3)	387

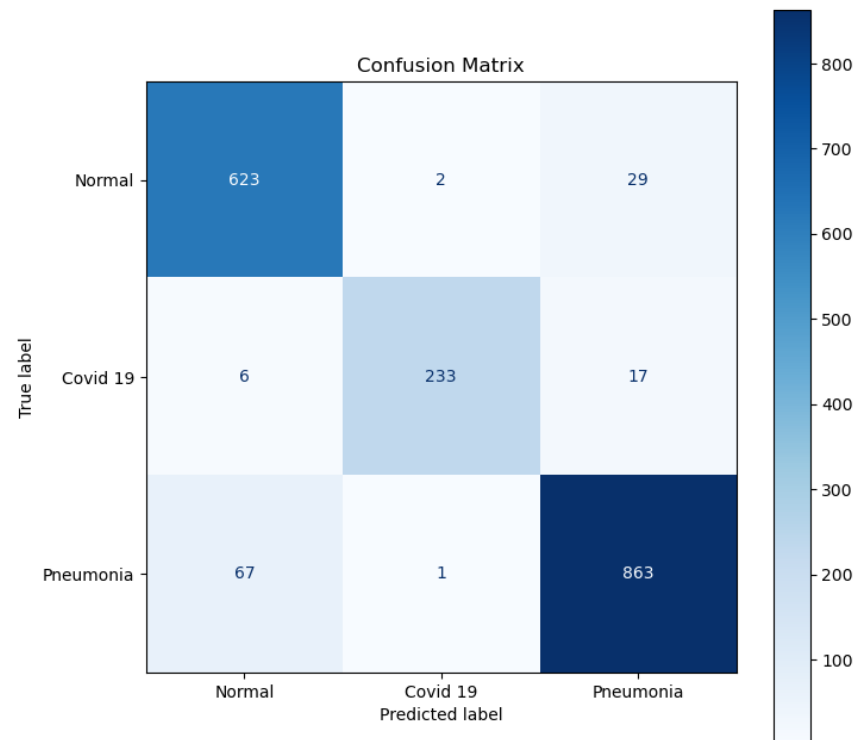
a. Jumlah dan urutan lapisan konvolusi

Model CNN dibentuk dengan 3 lapisan konvolusi (Conv2D) yang masing-masing diikuti oleh lapisan MaxPooling2D, sehingga terdapat 6 lapisan utama sebelum *flatten*

- b. Ukuran filter dan jumlah *feature maps*
- Lapisan pertama: Conv2D dengan *output shape* (126, 126, 32) terdapat 32 *feature maps*
 - Lapisan kedua: Conv2D dengan *output shape* (61, 61, 64) terdapat 64 *feature maps*
 - Lapisan ketiga: Conv2D dengan *output shape* (28, 28, 128) terdapat 128 *feature maps*
- c. *Pooling* yang digunakan
- Digunakan MaxPooling2D setelah Conv2D dengan ukuran *stride* 2, yang mengurangi ukuran fitur sebanyak setengah.
- d. Jumlah *fully connected layers*
- Setelah proses konvolusi dan *flatten*, terdapat:
- *Dense layer* dengan 128 *neuron*
 - *Dropout*
 - *Dense output layer* dengan 3 *layer* (untuk klasifikasi 3 kelas)
- e. Fungsi aktivasi, inisialisasi bobot, dan metode optimasi
- Fungsi aktivasi: ReLu di *layer hidden*, *softmax* di *layer output*
 - Bobot: menggunakan insialisasi *default* Keras
 - *Optimizer*: Adam
 - *Loss function*: *caterogical_crossentropy*
- f. Prosedur training, validasi, dan *testing*
- *Training*: dilakukan selama 15 *epoch* menggunakan data augmentasi untuk generalisasi
 - Validasi: menggunakan generator dari direktori validasi
 - *Testing*: dilakukan pada data validasi, evaluasi dengan *confussion matrix* dan *classification report*
- g. Evaluasi akurasi



- *Training Accuracy* (garis kuning) pada *epoch* pertama adalah 0.6626 dan terus meningkat hingga akurasi tertinggi ada pada *epoch* 14 yaitu 0.9412 dan di *epoch* 15 turun menjadi 0.9409.
- *Validation Accuracy* (garis biru) pada *epoch* pertama adalah 0.8800 dan terus meningkat hingga akurasi tertinggi ada pada *epoch* 15 yaitu 0.9337



Berdasarkan *confusion matrix* tersebut, model CNN baik dalam memprediksi citra *Chest X-Ray*.

- Uji model dengan data baru dan lakukan prediksi terhadap gambar tersebut.
 - Prediksi data yang pertama dilakukan dengan menginputkan citra *Chest X-Ray* paru paru normal, klasifikasi menggunakan model CNN menunjukkan bahwa citra tersebut masuk ke dalam klasifikasi normal dengan akurasi sebesar 0.98.

Prediksi: Normal (0.98)



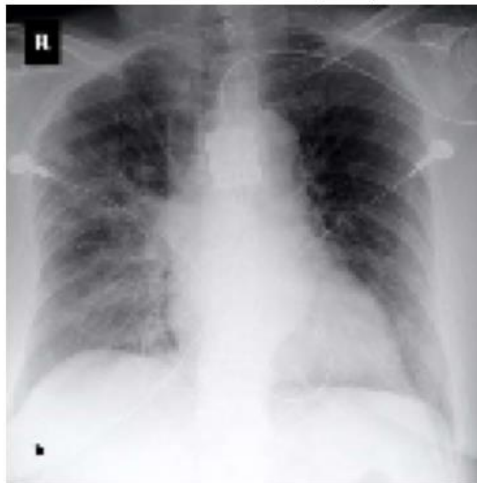
- b. Prediksi daya yang kedua dilakukan dengan memasukkan citra *Chest X-Ray* penderita Pneumonia, klasifikasi menggunakan CNN menunjukkan bahwa citra tersebut masuk ke dalam klasifikasi Pneumonia dengan akurasi sebesar 1.00.

