

# KLASIFIKASI GAMBAR PARU-PARU CT SCAN DENGAN CANNY EDGE DETECTOR

RISET INFORMATIKA



Oleh :

RAIHAN THOBIE N M

20081010169

PROGRAM STUDI INFORMATIKA

FAKULTAS ILMU KOMPUTER

UNIVERSITAS PEMBANGUNAN NASIONAL "VETERAN" JAWA TIMUR

2023

## DAFTAR ISI

# **BAB I**

## **PENDAHULUAN**

### **1.1 Latar Belakang**

Di tengah kemajuan teknologi, penggunaan alat pencitraan medis telah menjadi tonggak penting dalam bidang kedokteran. Teknologi ini memungkinkan kita untuk melakukan pemeriksaan dengan detail tanpa perlu melakukan prosedur bedah yang invasif. Namun, dalam penggunaannya, saya menyadari adanya perbedaan pendekatan antar dokter dalam menganalisis citra medis yang dihasilkan.

Perbedaan ini sering kali menyebabkan interpretasi yang berbeda, yang pada akhirnya dapat mengganggu akurasi diagnosis suatu kondisi medis. Hal ini merupakan tantangan yang signifikan dalam bidang kesehatan. Oleh karena itu, perlu adanya upaya untuk meningkatkan keakuratan diagnosis, terutama dengan memanfaatkan teknologi komputer dalam sistem pencitraan medis.

Teknik klasifikasi citra medis menjadi salah satu pendekatan yang menjanjikan dalam membantu menganalisis gambar medis. Teknik ini memungkinkan identifikasi bagian tubuh yang terkena dampak penyakit atau kelainan. Namun, untuk dapat meningkatkan akurasi klasifikasi, penting untuk melakukan proses ekstraksi informasi yang tepat dari citra medis.

Salah satu aspek yang menjadi fokus utama penelitian ini adalah pendeteksian tepi citra menggunakan metode Canny edge detector. Saya tertarik untuk mendalami teknik ini dan mengevaluasi sejauh mana metode pendeteksian tepi citra ini dapat meningkatkan akurasi diagnosis dalam pencitraan medis.

Penelitian saya ini diharapkan dapat memberikan kontribusi yang bermanfaat dalam memperkaya pemahaman kita terhadap penggunaan teknologi dalam diagnosis medis. Saya berharap hasil dari penelitian ini dapat menjadi landasan bagi pengembangan sistem yang lebih handal, yang pada akhirnya akan memberikan manfaat besar bagi peningkatan standar perawatan kesehatan dan diagnosis yang lebih tepat bagi pasien.

## **1.2 Rumusan Masalah**

Berdasarkan latar belakang yang telah diuraikan, maka perumusan masalah yang dapat diambil adalah sebagai berikut :

1. Bagaimana mengoptimalkan pendeteksian dan analisis citra dari hasil CT scan paru-paru dengan menggunakan canny edge detector untuk mengidentifikasi area yang terkena kelainan atau penyakit?
2. Apa metode atau algoritma pendeteksian citra yang paling efektif dalam mengidentifikasi pola atau ciri khas dari CT scan paru-paru untuk mendukung diagnosis kondisi kesehatan tertentu?
3. Sejauh mana keakuratan metode pendeteksian citra dari CT scan paru-paru menggunakan teknologi pemrosesan citra untuk membedakan antara area yang sehat dan area yang terkena penyakit atau kelainan?
4. Apa pendeteksian citra dari hasil CT scan paru-paru dapat menghasilkan kontribusi yang signifikan untuk diagnosis media?

## **1.3 Tujuan**

Berdasarkan rumusan masalah diatas, maka tujuan dalam penelitian ini adalah sebagai berikut :

1. Mengkaji dan membandingkan berbagai metode pendeteksian citra, terutama teknik pendeteksian tepi (edge detection), guna menentukan metode yang paling efektif dan akurat dalam mengidentifikasi kelainan atau penyakit pada hasil CT scan paru-paru.
2. Mengembangkan atau menyesuaikan algoritma pendeteksian citra yang tepat untuk membedakan pola atau ciri khas dari gambaran CT scan paru-paru yang mencerminkan berbagai kondisi kesehatan, seperti pneumonia, tumor, atau penyakit lainnya.
3. Mengevaluasi keakuratan, ketepatan, dan reliabilitas dari metode pendeteksian citra yang dipilih, serta mengukur sejauh mana teknologi pemrosesan citra mampu membedakan secara akurat antara area yang sehat dan yang terkena penyakit pada CT scan paru-paru.

4. Menghasilkan kontribusi signifikan dalam pengembangan teknologi pencitraan medis dengan fokus pada pendeteksian citra dari hasil CT scan paru-paru, guna mendukung diagnosis dini, pemantauan, dan perawatan yang lebih efektif terhadap penyakit atau kelainan pada paru-paru.

#### **1.4 Manfaat**

Melalui penelitian ini diharapkan dapat memberi manfaat sebagai berikut :

1. Memberikan kontribusi dalam meningkatkan akurasi diagnosis kondisi paru-paru untuk identifikasi dini serta pemantauan yang lebih akurat terhadap penyakit seperti pneumonia, tumor, atau kondisi lainnya.
2. Metode pendeteksian citra yang efektif dapat meningkatkan efisiensi di lingkungan klinis, memungkinkan dokter untuk membuat keputusan diagnosis dengan lebih cepat dan akurat.
3. Dengan diagnosis yang lebih akurat, membuat rencana perawatan yang lebih tepat dan tindak lanjut yang lebih terarah, memungkinkan intervensi medis yang lebih dini dan lebih tepat sasaran.

