**SEMINAR PROPOSAL**

**DETEKSI PENYAKIT PNEUMONIA MENGGUNAKAN ALGORITMA *FASTER REGION CONVOLUTIONAL NEURAL NETWORK* (R-CNN) BERBASIS CITRA DIGITAL RONTGEN DADA**

**Diajukan Untuk Memenuhi Salah Satu Syarat Memperoleh Gelar**

**Sarjana Teknik**

****

**HADIJAH NISA IFAYATIN**

**E1E120008**

**JURUSAN TEKNIK INFORMATIKA**

**FAKULTAS TEKNIK**

**UNIVERSITAS HALU OLEO**

**KENDARI**

**2024**

# **LEMBAR PENGESAHAN**

**Seminar Proposal**

**DETEKSI PENYAKIT PNEUMONIA MENGGUNAKAN ALGORITMA *FASTER REGION CONVOLUTIONAL NEURAL NETWORK* (R-CNN) BERBASIS CITRA DIGITAL RONTGEN DADA**

Adalah benar dibuat oleh saya sendiri dan belum pernah dibuat dan diserahkan sebelumnya baik sebagian ataupun seluruhnya, baik oleh saya ataupun orang lain, baik di Universitas Halu Oleo ataupun institusi pendidikan lainnya.

Kendari, Desember 2023

**Hadijah Nisa Ifayatin**

**NIM.E1E120008**

|  |  |
| --- | --- |
| **Pembimbing I**  **Dr. Ir. Muh. Ihsan Sarita, M.Kom.**  **NIP.19650209 198902 1 000** | **Pembimbing II**  **Rizal Adi Saputra, ST., M.Kom**  **NIP.19910406 201903 1 021** |

**Mengetahui,**

**Ketua Jurusan Teknik Informatika**

**Fakultas Teknik Universitas Halu Oleo**

**Isnawaty, S.Si., MT.**

**NIP.19761117 200812 2 001**

# **DAFTAR ISI**

[LEMBAR PENGESAHAN i](#_Toc156118735)

[DAFTAR ISI ii](#_Toc156118736)

[INTISARI v](#_Toc156118737)

[*ABSTRACT* vi](#_Toc156118738)

[DAFTAR GAMBAR vii](#_Toc156118739)

[DAFTAR TABEL viii](#_Toc156118740)

[BAB I PENDAHULUAN 1](#_Toc156118741)

[1.1 Latar Belakang 1](#_Toc156118742)

[1.2 Rumusan Masalah 3](#_Toc156118743)

[1.3 Batasan Masalah 3](#_Toc156118744)

[1.4 Tujuan Penelitian 4](#_Toc156118745)

[1.5 Manfaat Penelitian 4](#_Toc156118746)

[1.6 Sistematika Penulisan 4](#_Toc156118747)

[1.7 Tinjauan Pustaka 5](#_Toc156118748)

[BAB II LANDASAN TEORI 7](#_Toc156118749)

[2.1 Deteksi Objek 7](#_Toc156118750)

[2.2 Pneumonia 7](#_Toc156118751)

[2.3 *Deep Learning* 9](#_Toc156118752)

[2.4 *Convolutional Neural Network* (CNN) 10](#_Toc156118753)

[2.5 *Region-based Convolutional Neural Network* (R-CNN) 11](#_Toc156118754)

[2.6 *Faster Region Convolutional Neural Network* (R-CNN) 11](#_Toc156118755)

[2.6.1 *Convolutional Backbone (Backbone ConvNet)* 13](#_Toc156118756)

[2.6.2 *Region Proposal Network* (RPN) 17](#_Toc156118757)

[2.6.3 *Feature Extraction* 19](#_Toc156118758)

[2.6.4 *Rol (Region of Interest) Pooling* 19](#_Toc156118759)

[2.6.5 Klasifikasi dan *Bounding Box Regression* 20](#_Toc156118760)

[2.7 Citra Digital 20](#_Toc156118761)

[2.7.1 Citra RGB 21](#_Toc156118762)

[2.7.2 Citra *Grayscale* 22](#_Toc156118763)

[2.8 Format Citra Digital 23](#_Toc156118764)

[2.8.1 *Digital Imaging and Communications in Medicine* 23](#_Toc156118765)

[2.8.2 *Joint Photographic Experts Group* 24](#_Toc156118766)

[2.8.3 *Portable Network Graphics* 24](#_Toc156118767)

[2.9 Pengolahan Citra Digital 25](#_Toc156118768)

[2.9.1 *Resizing* 25](#_Toc156118769)

[2.9.2 Normalisasi Citra 26](#_Toc156118770)

[2.9.3 Konversi Citra 26](#_Toc156118771)

[2.10 *Image Classification* 27](#_Toc156118772)

[2.11 Citra Rontgen Dada 27](#_Toc156118773)

[2.12 Python 28](#_Toc156118774)

[2.13 TensorFlow 29](#_Toc156118775)

[2.14 Google Collaboratory 29](#_Toc156118776)

[2.15 *Rational Unified Process* (RUP) 30](#_Toc156118777)

[2.16 *Confusion Matrix* 31](#_Toc156118778)

[2.17 *Flowchart* 33](#_Toc156118779)

[2.18 *Graphical User Interface* (GUI) 34](#_Toc156118780)

[2.19 Pengujian *Black Box* 34](#_Toc156118781)

[BAB III METODOLOGI PENELITIAN 35](#_Toc156118782)

[3.1 Metode Pengumpulan Data 35](#_Toc156118783)

[3.2 Metode Pengembangan Sistem 37](#_Toc156118784)

[3.3 Waktu dan Tempat Penelitian 38](#_Toc156118785)

[3.3.1 Waktu Penelitian 38](#_Toc156118786)

[3.3.2 Tempat Penelitian 38](#_Toc156118787)

[3.4 Analisis Kebutuhan Sistem 39](#_Toc156118788)

[3.4.1 Analisis Kebutuhan Fungsional 39](#_Toc156118789)

[3.4.2 Analisis Kebutuhan Non Fungsional 40](#_Toc156118790)

[3.5 Perancangan Sistem 41](#_Toc156118791)

[3.5.1 Perancangan Alur Sistem Secara Umum 41](#_Toc156118792)

[3.5.2 Perancangan Alur Metode *Faster* R-CNN 43](#_Toc156118793)

[3.5.3 Perancangan Antarmuka 43](#_Toc156118794)

[3.6 Skema Evaluasi Kinerja Sistem 44](#_Toc156118795)

[3.6.1 Pengujian Metode *Faster Region Convolutional Neural Network* 44](#_Toc156118796)

[3.6.2 Pengujian Fungsionalitas Perangkat Lunak 47](#_Toc156118797)

[DAFTAR PUSTAKA 48](#_Toc156118798)

**INTISARI**

Hadijah Nisa Ifayatin, E1E120008

**DETEKSI PENYAKIT PNEUMONIA MENGGUNAKAN ALGORITMA *FASTER REGION CONVOLUTIONAL NEURAL NETWORK* (R-CNN) BERBASIS CITRA DIGITAL RONTGEN DADA**

Proposal, Fakultas Teknik, 2023

Kata Kunci: Citra Rontgen Dada, *Deep Learning,* Deteksi Pneumonia, *Faster* R-CNN, *VGG-16.*

Pneumonia menjadi perhatian utama dalam layanan kesehatan yang masuk pada 10 penyakit terbanyak di fasilitas pelayanan kesehatan. Pneumonia, infeksi pada kantung udara di paru-paru merupakan penyakit serius yang dapat menyebabkan kematian. Data statistik dari Badan Pusat Statistik Provinsi Sulawesi Tenggara menunjukkan adanya peningkatan kasus pneumonia, terutama di rumah sakit. Kasus pneumonia pada balita di Provinsi Sulawesi Tenggara pada tahun 2022 mencapai 13.214, namun hanya sekitar 11,14% yang teridentifikasi dan ditangani.

Penelitian ini bertujuan untuk mengimplementasikan metode *Faster Convolutional Neural Network* (*Faster* RCNN) dengan menggunakan model VGG16. Metode ini digunakan untuk membagi citra rontgen dada menjadi wilayah-wilayah tertentu, yang kemudian diekstraksi guna mendeteksi pneumonia melalui analisis gambar dari sekumpulan data. Algoritma *Faster*-RCNN dipilih karena keunggulannya dalam bidang *computer vision* dan tingkat akurasi yang tinggi. Sistem ini diharapkan dapat mendeteksi penyakit pneumonia dan normal melalui menggunakan *Faster* RCNN.

Pada tahap pengujian, sistem ini akan dievaluasi dengan menggunakan *confusion matrix* sebagai metode evaluasi utama. Pengujian ini akan mencakup nilai rata-rata *precision, specificity, sensitivity*, dan *accuracy*  untuk setiap fitur utama dari sistem yang diusulkan. Dengan demikian, penelitian ini bertujuan untuk mengimplementasikan model deteksi pneumonia. Evaluasi kinerja sistem juga menggunakan pengujian *black box*. Hasil pengujian menunjukkan bahwa sistem berhasil mengenali citra rontgen dada yang terkena pneumonia atau mormal dan hasil akhir diperoleh berdasarkan kemiripan antara data uji dan data latih yang telah disiapkan untuk mengklasifikasikan output antara pneumonia dan normal.