Prediksi: Pneumonia (1.00)



- c. Prediksi daya yang kedua dilakukan dengan memasukkan citra *Chest X-Ray* penderita Covid 19, klasifikasi menggunakan CNN menunjukkan bahwa citra tersebut masuk ke dalam klasifikasi Covid 19 dengan akurasi sebesar 1.00.

Prediksi: Covid 19 (1.00)



4. Tulis analisis tentang performa CNN dan bagaimana proses *tuning* dilakukan untuk mendapatkan akurasi optimal

Model *Convolutional Neural Network* (CNN) yang digunakan dalam penelitian ini dirancang untuk melakukan klasifikasi citra *rontgen* dada ke dalam tiga kategori: Normal, Covid-19, dan Pneumonia. Arsitektur CNN terdiri dari tiga lapisan konvolusi berurutan dengan jumlah filter 32, 64, dan 128, masing-masing berukuran kernel 3×3 dan diikuti oleh lapisan *MaxPooling2D* untuk mereduksi dimensi spasial dan mempertahankan fitur penting. Lapisan-lapisan ini diakhiri dengan proses flattening dan satu lapisan *fully connected* (*Dense*) berukuran 128 *neuron*, yang kemudian disambungkan ke lapisan *output* dengan fungsi aktivasi *softmax* untuk klasifikasi multi-kelas.

Model ini dioptimasi menggunakan algoritma Adam *optimizer* dengan fungsi *loss categorical crossentropy*, yang sesuai untuk tugas klasifikasi tiga kelas. Selama proses pelatihan, dilakukan augmentasi data berupa rotasi, *zoom*, dan *flip horizontal* untuk meningkatkan generalisasi model terhadap data validasi.

Evaluasi performa model dilakukan melalui metrik akurasi serta visualisasi *confusion matrix* dan *classification report*. Hasil pelatihan menunjukkan bahwa model mampu mencapai akurasi yang tinggi, dengan distribusi prediksi yang relatif seimbang antar kelas. Untuk mendapatkan akurasi yang optimal, dilakukan proses *tuning hyperparameter*, meliputi:

- Jumlah filter dan ukuran *layer* pada bagian konvolusi dan *dense*.
- Tingkat *dropout* untuk mencegah *overfitting*.
- *Learning rate*, yang dapat diatur untuk mempercepat konvergensi atau mencegah stagnasi.
- Jumlah *epoch* dan *batch size*, yang dievaluasi berdasarkan kurva pelatihan dan validasi.
- Data *augmentation*, yang diterapkan untuk memperkaya variasi data dan meningkatkan robustitas model.
- *Callback* seperti *EarlyStopping* (jika digunakan) untuk menghentikan pelatihan saat performa validasi tidak meningkat.

Bagian 4: Kesimpulan dan Analisis Perbandingan

1. Perbandingan kinerja model ANN dan CNN:

Aspek	ANN	CNN
Akurasi Validasi	Akurasi 85.98%	Akurasi 94.12%
Waktu Pelatihan	119 menit untuk 15 <i>epoch</i>	67 menit untuk 15 <i>epoch</i>
Kompleksitas Model	Relatif sederhana, hanya <i>dense layer</i>	Lebih kompleks, terdiri dari <i>convolutional layer</i>
Kemampuan Generalisasi	Baik dalam menggeneralisir data baru, namun akurasi untuk prediksi data baru lebih rendah dibanding CNN	Baik dalam mengenali data baru, akurasi untuk prediksi kelas data baru juga tinggi.

2. Jelaskan kelebihan dan kekurangan dari kedua metode untuk permasalahan klasifikasi citra

Berdasarkan tabel perbandingan antara ANN dan CNN, dapat disimpulkan bahwa CNN memiliki keunggulan signifikan dalam hal klasifikasi citra. Meskipun waktu pelatihan CNN lebih singkat (67 menit) dibandingkan ANN (119 menit), CNN

tetap mampu mencapai akurasi validasi yang lebih tinggi, yaitu 94.12%, dibandingkan dengan ANN yang hanya mencapai 85.98%. Hal ini menunjukkan bahwa CNN lebih efisien dan efektif dalam belajar dari data.

Dari sisi kompleksitas, ANN lebih sederhana karena hanya terdiri dari lapisan dense (fully connected), sementara CNN memiliki arsitektur yang lebih kompleks dengan adanya lapisan konvolusi dan pooling yang secara khusus dirancang untuk mengenali pola visual dalam citra. Kompleksitas ini justru menjadi kekuatan CNN dalam mengekstraksi fitur penting dari gambar secara otomatis.

Dalam hal kemampuan generalisasi, CNN juga menunjukkan performa lebih baik dalam mengklasifikasikan data baru. Meskipun ANN cukup baik dalam generalisasi, hasil prediksi terhadap data yang belum pernah dilihat menunjukkan bahwa CNN lebih konsisten dan akurat. Hal ini karena CNN dapat menangkap struktur spasial dalam gambar yang tidak dapat dipahami secara efektif oleh ANN. Oleh karena itu, untuk permasalahan klasifikasi citra, CNN lebih unggul secara menyeluruh dibandingkan ANN.