#### **1.5 Batasan Masalah**

Batasan masalah yang terdapat pada penelitian ini antara lain :

1. Dataset berupa kumpulan citra yang diambil dari sumber open source yaitu Kaggle dengan nama “Pneumonia X-ray Images”, yang terdiri dari 234 paru-paru normal dan 390 paru-paru terdampak Pneumonia.
2. Data citra x-ray yang digunakan sudah ditentukan jenisnya yaitu normal dan terdampak Pneumonia.
3. Penelitian ini hanya melakukan pengklasifikasian terhadap citra paru-paru terdampak Pneumonia dan normal.
4. Penelitian ini menggunakan metode deteksi tepi “Canny Edge Detector”.
5. Keluaran yang dihasilkan pada penelitian ini berupa hasil akurasi ketepatan klasifikasi paru-paru terdampak Pneumonia dan normal.

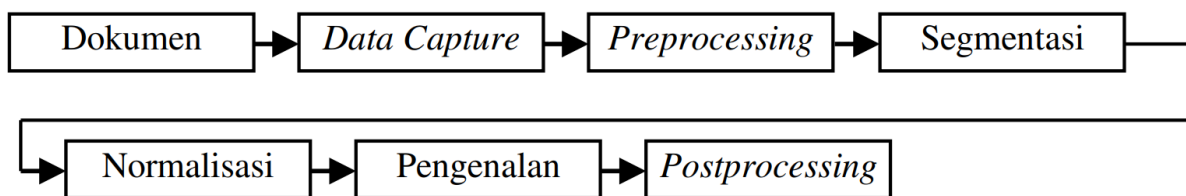
## BAB II

### TINJAUAN PUSTAKA

#### 2.1 Pencitraan Digital

Citra digital merupakan representasi gambar dua dimensi yang berasal dari citra analog dua dimensi yang kontinu melalui proses pengambilan sampel. Citra ini dapat diolah oleh komputer dan terdiri dari larik nilai-nilai kompleks yang direpresentasikan dalam serangkaian bit tertentu. Pengolahan citra digital merupakan bidang ilmu yang mempelajari cara meningkatkan kualitas suatu citra, melakukan transformasi citra, mengidentifikasi fitur citra yang optimal untuk analisis, menyimpan data dengan kompresi, melakukan transmisi data, serta mengatur waktu proses data.

Dalam prosesnya sendiri, pencitraan digital dapat digambarkan secara sederhana seperti gambar dibawah



*gambar 1.1 proses pengolahan citra dengan pola*

Sebuah citra digital dapat mewakili suatu matriks dengan M kolom dan N baris, di mana setiap perpotongan kolom dan baris dikenal sebagai piksel. Piksel merupakan unit terkecil dari sebuah citra dan memiliki dua parameter utama: intensitas dan koordinat. Nilai yang terkait dengan koordinat (x, y) dalam citra adalah  $f(x, y)$ , yaitu nilai intensitas atau warna dari piksel yang terletak pada titik tersebut.

$$f(x, y) = \begin{bmatrix} f(1,1) & f(1,2) & \cdots & f(1,N) \\ f(2,1) & f(2,2) & \cdots & f(2,N) \\ \vdots & \vdots & \vdots & \vdots \\ f(M, 1) & f(M, 2) & \cdots & f(M, N) \end{bmatrix}$$

*gambar 2.2 Contoh rumus*

Berdasarkan rumus pada gambar 1.2, maka secara matematis citra digital dapat dituliskan sebagai intensitas  $f(x,y)$  dimana nilai x (baris) dan y (kolom) merupakan koordinat posisi serta  $f(x,y)$  merupakan nilai fungsi pada setiap titik (x,y) yang menyatakan besar intensitas yang diterima oleh citra baik tingkat keabuan atau warna dari piksel di titik tersebut. Besarnya nilai tergantung pada intensitas yang dipantulkan oleh objek. Hal ini berarti  $f(x,y)$  sebanding dengan energi yang dipancarkan oleh sumber cahaya.

## **2.2 Machine Learning**

Machine Learning (ML) adalah kombinasi dari beragam algoritma matematika dan penerapan komputasional yang mengadopsi proses pembelajaran dari data untuk memungkinkan komputer membuat prediksi serta mengambil keputusan di masa depan. Proses pembelajaran ini terdiri dari dua tahap utama, yaitu tahap pelatihan (training) dan tahap pengujian (testing). Dalam dunia Machine Learning, terdapat beberapa masalah yang sering dihadapi dan diselesaikan menggunakan berbagai pendekatan, di antaranya adalah klasterisasi dan klasifikasi. Klasterisasi bertujuan untuk mengelompokkan data berdasarkan kemiripan fitur yang dimilikinya, sehingga data-data yang serupa dikelompokkan bersama. Sementara itu, klasifikasi bertujuan untuk mengelompokkan data ke dalam kelas-kelas atau kategori tertentu berdasarkan ciri-ciri atau karakteristik tertentu yang telah ditentukan sebelumnya. Perbedaan mendasar antara kedua permasalahan ini adalah bahwa klasterisasi tidak memerlukan adanya label atau klasifikasi sebelumnya pada data, sementara klasifikasi membutuhkan penentuan label-label tertentu sebagai target pengelompokan.