***ABSTRACT***

*Hadijah Nisa Ifayatin, E1E120008*

***PNEUMONIA DETECTION USING FASTER REGION CONVOLUTIONAL NEURAL NETWORK (R-CNN) ALGORITHM BASED ON DIGITAL CHEST X-RAY IMAGE***

*Proposal, Fakulty of Engineering, 2023*

*Key Word: Chest X-ray Image, Deep Learning, Pneumonia Detection, Faster R-CNN, VGG-16.*

*Pneumonia is a major concern in healthcare, ranking among the top 10 most common diseases in healthcare facilities. Pneumonia, an infection of the air sacs in the lungs, is a serious disease that can cause death. Statistics from the Central Bureau of Statistics of Southeast Sulawesi Province show an increase in pneumonia cases, especially in hospitals. The number of pneumonia cases among under-fives in Southeast Sulawesi Province in 2022 reached 13,214, but only about 11.14% were identified and treated.*

*This study aims to implement the Faster Convolutional Neural Network (Faster RCNN) method using the VGG16 model. This method is used to divide chest X-ray images into specific regions, which are then extracted to detect pneumonia through image analysis of a set of data. The Faster-RCNN algorithm was chosen due to its advantages in the field of computer vision and its high accuracy rate. Hopefully this system detects pneumonia and normal diseases through using Faster RCNN.*

*In the testing phase, this system will be evaluated using confusion matrix as the main evaluation method. This test will include the average value of precision, specificity, sensitivity, and accuracy for each main feature of the proposed system. Thus, this study aims to implement a pneumonia detection model. The evaluation of the system performance also uses black box testing. The test results show that the system successfully recognises chest X-ray images affected by pneumonia or normal and the final result is obtained based on the similarity between the test data and the training data that has been prepared to classify the output between pneumonia and normal.*

# **DAFTAR GAMBAR**

[Gambar 2.1 Citra Rontgen Pneumonia 8](#_Toc156118858)

[Gambar 2.2 Arsitektur *Deep Learning* 10](#_Toc156118859)

[Gambar 2.3 Arsitektur CNN Secara Umum 10](#_Toc156118860)

[Gambar 2.4 Arsitektur R-CNN 11](#_Toc156118861)

[Gambar 2.5 Arsitektur *Faster* RCNN 12](#_Toc156118862)

[Gambar 2.6 Arsitektur Model VGG-16 13](#_Toc156118863)

[Gambar 2.7 Arsitektur RPN 17](#_Toc156118864)

[Gambar 2.8 Area (a) Tumpang Tindih dan (b) Gabungan 18](#_Toc156118865)

[Gambar 2.9 Citra Digital 21](#_Toc156118866)

[Gambar 2.10 Representasi Citra RGB 22](#_Toc156118867)

[Gambar 2.11 Citra *Grayscale* 22](#_Toc156118868)

[Gambar 2.12 Visualisasi Aras Keabuan 23](#_Toc156118869)

[Gambar 2.13 Perbedaan Citra Asli dan Citra *Resize* 26](#_Toc156118870)

[Gambar 2.14 Citra RGB dan Citra *Grayscale* 26](#_Toc156118871)

[Gambar 2.15 Klasifikasi Citra Rontgen Dada Pneumonia & Normal 27](#_Toc156118872)

[Gambar 2.16 Contoh Citra Rontgen Dada 27](#_Toc156118873)

[Gambar 2. 17 Alur Kerja *Rational Unified Process* 30](#_Toc156118874)

[Gambar 3.1 Dataset Citra Rontgen Dada Normal 36](#_Toc156118844)

[Gambar 3.2 Dataset Citra Rontgen Dada Pneumonia 36](#_Toc156118845)

[Gambar 3.3 Citra Rontgen Normal RSUD Konawe Selatan 36](#_Toc156118846)

[Gambar 3.4 Citra Pneumonia RSUD Konawe Selatan 36](#_Toc156118847)

[Gambar 3.5 Citra Normal RSUD Kota Kendari 36](#_Toc156118848)

[Gambar 3.6 Citra Pneumonia RSUD Kota Kendari 36](#_Toc156118849)

[Gambar 3.7 *Flowchart* Sistem Secara Umum 41](#_Toc156118850)

[Gambar 3.8 Arsitektur Sistem *Faster* RCNN 43](#_Toc156118851)

[Gambar 3.9 Perancangan *Interface* 44](#_Toc156118852)

[Gambar 3.10 Contoh Grafik *Confusion Matrix* 45](#_Toc156118853)

**DAFTAR TABEL**

[Tabel 2.1 *Confussion Matrix* 32](#_Toc156119007)

[Tabel 3.1 Waktu Pelaksanaan Penelitian 38](#_Toc156118998)

[Tabel 3.2 Spesifikasi Perangkat Keras 40](#_Toc156118999)

[Tabel 3.3 Spesifikasi Perangkat Lunak 41](#_Toc156119000)

[Tabel 3.4 Contoh Tabel *Confusion Matrix* 44](#_Toc156119001)

[Tabel 3.5 Hasil Skenario Pengujian Metode *Faster* RCNN 46](#_Toc156119002)

[Tabel 3.6 Pengujian *Black Box* 47](#_Toc156119003)

**BAB I   
PENDAHULUAN**

1. **Latar Belakang**

Berdasarkan statistik dari World Health Organization (WHO) tahun 2019, pneumonia menjadi penyebab 14% dari total kematian pada anak-anak di bawah usia 5 tahun, dengan jumlah kematian mencapai 740.180 jiwa. Menurut data Riset Kesehatan Dasar Indonesia tahun 2018, angka penderita pneumonia cenderung meningkat seiring dengan pertambahan usia. Pada kelompok usia 55-64 tahun, tingkat kejadian mencapai 2,5%, sementara pada kelompok usia 65-74 tahun mencapai 3,0%, dan pada kelompok usia 75 tahun keatas, tingkat kejadian mencapai 2,9% (Hatim, 2022).

Pembangunan sektor kesehatan sebagaimana terintegrasikan dalam tujuan *Sustainable Development Goals* (SDGs) menjamin kehidupan yang sehat dan upaya meningkatkan kesejahteraan seluruh penduduk pada semua usia. ISPA (Infeksi Saluran Pernafasan Akut) merupakan penyakit yang disebabkan oleh lebih dari 300 jenis mikroorganisme yang menyerang organ pernafasan dari hidung sampai alveoli dan organ adneksa. Penyakit ISPA masuk kedalam 10 penyakit terbanyak di fasilitas pelayanan kesehatan, mulai ISPA ringan seperti rhinitis hingga yang menyebabkan kematian yaitu pneumonia (Direktorat Pencegahan dan Pengendalian Penyakit Menular Kementrian Kesehatan, 2022). Pneumonia adalah infeksi yang mengobarkan kantung udara disalah satu atau kedua paru-paru. Kantung udara dapat berisi cairan atau nanah (bahan bernanah), menyebabkan batuk berdahak atau nanah, demam, menggigil, dan kesulitan bernapas (Hastuti, 2023).

[Menurut Badan Pusat Statistik Provinsi Sulawesi Tenggara, pada tahun 2020, terdapat 3.051 kasus penderita pneumonia di Sulawesi Tenggara, yang merupakan sekitar 11,61% dari perkiraan penderita](https://sultra.bps.go.id/statictable/2021/04/27/3075/jumlah-kasus-penyakit-menurut-kabupaten-kota-dan-jenis-penyakit-di-sulawesi-tenggara-2020.html) (Jumakil *et al*., 2019). Pneumonia termasuk dalam daftar 10 penyakit dengan jumlah kasus terbesar dan bisa menyebabkan kematian. Penyakit ini bersifat kritis karena berisiko memicu peradangan pada kantong-kantong udara (*alveoli*) pada salah satu atau kedua paru-paru.  Golongan sebab sakit (rawat inap) RSUD Kota Kendari Tahun 2022 yang berada pada urutan pertama adalah pneumonia sebanyak 595 pasien (Fitrianingsi *et al*., 2022).

Estimasi kasus pneumonia pada balita di Provinsi Sulawesi Tenggara pada tahun 2022 mencapai 13.214, namun hingga saat ini, hanya 1.476 kasus balita yang teridentifikasi dan ditangani, atau sekitar 11,14% dari perkiraan jumlah penderita (Tiku *et al*., 2023). Pada tahun 2021, pneumonia dan diare masih menjadi penyebab kematian terbanyak di Indonesia pada masa *post neonatal*, yaitu sebesar 14,4% kematian karena pneumonia (Sibuea *et al*., 2022). Proses diagnosis kelainan paru-paru saat ini masih sangat mengandalkan pengamatan visual dokter dalam menafsirkan hasil citra rontgen dada, sehingga penilaian cenderung bersifat subyektif tergantung pada interpretasi masing-masing dokter.

Penelitian yang dinyatakan dalam judul Klasifikasi Pneumonia pada Citra *X-Rays* Paru-paru Menggunakan GLCM dan LVQ bertujuan untuk mengklasifikasikan hasil sinar-x ke dalam sebuah program guna menentukan keberadaan pneumonia. Metode yang digunakan melibatkan ekstraksi fitur menggunakan *Gray Level Co-Occurrence Matrix* (GLCM) dengan arah 0°, 45°, 90°, 135°, dan klasifikasi menggunakan *Learning Vector Quantization* (LVQ). Akurasi terbaik pada data train mencapai 89,714%, sementara akurasi terbaik pada data test adalah 74,000%. Hasil ini diperoleh melalui pengujian dengan *learning rate* = 0,4 (Lin *et al*., 2021).

Ekstraksi fitur Sobel pada CNN untuk meningkatkan akurasi pada mesin pembelajaran. Hasil penelitian mencapai *precision* sebesar 91%, *recall* sebesar 92.8%, dan *accuracy*  sebesar 91.54% dengan nilai *epoch* 50, *learning rate* 0.0001, dan *batch size* 20. Penelitian tersebut terdapat pada penelitian berjudul Identifikasi Pneumonia Pada Citra *X-Ray* Paru-paru Menggunakan Metode *Convolutional Neural Network* (CNN) Berdasarkan Ekstraksi Fitur Sobel (Yopento *et al*., 2022).

Menurut penelitian dari jurnal yang berjudul *Pneumonia Detection Using an Improved Algorithm Based on Faster R-CNN* mengusulkan metode *DeepConv-DilatedNet* untuk mengidentifikasi pneumonia dalam gambar sinar-x. Menggunakan detektor *Faster* R-CNN dan *Feature Pyramid Network*, metode ini melibatkan jaringan saraf residual dengan *dilated bottleneck*. Algoritma ini mencapai *Mean Average Precision* (mAP) 39,23% pada dataset RSNA dan 38,02% pada *ChestX-ray14*, melampaui algoritma lain dan memperoleh akurasi sebesar 81%. Dengan arsitektur konvolusi penuh, *Soft-NMS*, dan *K-Means++*, algoritma ini dapat memberikan informasi lokasi lesi pneumonia kepada dokter (Yao *et al*., 2021).

Implementasi Metode *Faster Region Convolutional Neural Network* (*Faster* R-CNN) Untuk Pengenalan Jenis Burung *Lovebird* merupakan penelitian untuk mengenali jenis *Lovebird* dan menemukan motif atau warna asli berdasarkan tingkat akurasi deteksi objek. Penelitian ini mencapai akurasi 96% dan *loss* 0,2% pada data set 808 gambar *lovebird* dengan 8 kelas yang berbeda. Kesimpulannya, *Faster* R-CNN dapat diterapkan dengan sukses pada pengenalan jenis burung *lovebird* (Charli *et al*., 2020).

Berdasarkan latar belakang dan penelitian terdahulu yang telah dipaparkan perlunya penelitian yang lebih beragam pada implementasi metode *Faster* R-CNN maka penelitian berjudul **Deteksi Penyakit Pneumonia Menggunakan Algoritma *Faster Region Convolutional Neural Network* (R-CNN) Berbasis Citra Digital Rontgen Dada** diharapkan mampu diimplementasikan dan memiliki akurasi yang lebih baik.

1. **Rumusan Masalah**

Berdasarkan latar belakang masalah yang telah diuraikan diatas, maka penulis merumuskan masalah penelitian yaitu bagaimana menerapkan algoritma *Faster* *Region Convolutional Neural Network* (R-CNN)untuk deteksi penyakit pneumonia berbasis citra digital rontgen dada?

1. **Batasan Masalah**

Berdasarkan rumusan masalah yang ada, adapun batasan masalah dalam penulisan tugas akhir ini adalah:

1. Penelitian ini fokus pada pendeteksi penyakit pneumonia menggunakan citra rontgen dada.
2. Penelitian ini berfokus pada citra rontgen dada normal dan pneumonia dengan format citra .dcm, .jpeg, dan .png.
3. Interpretasi hasil deteksi penyakit pneumonia ini disajikan melalui visualisasi pada dashboard berbasis GUI.
4. **Tujuan Penelitian**

Adapun tujuan dari penelitian ini adalah mendeteksi citra rontgen dada untuk diklasifikasikan sebagai pneumonia dan normal menggunakan algoritma *Faster* *Region Convolutional Neural Network* (R-CNN).

1. **Manfaat Penelitian**

Manfaat dari penelitian ini adalah dapat mengimplementasikan algoritma *Faster* *Region Convolutional Neural Network* (R-CNN)untuk deteksi penyakit pneumonia berbasis citra digital rontgen dada.

1. **Sistematika Penulisan**

Sistematika penulisan yang dilakukan dalam penyusunan penelitian ini adalah:

**BAB I PENDAHULUAN**

Bab ini berisi tentang deskripsi umum isi laporan yang meliputi latar belakang, rumusan masalah, batasan masalah, tujuan penelitian, manfaat penelitian, sistematika penulisan dan tinjauan pustaka.

**BAB II LANDASAN TEORI**

Bab ini berisi teori-teori penunjang yang berhubungan dengan penelitian ini berupa teori yang bersumber dari media cetak maupun media elektronik berupa pengertian deteksi objek, pneumonia, *deep learning*, metode *Faster Region Convolutional Neural Network* (RCNN), citra digital, pengolahan citra digital, *image segmentation, image classification*, citra rontgen dada, Python, TensorFlow, dan GUI.

**BAB III METODOLOGI PENELITIAN**

Bab ini berisi tahapan tentang proses dan prosedur pengumpulan data, prosedur analisis metode, prosedur pengembangan sistem serta waktu dan tempat penelitian.

1. **Tinjauan Pustaka**

Berikut rangkuman penelitian terdahulu yang bertemakan pengolahan citra thorax dan juga penelitian yang menggunakan metode *Faster Region Convolutional Neural Network* yang menjadi rujukan dalam penyusunan penelitian ini:

Jurnal dari penelitian *Pneumonia and COVID-19 Classication in Chest Xrays Using Faster Region-Based Convolutional Neural Networks (Faster R-CNN)* mengklasifikasikan pneumonia, COVID-19 dan normal menggunakan metode *Faster Region-Based Convolutional Neural Networks* (CNN) yang menggunakan kombinasi 250 *epoch* dan panjang *epoch* 250 membuat model mencapai kinerjanya yang terbaik. Akurasi maksimum yang dicapai adalah 78% (Farhat *et al*., 2022).

Akurasi rata-rata sebesar 97,36%, dan mendapatkan nilai rata-rata sensitivitas, spesifisitas, dan skor F1 masing-masing sebesar 97,65%, 95,48%, dan 98,46%. Ini juga memberikan nilai presisi lebih dari 99,00% pada penelitian dengan judul *COVID Faster R-CNN: A Novel Framework to Diagnose Novel Coronavirus Disease (COVID-19) in X-Ray Images* (Hassan Shibly *et al*., 2020).

Meurut penelitian tentang kanker paru-paru dengan judul Sistem Deteksi Keganasan Kanker Paru-Paru pada CT *Scan* dengan Menggunakan Metode *Mask Region-based Convolutional Neural Network* (*Mask* R-CNN) untuk mendeteksi dan mengklasifikasikan tipe nodul paru-paru menjadi *benign* ukuran besar, *benign* ukuran kecil, dan *malignant*. Nodul *benign* mempunyai rata-rata performa yang tinggi dengan rerata *precision* 0.852, *sensitivity* 0.958, *specificity* 0.82, dan *dice similarity* 0.894 (Hermawati, 2021).

Penelitian lainnya yang dilakukan oleh Septian dan Septanto (2022), yakni Pengembangan Model Pendeteksian Gambar Alat Musik dengan Metode *Faster* R-CNN dengan *Library Keras* mengeksplorasi pengembangan aplikasi deteksi jenis alat musik menggunakan metode keras *Faster* R-CNN dan *ResNet50*. Metode *Faster* R-CNN digunakan untuk memperoleh *output* dengan lebih cepat daripada RCNN dengan hasil akurasi 99% (Septian *et al*., 2022).

Pada penelitian dalam pemanfaatan kecerdasan buatan dalam pengolahan data pencitraan sinar-x paru-paru anak pneumonia dan non-pneumonia dengan metode segmentasi dan deteksi tepi menghasilkan uji coba. Dengan data dari Rumah Sakit Sardjito, akurasi mencapai 73,3%, sensitivitas 62,5%, dan spesifisitas 77,2%. Uji coba dengan data Kaggle juga menunjukkan akurasi sebesar 83,3%, sensitivitas 70%, dan spesifisitas 90% (Prasetyo, 2020).

Penelitian berjudul Implementasi Metode *Region Convolutional Neural Network* Dalam Mendiagnosa Anomali Pneumonia Pada Foto Thorax fokus pada prediksi objek pneumonia pada citra thorax di Indonesia menggunakan metode *Region Convolutional Neural Network* (R-CNN). Pengujian dilakukan pada 170 data masukan dengan 160 data latih dan 10 data uji, dengan hasil mAP sebesar 42.84% (Fauzy, 2019).

Judul penelitian *DeePNeu: Robust Detection of Pneumonia Symptoms using Faster R-CNN* menggunakan algoritma *Faster* R-CNN untuk mendeteksi gejala pneumonia dengan efektif. Hasil eksperimen menunjukkan peningkatan akurasi model. Model ini dapat mendeteksi citra sinar-x pneumonia sebanyak 4.032 yang terdiri dari data latih 3.226 dan data uji sebanyak 806 data dengan akurat menggunakan *bounding box*. Proses pelatihan menghasilkan *loss score* = 0.1988, *area box loss* = 0.1230, *objektivitas loss* = 0.0030, dan RPN *box loss* = 0.0039 (Ordiyasa *et al*., 2022).