## **2.3 Canny Edge**

Algoritma deteksi tepi Canny merupakan salah satu pendekatan yang umum digunakan dalam pengolahan citra untuk menemukan tepi objek dengan akurasi tinggi dan pengurangan noise yang optimal. Dikembangkan oleh John F. Canny pada tahun 1986, algoritma ini telah menjadi salah satu pendekatan yang paling banyak diadopsi dalam identifikasi tepi pada citra. Metode Canny menggabungkan beberapa langkah dalam prosesnya. Pertama-tama, langkah pertama adalah pengurangan noise dengan menggunakan filter Gaussian untuk menghilangkan gangguan kecil atau noise pada citra. Langkah ini bertujuan untuk membuat citra lebih bersih sebelum proses deteksi tepi. Langkah berikutnya adalah menemukan gradien citra dengan menggunakan operasi Sobel atau metode lainnya untuk mengidentifikasi perubahan intensitas yang signifikan antara piksel-piksel di dalam citra. Ini membantu dalam menentukan di mana tepi yang sebenarnya terletak.

Selanjutnya, dengan menggunakan teknik non-maximum suppression, algoritma Canny memastikan bahwa tepi yang diidentifikasi adalah tepi yang tepat, dengan menyaring tepi-tepi yang bukan merupakan bagian dari kontur objek yang sebenarnya. Tahap terakhir dalam algoritma ini adalah penggunaan hysteresis thresholding, di mana dua nilai ambang (threshold) digunakan untuk mengklasifikasikan piksel-piksel tepi menjadi tepi yang kuat dan tepi yang lemah, memungkinkan untuk menghubungkan piksel-piksel tepi yang kuat untuk membentuk kontur tepi yang utuh. Algoritma deteksi tepi Canny telah menjadi standar dalam banyak aplikasi pengolahan citra, terutama dalam bidang penglihatan komputer, pengenalan pola, dan analisis citra medis. Keunggulan utamanya terletak pada kemampuannya dalam menghasilkan tepi yang tajam, akurat, serta mampu mengatasi noise dengan baik, sehingga menjadikannya salah satu metode yang paling diandalkan untuk deteksi tepi dalam berbagai aplikasi pengolahan citra.

## **2.4 Convolutional Neural Network**

Sejak diperkenalkannya deep learning, berbagai ide telah muncul terkait metode pembelajarannya, salah satunya adalah convolutional neural networks (CNN). Terinspirasi dari penelitian David Hubel dan Torsten Wiesel tentang visual cortex pada tahun 1962, jaringan CNN pertama kali diperkenalkan pada tahun 1990-an. Model awal ini dikembangkan untuk

mengenali tulisan tangan dan angka, menunjukkan hasil kinerja yang menjanjikan (LeCun, Bottou, Bengio, & Haffner, 1998). Pada tahun 2012, perhatian yang besar terhadap jaringan ini muncul di kalangan peneliti setelah Alex Krizhevsky meraih kemenangan dalam kompetisi "ImageNet Large Scale Visual Recognition Challenge". Ia berhasil menggunakan model CNN yang dikenal sebagai "AlexNet" untuk mengklasifikasi lebih dari 1,2 juta citra beresolusi tinggi dalam 1000 kategori, dengan tingkat kesalahan sebesar 16,4%. Keberhasilan ini membuktikan bahwa pendekatan CNN mampu mengungguli metode Machine Learning lainnya dalam hal klasifikasi objek pada citra (Lopez Pinaya et al., 2019).

Convolutional Neural Network (CNN) merupakan salah satu algoritma dalam deep learning yang merupakan perluasan dari konsep Multilayer Perceptron (MLP) dan termasuk dalam jenis neural network feed forward. Dirancang khusus untuk mengelola data dua dimensi, CNN tergolong dalam Deep Neural Network karena memiliki struktur yang dalam dan sering digunakan dalam pengolahan data citra. Secara teknis, CNN adalah suatu arsitektur yang terlatih dan terdiri dari beberapa tahap, yaitu tahap masukan (input) dan keluaran (output) (Ilahiyah & Nilogiri, 2018).

Secara garis besar, metode CNN tidak jauh berbeda dari neural network lainnya seperti backpropagation dan perceptron. CNN terdiri dari neuron yang memiliki bobot, bias, dan fungsi aktivasi (Nugroho et al., 2020). Dalam struktur CNN, terdapat beberapa lapisan yang mencakup convolution layer, fungsi aktivasi, pooling layer, lapisan flatten, dan lapisan fully connected (Handono et al., 2020).

Arsitektur CNN memiliki tahapan yang penting, seperti lapisan konvolusi yang bertugas untuk mengekstraksi fitur-fitur penting dari citra dengan menggunakan filter, fungsi aktivasi untuk menambahkan non-linearitas, lapisan pooling untuk mengurangi dimensi spasial dari hasil konvolusi, lapisan flatten untuk mengubah data menjadi bentuk satu dimensi, dan lapisan fully connected yang bertugas untuk menghubungkan hasil dari lapisan sebelumnya ke dalam jaringan yang sepenuhnya terhubung (Ilahiyah & Nilogiri, 2018; Nugroho et al., 2020; Handono et al., 2020). Struktur yang kompleks ini memungkinkan CNN untuk belajar representasi hierarkis dari data citra dengan cara yang efektif.

## **2.5 Optimizer**

Optimizer adalah komponen penting dalam proses pelatihan neural network yang bertanggung jawab untuk menemukan bobot optimal, mengurangi tingkat kesalahan, dan meningkatkan tingkat akurasi model. Saat melatih model, parameter seperti bobot diubah untuk meminimalkan fungsi kesalahan sehingga prediksi yang dihasilkan semakin akurat (Zakiya et al., 2021).

Learning rate, yang merupakan bagian dari proses optimasi, membantu optimizer menentukan seberapa besar koreksi yang diperlukan pada nilai bobot saat melatih data. Rentang nilai learning rate biasanya dari nol hingga satu. Semakin besar nilai learning rate, semakin cepat proses pelatihan dilakukan. Namun, ada konsekuensi dari penggunaan nilai learning rate yang besar yaitu optimizernya dapat menjadi kurang tepat dalam menemukan solusi yang optimal dan akurat.

Berikut beberapa penjelasan optimizer yang umum digunakan dalam proses pelatihan neural network antara lain:



**1. Adaptive Moment (ADAM):**

Adaptive Moment Optimization merupakan salah satu optimasi yang menggabungkan antara metode rmsprop dengan momentum. Adam adalah algoritma optimasi berbasis gradien orde pertama dimana efisien secara komputasi membutuhkan memori yang rendah.

**2. Stochastic Gradient Descent (SGD):**

Optimizer dasar yang menggunakan metode turunan fungsi kesalahan terhadap bobot untuk menemukan nilai bobot yang lebih optimal. Tujuan optimizer ini adalah untuk menemukan parameter yang dapat meminimalisir loss.

**3. Root Mean Square Propagation (RMSProp):**

Root Mean Square Propagation (RMSProp) adalah optimizer yang memanfaatkan besarnya gradient terbaru untuk melakukan normalisasi gradient sehingga terjaga di atas gradient Root Mean Square. Keunggulan RMSProp salah satunya adalah dapat menangani tujuan stokastik dengan baik

**4. AdaGrad:**

AdaGrad adalah salah optimizer ini merupakan salah satu varian dari Stochastic Gradient Descent. Kecepatan pembelajaran dalam optimizer ini dapat disesuaikan dengan seberapa sering parameter diperbarui pada saat proses pelatihan.

## **2.6 Confusion Matrix**

Confusion matrix adalah sebuah metode yang digunakan untuk mengevaluasi kinerja dari suatu model dalam data mining atau pembelajaran mesin. Tujuan dari confusion matrix adalah untuk memberikan informasi mengenai tingkat akurasi klasifikasi yang dilakukan oleh suatu sistem. Confusion matrix terdiri dari tabel yang memuat empat nilai utama yang menyediakan informasi tentang hasil klasifikasi yang dilakukan oleh sistem.

Keempat nilai yang terdapat dalam confusion matrix adalah:

- **True Positive (TP):** Menunjukkan jumlah sampel positif yang terklasifikasi dengan benar sebagai positif oleh model.
- **True Negative (TN):** Menunjukkan jumlah sampel negatif yang terklasifikasi dengan benar sebagai negatif oleh model.
- **False Positive (FP):** Merupakan jumlah sampel negatif yang keliru diklasifikasikan sebagai positif oleh model.
- **False Negative (FN):** Merupakan jumlah sampel positif yang keliru diklasifikasikan sebagai negatif oleh model.

Dengan menggunakan kombinasi nilai-nilai ini, kita dapat menghitung berbagai metrik evaluasi seperti tingkat akurasi (accuracy), presisi (precision), recall (recall/sensitivity), serta f1-score yang memberikan gambaran lebih detail tentang performa model klasifikasi.

Confusion matrix memberikan wawasan yang jelas tentang seberapa baik atau buruknya performa model dalam melakukan klasifikasi pada data. Informasi yang terdapat di dalamnya dapat membantu para praktisi dan peneliti untuk memahami secara lebih mendalam bagaimana model tersebut melakukan prediksi dan di mana letak kelemahan atau keunggulan dari model tersebut (Gunawan et al., 2018).

## **2.7 Python**

Python adalah bahasa pemrograman serbaguna yang bersifat interpretatif, dirancang dengan fokus pada tingkat keterbacaan kode. Bahasa pemrograman ini diklaim memiliki kombinasi unik antara kemampuan yang luas dan sintaksis kode yang sangat jelas. Kemudahan dalam belajar Python terletak pada keserupaan struktur syntax-nya dengan bahasa manusia, memudahkan dalam membaca dan memahami kode program.

Salah satu keunggulan utama Python adalah kemampuannya mendukung pemrograman multiparadigm, termasuk:

1. Pemrograman berorientasi objek: Python mendukung paradigma pemrograman berorientasi objek, memungkinkan penggunaan konsep seperti pewarisan (inheritance), polimorfisme, dan enkapsulasi.
2. Pemrograman imperatif: Python juga mendukung paradigma pemrograman imperatif yang berfokus pada perintah-perintah dan perubahan keadaan program.
3. Pemrograman fungsional: Bahasa ini juga menyediakan fitur-fitur pemrograman fungsional seperti lambda functions, map, reduce, dan filter.

Salah satu kelebihan yang dimiliki Python adalah ketersediaan modul-modul library yang melimpah, memperkaya fungsionalitasnya dan menciptakan komunitas pengguna yang luas. Dengan banyaknya modul library ini, pengguna Python dapat dengan mudah mengakses berbagai fungsi dan algoritma yang sudah tersedia.

Selain itu, Python juga dikenal karena kemampuannya berjalan di berbagai platform sistem operasi, seperti Linux, Windows, dan macOS. Hal ini membuat Python menjadi pilihan utama dalam pengembangan perangkat lunak lintas platform (Syahrudin & Kurniawan, 2018).

## **2.8 Keras**

Keras adalah sebuah application programming interface (API) yang ditujukan untuk jaringan saraf, dirancang untuk beroperasi di atas platform Tensorflow dan Theano. Sebagai salah satu library yang tersedia dalam lingkungan Python, Keras memberikan kekuatan dan fleksibilitas bagi pengguna dalam melakukan pengujian terhadap jaringan saraf. Salah satu karakteristik utama dari Keras adalah penyediaan fungsi-fungsi yang mudah dipahami oleh pengguna, sambil memberikan beragam pilihan untuk mengoptimalkan jaringan.