Berdasarkan dari tinjauan pustaka yang telah dikaji bahwa *Faster* RCNN memiliki akurasi cukup tinggi dan waktu eksekusi yang lebih cepat dalam deteksi objek serta memiliki nilai *loss* yang rendah, maka diusulkan **Deteksi Penyakit Pneumonia Menggunakan Algoritma *Faster Region Convolutional Neural Network* (R-CNN) Berbasis Citra Digital Rontgen Dada** dengan harapan pada penelitian ini mencoba untuk mengimplementasikan algoritma *Faster* RCNN untuk deteksi pneumonia menggunakan citra rontgen dan memperbaiki akurasi pada penelitian sebelumnya.

**BAB II  
LANDASAN TEORI**

1. **Deteksi Objek**

Deteksi objek adalah masalah yang lebih kompleks dari kedua pelokalan posisi suatu objek dalam suatu gambar serta menunjukkan jenis objek, karena itu diperlukan teknik yang lebih kompleks. Deteksi objek atau lebih dikenal sebagai pengenalan objek dalam bahasa Indonesia, adalah sistem yang beroperasi untuk mengidentifikasi suatu objek dalam gambar atau video. Secara teknis *object detection* merupakan suatu algoritma dalam *computer vision* yang menghasilkan kotak pembatas untuk kategori objek tertentu dan menetapkan skor untuk klasifikasi. Deteksi objek dibagi ke dalam dua bidang yaitu *soft detection*, yang hanya mendeteksi keberadaan objek, serta *hard detection* yang mendeteksi keberadaan dan lokasi dari suatu objek (Imantiyar, 2021).

*Object detection* atau deteksi objek merupakan suatu metode yang dikembangkan untuk membuat sistem bisa mengenali suatu objek yang ada di dalam sebuah foto atau video. Secara umum, deteksi objek diciptakan untuk dapat mengidentifikasi objek seperti manusia, tumbuhan, kendaraan, atau objek lainnya yang sesuai dengan keinginan pengembang. Seiring dengan kemajuan teknologi, deteksi objek memiliki banyak penerapan di bidang komputer visi, termasuk dalam peningkatan automasi. Deteksi objek atau *object detection* bagian dari *computer vision*. *Object detection* merujuk pada kemampuan komputer untuk mengidentifikasi sejumlah objek dalam suatu gambar. Hal ini dapat dicapai dengan mengekstrak fitur gambar seperti garis, sudut, kontur, dan warna (Rahmad *et al*., 2019).

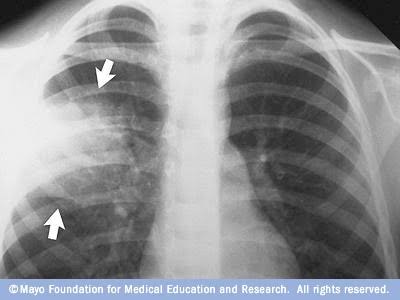
1. **Pneumonia**

Pneumonia merupakan penyakit dari paru-paru dan sistem pernapasan di mana alveoli mikroskopik udara mengisi kantong dari paru yang bertanggung jawab untuk menyerap oksigen dari atmosfer menjadi radang dan dengan penimbunan cairan. Pneumonia disebabkan oleh berbagai macam sebab, meliputi infeksi karena bakteri, virus, jamur atau parasit. Pneumonia juga dapat terjadi karena bahan kimia atau kerusakan fisik dari paru-paru atau secara tak langsung dari penyakit lain seperti kanker paru atau penggunaan alkohol (Honainah, 2022).

Gejala khas yang berhubungan dengan pneumonia meliputi batuk, nyeri dada, demam, dan sesak nafas. Alat diagnosa meliputi sinar-x dan pemeriksaan sputum. Pengobatan pneumonia bergantung pada penyebabnya, jika disebabkan oleh bakteri maka diobati dengan antibiotika. Pneumonia merupakan penyakit yang umumnya terjadi pada semua kelompok umur dan menunjukkan penyebab kematian pada orang tua dan orang dengan penyakit kronik. Prognosis untuk tiap orang berbeda tergantung dari jenis pneumonia, pengobatan yang tepat, ada tidaknya komplikasi dan kesehatan orang tersebut serta tes penting untuk mendeteksi pneumonia pada keadaan yang tidak jelas ialah dengan foto thorax.

Foto thorax dapat menampakkan daerah opak (terlihat putih) yang menggambarkan konsolidasi. Pneumonia tidak selalu dilihat oleh sinar-x karena penyakit pneumonia pada tingkat permulaan atau karena mengenai bagian paru tertentu yang sulit dilihat dengan sinar-x. Foto thorax juga digunakan untuk evaluasi adanya komplikasi dari pneumonia (Basyir, 2021).

Pneumonia pada citra rontgen dada dapat ditandai oleh berbagai ciri, termasuk penyamaran struktur normal paru-paru, infiltrasi yang menunjukkan cairan atau materi infeksi, penebalan garis batas antara paru-paru dan rongga dada, gambaran konsolidasi, penyempitan ruang udara, serta gambaran *fluffy* atau *cotton wool*. Selain itu, pneumonia dapat menyebabkan pembesaran hati atau limpa dan ketidakjelasan pada batas *mediastinum*. Meskipun citra rontgen memberikan gambaran visual, diagnosis pneumonia memerlukan evaluasi klinis menyeluruh, riwayat medis, dan tes tambahan. Adapun citra rontgen penderita pneumonia dapat dilihat pada Gambar 2.1.



Gambar 2.1 Citra Rontgen Pneumonia

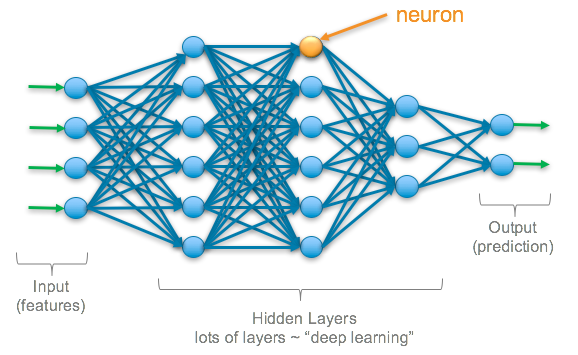
1. ***Deep Learning***

Pembelajaran mendalam atau *deep learning* adalah cabang dari pembelajaran mesin yang memanfaatkan jaringan saraf mendalam dengan banyak lapisan untuk melakukan prediksi, khususnya berhasil dalam bidang visi komputer, pengenalan suara, pemahaman bahasa alami, dan berbagai aplikasi lainnya. *Deep Learning* memungkinkan model komputasi yang terdiri dari beberapa lapisan pengolahan untuk mempelajari representasi data dengan berbagai tingkat abstraksi. *Deep learning* menemukan struktur yang rumit dalam kumpulan data yang besar dengan menggunakan algoritma *backpropagation* untuk menunjukkan bagaimana sebuah mesin harus mengubah parameter internalnya yang digunakan untuk menghitung representasi pada setiap lapisan dari representasi pada lapisan sebelumnya (Marpaung *et al*., 2022).

Kelebihan dari *deep learning* juga mencakup kemampuannya untuk secara otomatis mengekstraksi fitur dari data tanpa perlu adanya fitur-fitur yang telah diidentifikasi sebelumnya. Namun, *deep learning* juga menuntut daya komputasi yang tinggi dan memerlukan set data pelatihan yang luas. Berkat potensinya yang besar dalam menangani masalah-masalah yang kompleks, *deep learning* telah menjadi fokus utama dalam kemajuan teknologi kecerdasan buatan dan telah diimplementasikan dalam beragam aplikasi, mulai dari kendaraan otonom hingga pengenalan suara dan pemahaman bahasa alami (Steno *et al*., 2021).

Dalam konsep *deep learning*, terdapat beberapa lapisan yang membentuk arsitektur jaringan saraf tiruan (*neural network*). Pertama adalah *input layer*, yang berfungsi untuk menerima berbagai jenis input data, seperti gambar, teks, atau data numerik. Jumlah node dalam input layer disesuaikan dengan jumlah fitur pada data yang dimasukkan. Selanjutnya, terdapat *hidden layers* yang bertugas mengekstraksi fitur kompleks dari data input. *Hidden layers* ini terdiri dari node (*neuron*) yang terhubung satu sama lain, dilengkapi dengan bobot (*weight*) dan bias yang dapat disesuaikan selama proses pelatihan. Hasil akhir dari model dihasilkan oleh *output layer*, yang sesuai dengan tugas khusus yang dihadapi, seperti klasifikasi, regresi, atau tugas lainnya. Top of Form

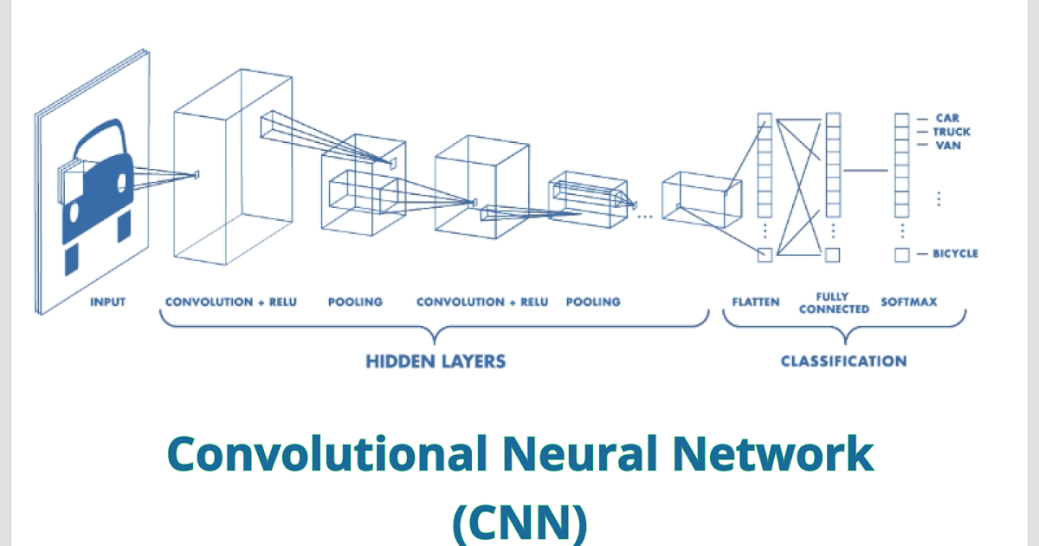
Adapun arsitektur *deep learning* secara umum bisa dilihat pada Gambar 2.2.



Gambar 2.2 Arsitektur Deep Learning

1. ***Convolutional Neural Network* (CNN)**

*Convolutional Neural Networks* (CNN) merupakan varian jaringan saraf tiruan yang umumnya diterapkan dalam pengolahan citra. CNN menjadi alat yang sangat efektif untuk menganalisis serta memahami pola-pola kompleks pada data visual. Fungsi utama CNN adalah untuk mengenali objek-objek dalam suatu citra. Teknik ini mencerminkan upaya manusia dalam menciptakan kemampuan persepsi visual, serupa dengan proses dimana mamalia menghasilkan gambaran dan pemahaman visual. Arsitektur CNN menyerupai jaringan saraf tiruan konvensional, namun dilengkapi dengan neuron-neuron yang memiliki bobot, bias, dan fungsi aktivasi. Metode utama yang digunakan oleh CNN adalah operasi konvolusi, yang menjadi dasar dari algoritmanya (Muharom *et al*, 2019).

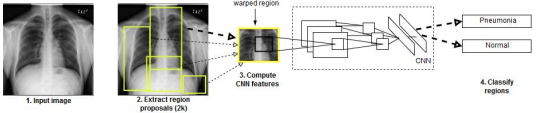


Gambar 2.3 Arsitektur CNN Secara Umum

Pada Gambar 2.3 merupakan gambaran umum jaringan saraf konvolusional (*Convolutional Neural Network* atau CNN) merupakan salah satu jenis dari jaringan saraf tiruan *feed-forward* yang sering digunakan dalam analisis citra. CNN terstruktur dengan satu lapisan masukan (*input layer*), satu lapisan keluaran (*output layer*) yang dapat mengklasifikasikan hasil input, biasanya lapisan ini terdiri dari lapisan *flatten, softmax* dan sejumlah lapisan tersembunyi (*hidden layer*). Lapisan tersembunyi umumnya melibatkan *convolutional layer, pooling layer, ReLU* (*Rectified Linear Unit*) *layer, fully connected layer,* dan *loss layer.*

1. ***Region-based Convolutional Neural Network* (R-CNN)**

*Region-based Convolutional Neural Network* (R-CNN) merupakan metode pengenalan objek yang diperkenalkan oleh Ross Girshick, Jeff Donahue, Trevor Darrell, dan Jitendra Malik pada tahun 2014 dan membagi arsitektur R-CNN dalam 3 tahap, yaitu *extract region proposals* (~2k), *compute* *CNN* *feature*, dan *classify regions* (Megawan *et al*., 2020). Arsitektur tersebut dapat dilihat pada Gambar 2.4.



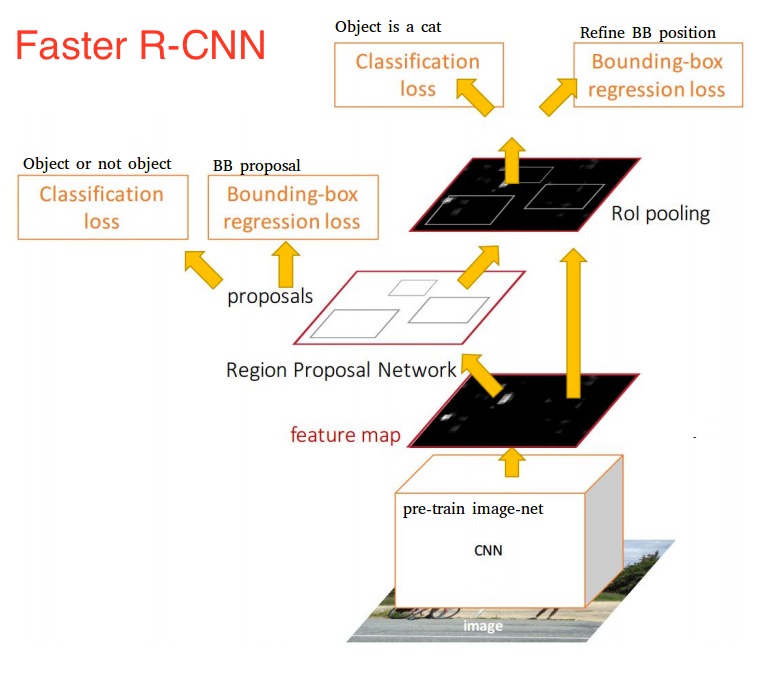
Gambar 2.4 Arsitektur R-CNN

Pada tahap *extract region proposals*, R-CNN melakukan *region search* (pencarian wilayah) menggunakan metode *selective search* hingga menghasilkan 2000 kandidat region proposal dalam bentuk persegi. Pada tahap *compute CNN feature*, *region proposal* tersebut di masukan kedalam *Convolutional Neural Network* (CNN) hingga menjadi fitur. Pada tahap *classify region*, fitur tersebut kemudian menjadi masukan untuk *Support Vector Machine* (SVM) untuk diklasifikasi dan dilakukan perhitungan *bounding-box regression* untuk menghasilkan *bounding box*.

1. ***Faster Region Convolutional Neural Network* (R-CNN)**

*Faster* R-CNN merupakan algoritma deteksi objek berbasis wilayah yang menunjukkan hasil luar biasa pada berbagai deteksi objek. Ada banyak penelitian yang sudah menggunakan *Faster* R-CNN pada bidang deteksi objek, seperti deteksi objek pada gambar dan deteksi mobil pada video. Kinerja *Faster* R-CNN sangat bergantung pada kualitas data *training* yang besar (Megawan *et al*., 2020).

Untuk meningkatkan efisiensi waktu dan menciptakan proposal wilayah yang lebih akurat, digunakan sebuah jaringan yang dikenal dengan nama *Faster* R-CNN. *Faster* R-CNN didesain dengan mengintegrasikan *Faster* R-CNN dengan jaringan saraf baru yang sepenuhnya lapisan konvolusi, yaitu *Region Proposal Network* (RPN). RPN tidak hanya mampu menghasilkan proposal wilayah berkualitas tinggi, melainkan juga mampu secara simultan mengusulkan batasan objek dan mengukur skor objektivitas di setiap posisi. RPN menghasilkan beberapa *bounding box*, setiap *box* memiliki dua skor probabilitas, menunjukkan apakah terdapat objek pada lokasi tersebut atau tidak. *Region* yang dihasilkan tersebut akan menjadi input untuk arsitektur yang serupa dengan *Faster* R-CNN (Hanun *et al*., 2023). Gambar 2.5 adalah arsitektur *Faster* RCNN.

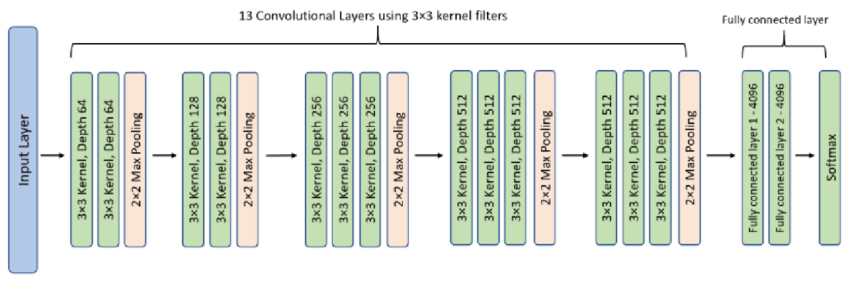


Gambar 2.5 Arsitektur Faster RCNN

Pada Gambar 2.5 merupakan arsitektur *Faster* RCNN secara umum. Citra dataset akan dilatih dalam model CNN menggunakan VGG-16 yang menghasilkan *Fature Map* atau lapisan konvolusi. *Feature map* kemudian digunakan oleh *Region Proposal Network* (RPN). RPN melakukan pemindaian (*sliding window*) pada *feature map* untuk mengidentifikasi potensi lokasi objek dan menghasilkan proposal wilayah (*bounding box*) beserta skor kecocokan (*objectness score*) untuk setiap proposal dan setelah itu menghasilkan klasifikasi. Jika ukuran dalam *bounding box* berbeda maka akan melewati *RoI pooling* untuk mengekstraksi fitur dari area yang berbeda ukuran dan menghasilkan representasi fitur yang tetap berukuran dan mengklasifikasikannya.

1. ***Convolutional Backbone (Backbone ConvNet)***

Arsitektur *Faster* R-CNN menggunakan sebuah *Convolutional Neural Network* (CNN) sebagai bagian dasarnya untuk mengekstrak fitur dari gambar. Pada penelitian ini digunakana VGG-16 sebagai *backbone* untuk ekstraksi fitur.