Keras mengadopsi konsep modularitas yang memungkinkan pengguna untuk secara efektif mengatur modul-modul yang terkait untuk membangun model jaringan saraf. Model yang sudah dibuat dapat dengan efisien dilatih dan diuji dengan menggunakan Keras. Setiap fungsi yang disediakan oleh modul Keras dapat diatur dan disesuaikan sesuai dengan kebutuhan pengguna, seperti pemilihan fungsi aktivasi, fungsi loss, optimizer, dan berbagai parameter lainnya (Negi et al., 2017). Keandalan Keras dalam menyediakan fleksibilitas dan kemudahan konfigurasi ini membuatnya menjadi salah satu pilihan yang populer bagi para praktisi dan peneliti dalam pengembangan dan pengujian model jaringan saraf.

## **2.9 TensorFlow**

TensorFlow adalah salah satu modul library atau framework yang digunakan dalam bidang machine learning dan penelitian jaringan saraf tiruan. Sebagai sebuah library open source, TensorFlow didukung oleh tim pengembang dari Google, yang berkomitmen untuk mengembangkan riset berbasis kecerdasan buatan. TensorFlow telah menjadi salah satu library yang paling populer dalam bidang ilmu data (data science), dengan komunitas pengembang yang besar dan aktif.

Beberapa fitur utama yang dimiliki oleh TensorFlow antara lain:

1. **Skalabilitas Komputasi Tinggi:**  
TensorFlow dirancang untuk menghadapi perhitungan yang berskala besar, mampu menangani komputasi di seluruh mesin dan kumpulan data yang besar. Hal ini membuatnya cocok untuk menangani tugas-tugas yang memerlukan perhitungan pada skala yang luas.
2. **Kemampuan Menjalankan pada CPU dan GPU:**  
TensorFlow dapat menjalankan program dengan baik pada Central Processing Unit (CPU) maupun Graphics Processing Unit (GPU). Penggunaan GPU dapat memberikan kecepatan komputasi yang lebih tinggi dalam banyak kasus, terutama dalam pemrosesan data yang besar.
3. **Definisi, Optimisasi, dan Komputasi Efisien dengan Tensor:** TensorFlow mengutamakan operasi pada tensor, yaitu struktur data multidimensi. Ini memungkinkan TensorFlow untuk mendefinisikan, mengoptimalkan, dan melakukan komputasi secara efisien yang melibatkan array multidimensi tersebut, memudahkan dalam pemrosesan data dalam bentuk tensor.

## **2.10 Scikit-Learn**

Scikit-Learn, atau yang sering disebut sebagai sklearn, merupakan salah satu library yang sangat populer dan sering digunakan dalam lingkungan machine learning di bahasa pemrograman Python. Scikit-Learn menonjol dengan beberapa keunggulan yang membuatnya menjadi pilihan utama di kalangan para praktisi machine learning:

1. **Kecepatan dalam Pemrosesan Dataset:**  
Scikit-Learn dikenal karena efisiensinya dalam melakukan berbagai tugas pemrosesan dataset. Algoritma dan fungsi-fungsinya telah dioptimalkan dengan baik, memberikan kinerja yang cepat dalam pemrosesan dataset yang besar, yang merupakan hal yang penting dalam machine learning.

## 2. Beragam Pilihan Algoritma:

Library ini menyediakan berbagai pilihan dari algoritma yang dapat digunakan untuk berbagai jenis tugas dalam machine learning. Mulai dari regresi untuk memprediksi nilai berkelanjutan, klasifikasi untuk mengkategorikan data, hingga algoritma clustering untuk mengelompokkan data berdasarkan kesamaan karakteristiknya.

Scikit-Learn juga dikenal karena kemudahan penggunaannya, dokumentasinya yang baik, dan integrasi yang baik dengan lingkungan pemrograman Python. Hal ini membuatnya menjadi pilihan yang sangat disukai bagi para pengguna, terutama bagi mereka yang baru dalam dunia machine learning maupun yang sudah berpengalaman. Keunggulan-keunggulan ini menjadikan Scikit-Learn sebagai salah satu library yang paling sering direkomendasikan untuk digunakan dalam mengimplementasikan berbagai teknik machine learning.

## BAB III

### METODOLOGI PENELITIAN

#### 3.1 Tahapan Penelitian

diperlukan langkah-langkah kerja yang sistematis yang berguna untuk memudahkan dalam menyelesaikan penelitian ini dengan baik, sehingga dapat menjawab rumusan masalah dan mencapai tujuan penelitian yang telah ditentukan pada bab sebelumnya.