Gambar 2.6 Arsitektur Model VGG-16

Ilustrasi alur kerja model VGG-16 dapat dilihat pada Gambar 2.6 menjelaskan bahwa VGG-16 memiliki arsitektur yang terdiri dari 16 lapisan, termasuk lapisan konvolusi (*convolutional layers*) dan lapisan *fully connected*. Arsitektur ini terkenal karena kejelasan strukturnya dan kemampuannya untuk mengekstraksi fitur secara hierarki dari citra. Meskipun arsitektur ini tergolong klasik, model-model modern sering kali menggunakan variasi yang lebih ringan dan efisien, terutama untuk tugas-tugas seperti deteksi objek atau segmentasi. Berikut adalah beberapa poin utama tentang cara kerja VGG-16:

1. *Convolutional Layer*

*Convolutional layer* terdiri dari 13 lapisan konvolusi berturut-turut dengan filter 3x3 dan fungsi aktivasi ReLU (*Rectified Linear Unit*). Setiap lapisan konvolusi diikuti oleh lapisan *pooling* maksimum (*max pooling*) dengan filter 2x2 dan *stride* 2. Rumus umum konvolusi untuk menghitung ukuran citra *output dapat dilihat pada* rumus 2.1.

*Keterangan:*

(2.1)

*W*in​: Lebar (*width*) citra *input* *W*out​: Lebar (*width*) citra *output*

*H*in​: Tinggi (*height*) citra *input* *P*: Padding

*H*out​: Tinggi (*height*) citra *output* *S*: Langkah (*stride*)

*K*: Ukuran kernel

Sebagai contoh, dimasukkan citra RGB dengan lebar *W*in​=32, tinggi *H*in​=32, *K*=2, padding *P*=0, dan *stride* *S*=2. Dapat dihitung nilai keluaran *W*out​ dan *H*out​.

Keluaran *W*out​ dan *H*out​ adalah 16 x 16.

1. *Max Pooling*

*Max pooling* membagi *output* dari *convolution layer* menjadi beberapa *grid* kecil lalu mengambil nilai maksimal dari setiap *grid* untuk menyusun matriks citra yang telah direduksi. Rumus umum untuk menghitung *max pooling* dapat dilihat pada rumus 2.2. *D*out​ ditentukan oleh parameter *F* (jumlah filter). Dengan demikian jumlah *channel* pada *output max pooling* sama dengan jumlah filter yang digunakan pada operasi *max pooling*.

(2.2)

*D*out​ = *F*

*Keterangan:*

*F*: Jumlah filter

*D*out​: Jumlah *channel* (*depth*) citra *output*

Sebagai contoh, dimasukkan citra RGB dengan lebar *W*in​=32, tinggi *H*in​=32, lalu di filter menggunakan filter *grayscale* *K*=2, padding *P*=0, nilai *F*=1, dan *stride* *S*=2. Dapat dihitung *W*out​ dan *H*out​.

*D*out​ = *1*

Hasil yang diperoleh setelah dilakukan perhitungan pada rumus 2.2 dapat dituliskan dengan 16 x 16 x 1.

1. *Fully Connected Layer*

*Fully connected layer* dalam jaringan saraf adalah lapisan di mana setiap neuron atau node pada lapisan tersebut terhubung ke setiap neuron pada lapisan sebelumnya dan setelahnya. Dalam lapisan ini, setiap *input* memiliki bobot (*weight*) tersendiri yang dihubungkan dengan setiap neuron pada lapisan tersebut. Fungsi utama dari *fully connected layer* adalah untuk menyatukan informasi dari seluruh area spasial dari *input* dan membuat keputusan akhir berdasarkan informasi tersebut. Terdapat tiga lapisan *fully connected* dengan fungsi *aktivasi ReLU*. Lapisan *fully connected* terakhir memiliki 1.000 node yang sesuai dengan jumlah kelas dalam dataset *ImageNet*. Pada beberapa lapisan *fully connected*, VGG-16 menggunakan teknik *dropout* untuk mencegah *overfitting*. *Dropout* mengacaukan sebagian node secara acak selama pelatihan untuk meningkatkan generalisasi model.

1. *Softmax Layer*

*Softmax layer* merupakan elemen akhir yang mengubah *output* numerik dari lapisan sebelumnya menjadi distribusi probabilitas kelas. Proses ini dimulai dengan mengaplikasikan fungsi eksponensial pada skor untuk membentuk vektor eksponensial. Proses selanjutnya melibatkan penjumlahan semua nilai dalam vektor eksponensial, yang memberikan nilai . Dengan rumus ini, nilai probabilitas untuk masing-masing kelas dihitung, terwakili dalam vektor . Probabilitas ini mencerminkan sejauh mana *input* model cocok dengan masing-masing kelas. Dengan demikian, kelas dengan probabilitas tertinggi dianggap sebagai prediksi akhir dari model. Rumus *softmax* yang diterapkan untuk menghitung probabilitas kelas *classi*​ dapat dilihat pada rumus 2.3.

Keterangan:

(2.3)

​: Probabilitas bahwa *input* termasuk dalam kelas ke-*i*.

​ : Nilai eksponensial dari skor atau nilai *input* untuk kelas ke-*i*.

: Jumlah dari semua nilai eksponensial dari skor untuk seluruh kelas (*n* adalah jumlah kelas).

: Skor atau nilai *input* untuk kelas ke-*j*.

: Skor atau nilai *input* untuk kelas ke-*i*.

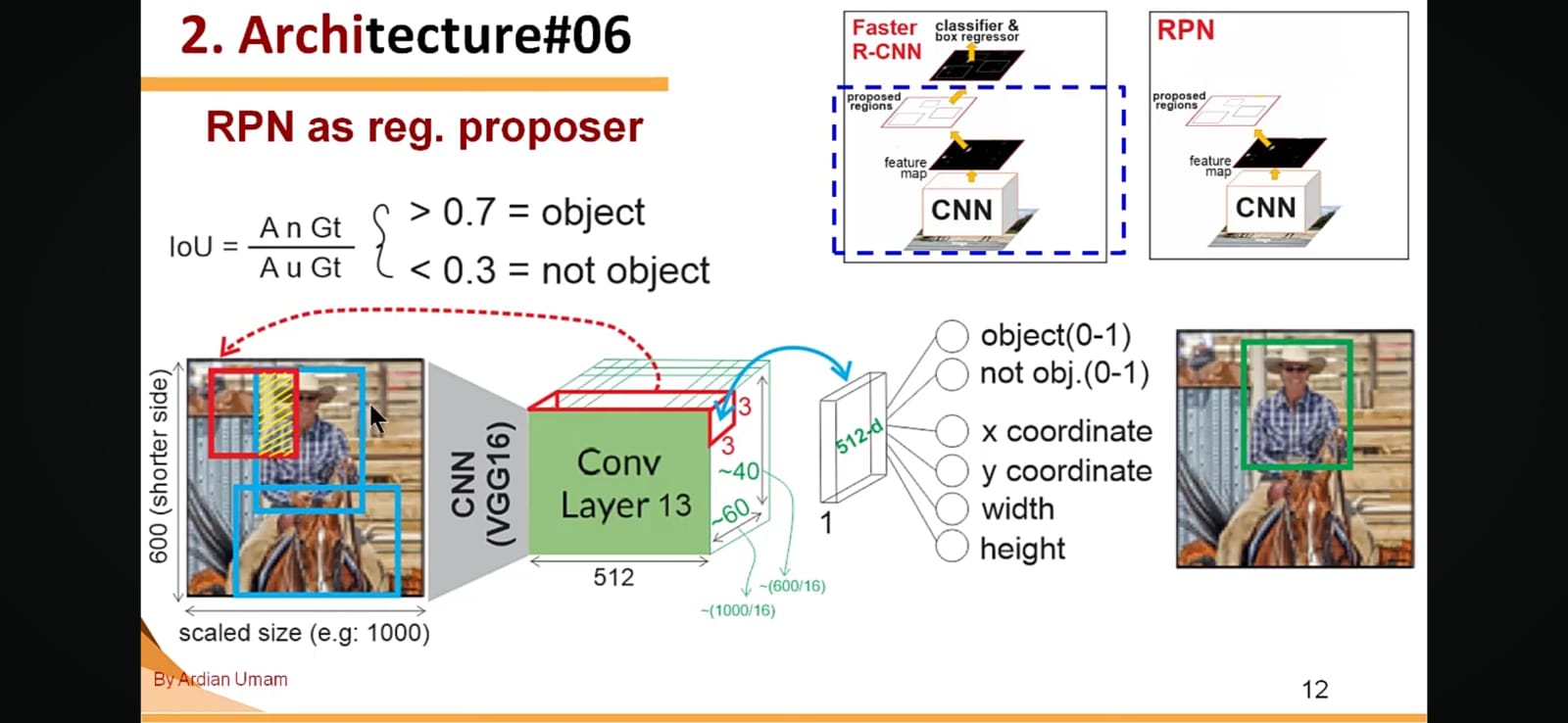
*j* : Indeks yang digunakan dalam penjumlahan atau perulangan (mewakili seluruh kelas).

*n* : Jumlah total kelas yang mungkin.