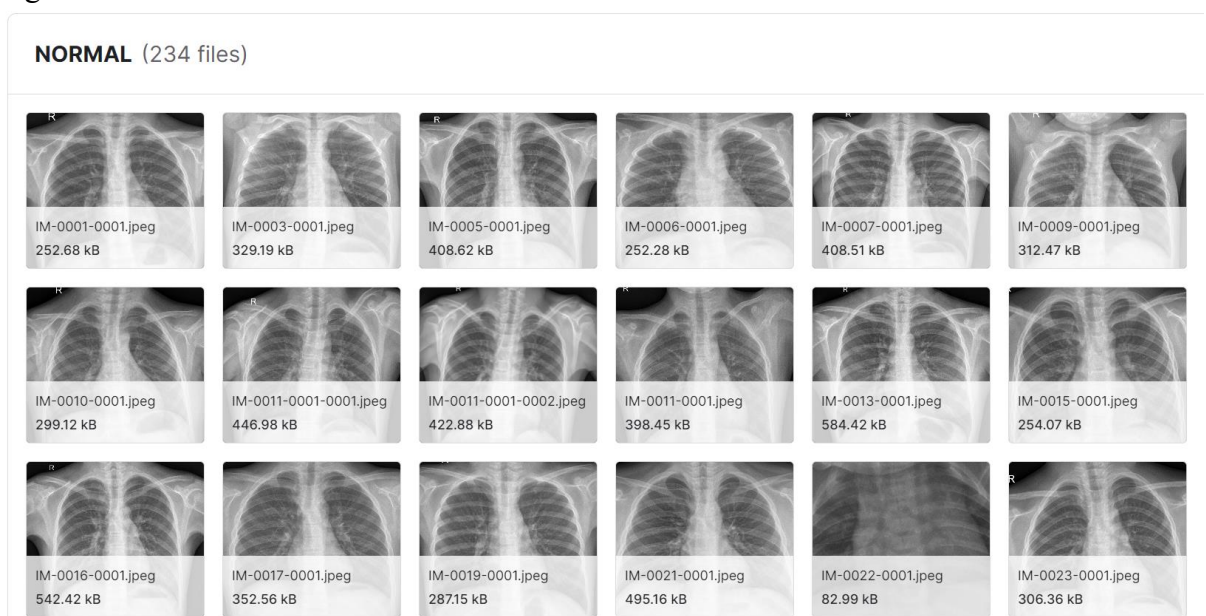
Berikut merupakan langkah-langkah yang akan dilakukan dalam penelitian :

1. Pengumpulan data,
2. Pre-processing data,,
3. Segmentasi data
4. Ekstraksi fitur, dan
5. Klasifikasi data.

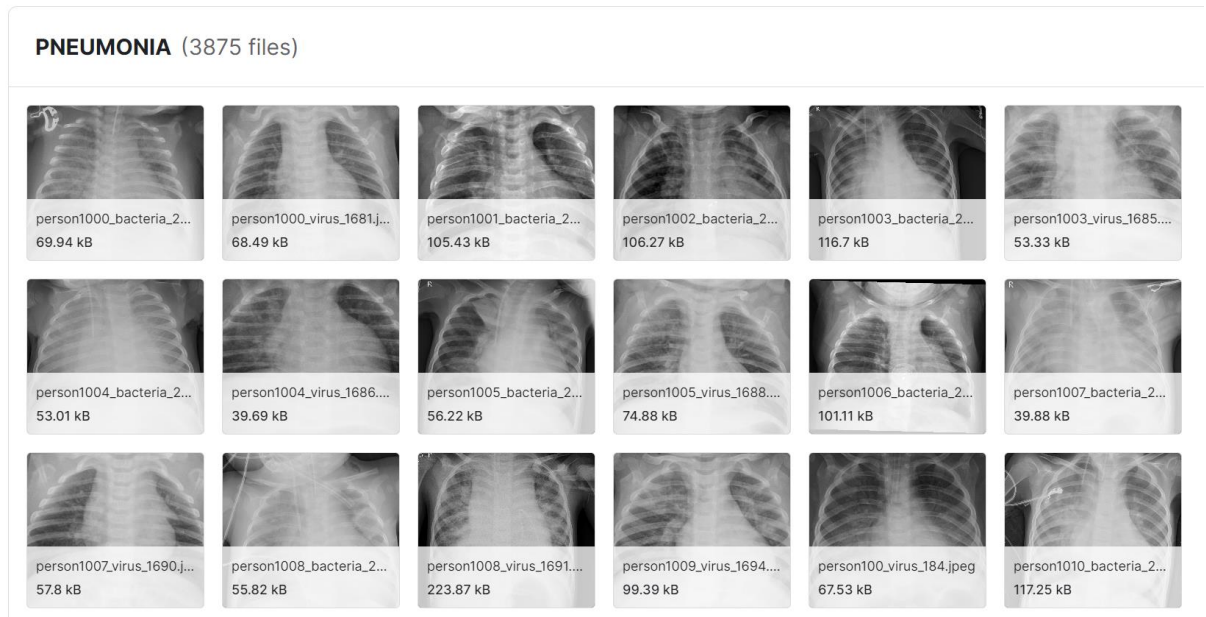
Semua tahapan ini akan dijalankan menggunakan phyton pada Google Collab.

#### 3.2 Pengumpulan Data

Pada penelitian ini penulis menggunakan dataset melalui website open source Kaggle dengan nama “Chest X-Ray Images (Pneumonia)” yang diunggah oleh Paul Mooney, dataset tersebut dapat digunakan oleh siapapun atau bersifat umum. Data yang digunakan merupakan data sekunder dari citra X-ray dada manusia, total keseluruhan data yang didapat berjumlah 5863 citra dengan format file JPG. Dataset tersebut terbagi menjadi 2 kelas yaitu citra paru-paru yang terkena Pneumonia dan citra paru-paru normal. Berikut merupakan contoh citra dari dataset yang digunakan:



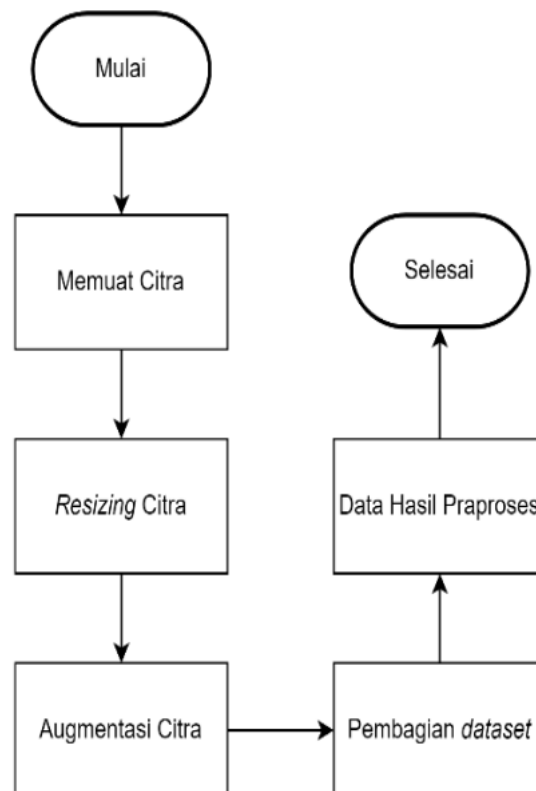
Pada gambar diatas dapat dilihat bahwa hasil foto rontgen pada paru-paru normal terlihat dengan jelas, tidak terdapat area yang lebih terang atau bercak abu-abu. Sedangkan pada paru-paru yang terdampak Pneumonia akan terdapat area yang lebih terang, seperti yang terlihat pada citra berikut:



Dalam dataset ini ditemukan ukuran citra yang berbeda-beda sehingga memerlukan tahapan pre-processing data. Tahap pre-processing data dilakukan sebelum pelatihan yang digunakan untuk mendapatkan hasil akhir yang lebih baik.

### 3.3 Pre-Processing Data

Sebelum masuk pada proses klasifikasi citra, seluruh citra akan diproses terlebih dahulu dalam tahap praproses data. Berikut merupakan tahapan pre-processing yang telah disusun :



Pada dataset “Chest X-Ray Images (Pneumonia)” terdapat berbagai macam ukuran citra sehingga diperlukan praproses pada dataset tersebut, oleh karena nya

langkah pertama yang dilakukan pada tahap pre-processing data adalah dengan memuat dataset citra yang terdapat dalam google drive kemudian mengubah semua ukuran data atau resizing citra menjadi ukuran yang sama yaitu dimana ukuran tersebut sama atau sesuai dengan ukuran input.

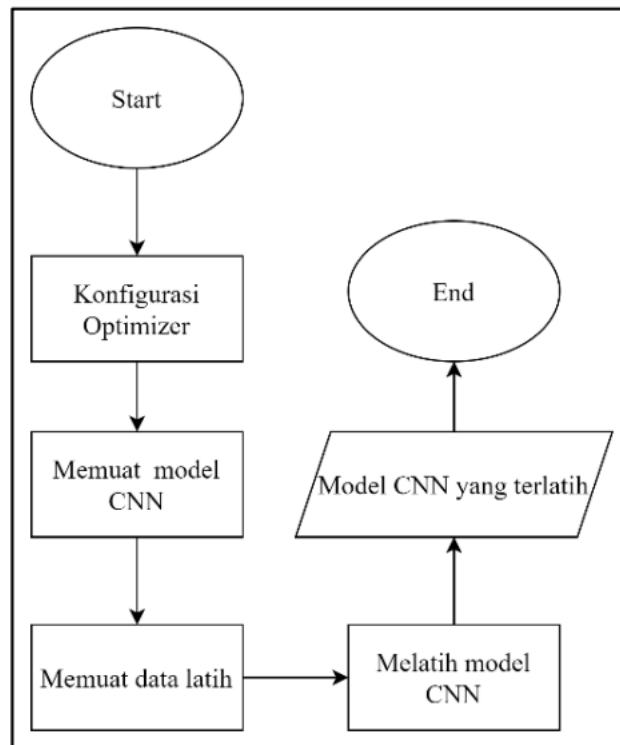
Setelah melalui tahap resizing, tahap selanjutnya adalah proses augmentasi data. Augmentasi data adalah strategi yang digunakan untuk meningkatkan variasi data secara signifikan yang tersedia untuk melatih model tanpa benar-benar mengumpulkan data baru. Ada beberapa teknik umum yang digunakan dalam augmentasi data seperti rotasi pada citra sebesar  $x$  derajat, flipping atau memutar citra secara horizontal maupun vertical, padding, cropping atau pemotongan pada pixel citra dan juga modifikasi kecerahan citra atau saturation. Namun pada penelitian ini hanya menggunakan beberapa teknik augmentasi data yaitu rotasi pada citra. Flipping, dan rescale.

1. Rotasi adalah salah satu teknik augmentasi dengan cara 35 melakukan pemutaran posisi citra sebesar  $x$  derajat. Citra akan dilakukan rotasi augmentasi dengan maksimal rotasi sebesar 15 derajat.
2. Flipping adalah salah satu teknik dalam augmentasi yang digunakan untuk melakukan pembalikan pada nilai piksel citra secara vertikal ataupun horizontal. Pada penelitian ini jenis augmentasi flipping yang digunakan adalah horizontal flipping.
3. Rescale adalah proses untuk mengubah rentang nilai piksel dalam citra ke rentang yang diinginkan atau lebih sesuai untuk analisis atau pengolahan lanjutan. Tujuan utamanya adalah untuk menyesuaikan atau menormalkan rentang nilai piksel agar cocok dengan kebutuhan tertentu, seperti memastikan bahwa nilai piksel berada dalam skala yang diinginkan atau sesuai dengan persyaratan model yang akan digunakan.

Setelah melalui proses augmentasi, akan dilakukan proses Splitting atau pembagian terhadap dataset yang digunakan. Sebelum melakukan proses tersebut akan dilakukan suatu proses yaitu konversi dataset menjadi dataframe agar data tersebut dapat diolah didalam proses splitting. Data yang telah diproses dalam splitting akan terbagi menjadi dua yaitu data latih dan data tes.