Dalam rangka memberikan ilustrasi terhadap konsep *softmax* suatu contoh dengan variabel *z* untuk *n*=3 kelas dan nilai 7.389, ​ 2.718, dan ​ 20.085 dengan mengacu pada rumus maka perhitungan distrubusi probabilitas *softmax* setiap kelas dapat diekspresikan sebagai berikut:

1. ***Region Proposal Network* (RPN)**

RPN merupakan bagian integral dari *ConvNet* yang terhubung dengan bagian utama (*backbone*) untuk menghasilkan proposal wilayah *(region proposals*). Proses ini melibatkan penelusuran gambar dan identifikasi kotak pembatas potensial (*bounding box*) yang memungkinkan adanya objek di dalamnya. RPN menghasilkan proposal dengan skor yang menunjukkan tingkat keyakinan bahwa suatu wilayah mungkin mengandung objek (Rahardjo, 2018).



Gambar 2.7 Arsitektur RPN

Pada gambar 2.7 Proses RPN bekerja dimulai dengan ekstraksi fitur menggunakan model VGG-16, dimana fitur-fitur hierarkis dari citra diekstraksi untuk representasi semantik yang tinggi. Kemudian RPN mengoperasikan fitur ini dan memasukkan *anchor boxes*, yang merupakan proposal wilayah yang akan diuji untuk deteksi objek. Lapisan konvolusi RPN menghasilkan *objectness score* atau skor klasifikasi dan parameter regresi untuk setiap *anchor box*. Skor klasifikasi mengindikasikan probabilitas *anchor box* mengandung objek, sedangkan parameter regresi membantu menyesuaikan posisi dan ukuran *anchor box*. Setelah itu, algoritma *Non-Maximum Suppression* (NMS) digunakan untuk menghilangkan tumpang tindih dan memilih proposal wilayah terbaik berdasarkan skor klasifikasi. Untuk pelatihan RPN, setiap jangkar diberikan skor objektivitas positif atau negatif berdasarkan *Intersection over Union* (IoU).

RPN bertindak sebagai generator proposal wilayah, menyajikan proposal wilayah yang kemudian diteruskan ke head deteksi objek untuk klasifikasi dan penyesuaian lebih lanjut. Dengan cara ini, RPN memainkan peran penting dalam memungkinkan sistem deteksi objek yang akurat dan efisien. Adapun rumus *Intersection over Union* (IoU), yang dapat dilihat pada rumus 2.4.

Keterangan:

IoU = *Intersect Over Union*



240

90

Gambar 2.8 Area (a) Tumpang Tindih dan (b) Gabungan

Berdasarkan ilustrasi gambar 2.8 diperoleh nilai *intersect* atau area tumpang tindih = 90 dan nilai *union* atau gabungan =240. Adapun cara menghitung nilai IoU adalah sebagai berikut.

Hasil yang di peroleh adalah 0,37 < 0,5 sehingga area prediksi dapat di jadikan sebagai *bounding box* dari objek. Parameter batas *bounding box* dapat di tentukan oleh masing-masing peneliti.

1. ***Feature Extraction***

Ekstraksi fitur atau yang dikenal sebagai *feature extraction*, merupakan langkah penting dalam proses mengidentifikasi dan mengekstrak informasi krusial dari data mentah seperti gambar, teks, atau suara. Tujuan utama dari ekstraksi fitur adalah untuk meningkatkan kinerja model *machine learning* dan meningkatkan akurasi prediksi. Dalam bidang *computer vision* melibatkan transformasi gambar menjadi representasi numerik yang dapat diproses oleh mesin, seperti menggunakan histogram warna, filter gabor, atau fitur khusus yang dihasilkan oleh *Convolutional Neural Network* (CNN).

Proses ekstraksi fitur ini memiliki peran kunci dalam pengolahan data, karena fitur-fitur yang berhasil diekstrak dapat memberikan informasi yang relevan untuk melatih model *machine learning*. Setelah fitur-fitur penting berhasil diidentifikasi, model *machine learning* dapat diinstruksikan menggunakan fitur-fitur tersebut untuk memprediksi hasil dengan tingkat akurasi yang lebih tinggi.

Dalam konteks deteksi objek menggunakan *Faster* R-CNN, ekstraksi fitur menjadi tahap awal yang signifikan. Pada tahap ini, fitur-fitur yang diekstraksi dari proposal daerah yang dihasilkan oleh *Region Proposal Network* (RPN) akan menjadi dasar untuk memprediksi label dan koordinat kotak pembatas pada gambar. Proses ini memastikan bahwa informasi yang relevan dari proposal daerah tersebut dapat dimanfaatkan secara efektif oleh model, memberikan kontribusi yang signifikan terhadap akurasi hasil prediksi (Edinesia, 2023).

1. ***Rol (Region of Interest) Pooling***

Setelah proposal wilayah dihasilkan oleh RPN, *RoI pooling* digunakan untuk mengekstrak fitur dari setiap proposal wilayah sehingga mereka dapat digunakan untuk klasifikasi objek*. Region of Interest* (RoI) *pooling* adalah operasi kunci dalam deteksi objek menggunakan *Convolutional Neural Networks* (CNN). Tujuannya adalah melakukan *max pooling* pada *input* non-seragam, menghasilkan peta fitur tetap, misalnya 7×7. *RoI pooling* memungkinkan penggunaan kembali peta fitur untuk semua proposal objek, meningkatkan kecepatan pelatihan dan pengujian. Lapisan ini memainkan peran vital dalam deteksi objek dengan memastikan ukuran yang konsisten dari proposal daerah. *RoI pooling* membantu mengatasi masalah kinerja dan kecepatan pemrosesan dalam tugas deteksi objek, mempercepat waktu pemrosesan dan meningkatkan efisiensi sistem secara keseluruhan (Grel, 2017).

1. **Klasifikasi dan *Bounding Box Regression***

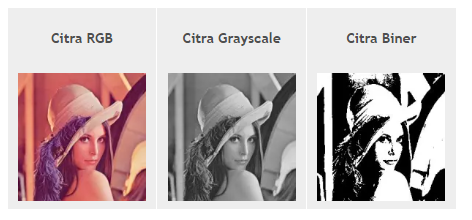
Proposal wilayah yang telah dihasilkan kemudian diproses lebih lanjut melalui lapisan-lapisan klasifikasi dan regresi. Lapisan klasifikasi menentukan kelas objek yang mungkin ada dalam wilayah tersebut, sedangkan lapisan regresi menyesuaikan kotak pembatas atau *bounding box* agar lebih akurat. Pada masalah deteksi objek, *output* yang dihasilkan berupa kotak pembatas (*bounding box*) hasil sistem prediksi terhadap objek yang telah ditentukan. *Bounding box* ini merepresentasikan posisi objek dalam sebuah gambar. Untuk *model* penemuan deteksi objek yang telah dilatih dengan beberapa cara, salah satu caranya adalah dengan menggunakan metode *Intersection Over Union* (IOU). IOU memanfaatkan kotak pembatas yang terdapat pada gambar.

IoU digunakan untuk mengevaluasi sejauh mana proposal wilayah (*bounding box*) yang dihasilkan oleh algoritma RPN cocok dengan objek yang sebenarnya. Proposal dengan IoU tinggi lebih mungkin mencakup objek dengan benar, dan dapat dipilih untuk langkah selanjutnya dalam sistem deteksi objek. Selain itu IoU digunakan dalam algoritma NMS untuk mengatasi masalah tumpang tindih antara beberapa proposal wilayah. NMS memastikan bahwa hanya proposal wilayah dengan IoU tertinggi yang tetap dipertahankan, sementara proposal lain yang tumpang tindih lebih rendah dihapus (Salim, 2020).

1. **Citra Digital**

Citra digital adalah representasi dari gambar nyata dalam bentuk serangkaian angka yang dapat disimpan dan diproses oleh komputer. Gambar digital terbagi menjadi area kecil yang dikenal sebagai piksel, yang merupakan elemen gambar terkecil. Setiap piksel memuat nilai numerik yang menggambarkan tingkat kecerahan (intensitas cahaya) atau warna (Hanggara *et al*., 2021).

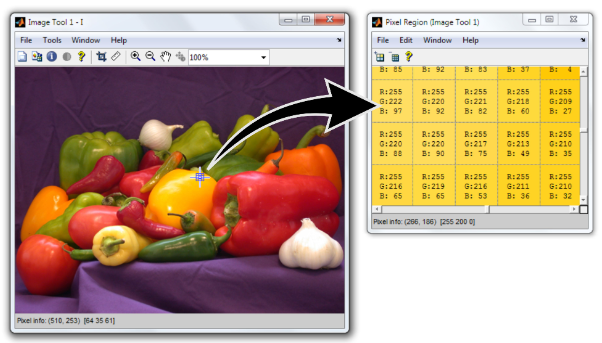
Secara literal, citra (*image*) adalah representasi visual pada bidang dua dimensi. Dari perspektif matematika, citra adalah fungsi berkelanjutan dari intensitas cahaya pada bidang dua dimensi. Cahaya dari sumber menyinari objek, yang kemudian memantulkan sebagian dari sinar tersebut. Pantulan cahaya ini kemudian diambil oleh alat optik, merekam bayangan objek dan membentuk citra. Citra digital dapat diartikan sebagai suatu fungsi dua dimensi *f*(*x*,*y*) dengan ukuran *M* baris dan *N* kolom, dimana *x* dan *y* adalah posisi koordinat spasial, dan *f* adalah amplitudo pada titik koordinat (*x*,*y*), yang disebut sebagai intensitas atau tingkat keabuan dari citra pada titik tersebut (Basyir, 2021). Gambar 2.9 adalah contoh citra digital.