### **3.4 Pelatihan Model**

Dalam sistem klasifikasi deteksi citra x-ray paru pneumonia atau normal ini untuk mendapatkan nilai akurasi yang tinggi maka pada model CNN yang telah dibuat perlu dilakukan pelatihan dengan data training yang telah disiapkan atau dilakukan pre-processing pada tahap sebelumnya, hal tersebut bertujuan untuk melatih sistem dalam menemukan fitur-fitur atau ciri-ciri yang terdapat dari setiap objek pada citra. Untuk melakukan pelatihan pada model diperlukan beberapa tahapan, tahap-tahap tersebut dapat dilihat pada gambar berikut.



### 3.5 Konfigurasi Optimizer

Pada penelitian ini penerapan optimizer memanfaatkan library dari tensorflow dengan menggunakan 1 jenis optimizer yaitu Adam

Adam optimizer adalah metode stochastic gradient descent yang didasarkan pada estimasi adaptif dari momen orde pertama dan orde kedua untuk memperbarui bobot. Pada optimizer adam nilai dari learning rate akan dipertahankan untuk setiap parameter dan akan diadaptasi secara terpisah saat proses pembelajaran. Adam optimizer adalah metode stochastic gradient descent yang didasarkan pada estimasi adaptif dari momen orde pertama dan orde kedua untuk memperbarui bobot. Pada optimizer adam nilai dari learning rate akan dipertahankan untuk setiap parameter dan akan diadaptasi secara terpisah saat proses pembelajaran.

### 3.6 Memuat Model CNN

Setelah melakukan konfigurasi optimizer tahap selanjutnya pada pelatihan model adalah dengan memuat model arsitektur CNN yang telah dibuat, model ini akan digunakan pada seluruh tahapan proses pelatihan. Setelah model dimuat maka akan dilakukan compile, pada saat melakukan compiling pada model ada beberapa hal yang harus di setting yaitu mengatur optimizer yang telah disesuaikan nilainya pada tahap sebelumnya serta mengatur loss function yang akan digunakan.



### 3.7 Memuat Data latih

Setelah melakukan konfigurasi pada optimizer, proses berikutnya adalah melakukan pemuatan pada data latih, data latih yang digunakan adalah dataset yang sudah melalui tahap preprocessing dan segmentasi.

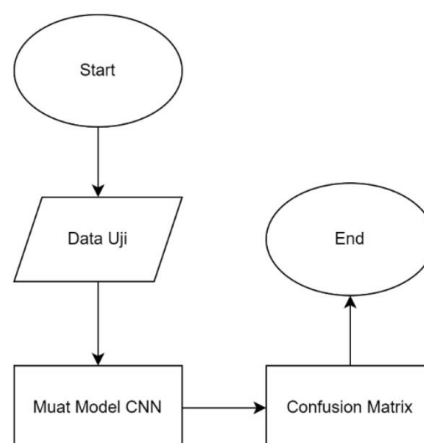
### 3.8 Melatih Model CNN

Tahap ini merupakan langkah terakhir dalam proses pelatihan model CNN. Pada tahap pelatihan ini akan menggunakan fungsi .fit yang didapat dari penggunaan library tensorflow. Didalam fungsi .fit terdapat beberapa parameter yang dapat dilakukan penyesuaian diantaranya sebagai berikut

No	Parameter
1	Verbose
2	Epoch
3	Steps_per_epoch

Verbose adalah parameter yang digunakan untuk menentukan grafik visual dari proses pelatihan. Epoch merupakan variabel yang mengatur jumlah iterasi pada saat pelatihan model, sedangkan Steps\_per\_epoch adalah variabel yang menyatakan jumlah total langkah sebelum menyatakan satu epoch selesai dan memulai epoch berikutnya.

### 3.9 Pengujian Model



Pada tahap pengujian model ini data uji yang digunakan adalah data yang sudah melalui tahap pre-processing. Dapat dilihat pada gambar diatas, tahap pertama yang dilakukan adalah dengan mengambil atau memuat data uji, lalu memuat model CNN yang telah dilatih dan dilakukan proses pengujian. Setelah melalui proses pengujian, nilai prediksi akan dibandingkan dengan nilai asli berdasarkan pada data

uji, kemudian dapat ditentukan jumlah prediksi yang benar dan setelahnya dapat dilakukan proses evaluasi model dengan menggunakan bantuan confusion matrix.

### 3.10 Evaluasi Model

Setelah melakukan pelatihan terhadap model maka akan didapatkan bobot dari setiap skenario, tahapan selanjutnya adalah melakukan evaluasi model pada setiap skenario. Evaluasi yang dilakukan menggunakan Confusion Matrix. Berikut merupakan tabel dari Confusion Matrix yang digunakan:

		Nilai Prediksi	
	Label	Negatif (0)	Positif (1)
Nilai Asli	Negatif (0)	True Negatif(TN)	False Positif(FP)
	Positif (1)	False Negatif(FN)	True Positif(TP)

### 3.11 Lingkungan Pengembangan

Pada pengembangan penelitian yang dibuat yaitu klasifikasi gambar paru-paru CT-Scan dengan Canny Edge Detector diperlukan beberapa bantuan

dari perangkat lunak dan perangkat keras sebagai penunjang pengembangan.

Beberapa perangkat keras yang digunakan adalah sebagai berikut:

1. Laptop Asus Vivobook dengan central unit processor (CPU) AMD Ryzen 3.
2. RAM 8 GB
3. Radeon R6

Sedangkan perangkat lunak yang digunakan dalam pengembangan penelitian ini adalah sebagai berikut:

1. Sistem operasi Windows 11 64 bit
2. Bahasa pemrograman python
3. Google Colab
4. Tensorflow