Gambar 2.9 Citra Digital

1. **Citra RGB**

Gambar RGB terdiri dari tiga komponen warna utama yaitu merah, hijau, dan biru. Setiap komponen ini diwakili oleh satu kanal dalam gambar. Masing-masing kanal memiliki kedalaman bit sebesar 8-bit, sehingga memiliki nilai intensitas piksel yang bervariasi dari 0 hingga 255. Pada kanal merah, nilai 255 merepresentasikan warna merah sempurna, sedangkan nilai 0 merepresentasikan warna hitam sempurna. Pada kanal hijau, nilai 255 merepresentasikan warna hijau sempurna, dan nilai 0 merepresentasikan warna hitam sempurna. Begitu juga pada kanal biru, nilai 255 merepresentasikan warna biru sempurna, dan nilai 0 merepresentasikan warna hitam sempurna. Dengan kombinasi intensitas dari ketiga kanal tersebut, kita dapat menghasilkan berbagai variasi warna yang terbentuk dari campuran merah, hijau, dan biru, menciptakan citra warna penuh dengan kekayaan visual.

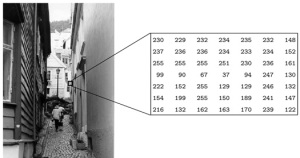
. 

Gambar 2.10 Representasi Citra RGB

Pada Gambar 2.10 merupakan representasi warna citra digital RGB. Suatu piksel yang memiliki nilai intensitas warna sebesar 255 pada kanal kuning menunjukkan pixel warna yang tersusun dari R,G dan B.

1. **Citra *Grayscale***

Citra *grayscale* adalah citra digital yang memiliki satu nilai kanal pada setiap pikselnya, dimana nilai merah (*red*), hijau (*green*), dan biru (*blue*) semuanya sama. Nilai-nilai ini mencerminkan intensitas warna. Citra ini menghasilkan gambar dalam skala warna abu-abu, dengan variasi mulai dari warna hitam pada bagian yang memiliki intensitas terlemah hingga warna putih pada bagian yang memiliki intensitas terkuat (Rizky Pratama *et al*., 2020). Contoh citra *grayscale* dapat dilihat pada Gambar 2.11.



Gambar 2.11 Citra Grayscale

Pada citra dengan tipe *greyscale*, tiap piksel merepresentasikan tingkat keabuan dengan nilai berkisar antara 0 (hitam) hingga 255 (putih). Rentang nilai ini menunjukkan bahwa setiap piksel menggunakan 8 bit atau 1 *byte* (Pamungkas, 2016). Visualisasi variasi warna *grayscale* dapat dilihat pada Gambar 2.12. Adapun rumus dari citra *grayscale* terdapat pada rumus 2.5.

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  |  |  |
| 0 | 1 | 2 | ……….. | 128 | ……….. | ………. | 255 |

Gambar 2.12 Visualisasi Aras Keabuan

(2.5)

Keterangan:

*R = Red, G = Green, B = Blue, X =* Nilai Intensitas warna

Sebagai contoh penerapan rumus 2.5 diberikan nilai R=100, G=150, B=200 dapat dihitung:

Maka, nilai warna menjadi RGB (150, 150, 150) yang menghasilkan warna abu-abu dengan intensitas keabuan yang seragam pada piksel tersebut.

1. **Format Citra Digital**
2. ***Digital Imaging and Communications in Medicine***

Standar Komunikasi dan Citra Digital dalam Kedokteran (*Digital Imaging and Communications in Medicine* (DICOM) menentukan protokol pertukaran data yang tidak berkepemilikan, format citra digital, dan struktur file untuk citra biomedis dan informasi terkait citra. DICOM memiliki file ekstensi .dcm yang tidak dapat dibaca oleh sistem operasi pada PC yang banyak digunakan seperti Windows ataupun Linux. Oleh karena itu, untuk membaca file DICOM harus digunakan DICOM *viewer* yang dapat diunduh secara gratis dari internet. DICOM memberikan informasi teknis terperinci yang dapat digunakan dalam spesifikasi antarmuka untuk mengaktifkan konektivitas jaringan di antara berbagai produk vendor. Standar ini menjelaskan cara memformat dan menukar gambar medis dan informasi terkait, baik di dalam rumah sakit maupun di luar rumah sakit (misalnya, teleradiologi, telemedis).

1. ***Joint Photographic Experts Group***

Format *Joint Photographic Experts Group* atau JPEG (.jpg atau .jpeg) adalah format standar yang paling umum digunakan untuk mentransmisikan informasi dalam bentuk gambar. Kedua istilah, JPEG dan JPG, merujuk pada gambar raster, yang berarti gambar tersebut bukanlah vektor. Karakteristik utamanya termasuk gradasi warna yang halus, bayangan yang halus, serta garis dan bentuk yang mungkin kurang tajam. Seiring gambar berbasis piksel, gambar jenis ini dapat mengalami degradasi kualitas ketika diperbesar, terlihat bergerigi, dan mengekspos piksel individu.

Gambar JPEG biasanya ditentukan dan ditampilkan pada resolusi tertentu, diukur dalam titik per inci (dpi). Semakin tinggi resolusi dpi suatu gambar raster, semakin jelas kualitas gambar tersebut. Gambar raster sering digunakan sebagai standar di web, membuatnya mudah ditemukan dan diakses secara online.

1. ***Portable Network Graphics***

PNG (*Portable Network Graphics*) adalah format citra bitmap dengan kompresi data lossless yang diciptakan sebagai penerus format GIF. PNG mendukung true color (16 juta warna), unggul daripada GIF yang hanya mendukung 256 warna. Format .png adalah format penyimpanan citra terkompresi yang mendukung citra *grayscale*, palet warna, dan citra *full-color*. PNG juga dapat menyimpan informasi kanal alpha dengan kedalaman per kanal sebesar 1 hingga 16 bit.

PNG cocok untuk citra dengan area warna seragam dan paling sesuai untuk pengeditan citra, sementara format *lossy* seperti JPG lebih baik untuk distribusi final fotografi. Meskipun sebelumnya tidak didukung oleh beberapa *browser*, sekarang format PNG didukung oleh semua *browser* modern. Fitur khususnya mencakup dukungan untuk informasi warna hingga 48 bit dan penggunaan skema 2D yang meningkatkan kecepatan tampilan citra. Gamma koreksi memastikan nilai-nilai yang ditampilkan tetap konsisten di berbagai platform (Labbay, 2020).

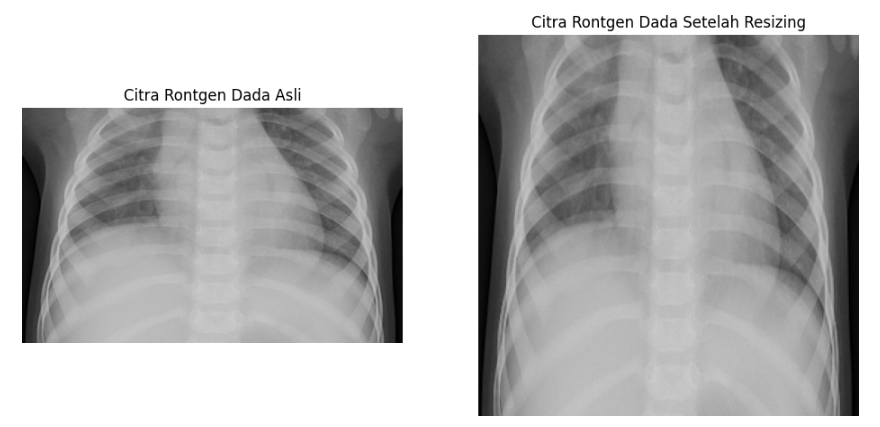
1. **Pengolahan Citra Digital**

Pengolahan citra digital merupakan sebuah cabang ilmu yang mempelajari tentang teknik-teknik mengelola citra. Citra yang dimaksud adalah gambar diam (foto) maupun gambar bergerak. sedangkan digital yang mempunyai maksud bahwa pengolahan citra atau gambar dilakukan secara digital menggunakan komputer. Citra merupakan fungsi kontinu dengan intensitas cahaya pada bidang dua dimensi, agar dapat diolah menggunakan komputer digital maka sebuah citra harus dipresentasikan secara numerik yang dapat diwakili oleh sebuah matriks dua dimensi f (x,y) yang terdiri dari M kolom dan N baris. Setiap foto dalam bentuk citra digital dapat diolah menggunakan perangkat lunak tertentu (Basyir, 2021).

Penggunaan citra digital mencakup berbagai aplikasi yang sangat relevan di berbagai sektor. Kemampuan untuk memproses dan menganalisis citra digital membuka peluang untuk inovasi dan perbaikan di berbagai industri serta disiplin ilmu. Pengolahan citra digital bertujuan utama untuk memproses dan memanipulasi citra digital sesuai keperluan tertentu. Proses ini melibatkan serangkaian operasi dengan tujuan meningkatkan kualitas citra, mengekstraksi informasi penting, serta membuat citra lebih mudah dipahami atau digunakan dalam berbagai aplikasi. Beberapa tujuan khusus melibatkan peningkatan kualitas citra, ekstraksi fitur, segmentasi citra, kompresi citra, dan pengenalan pola (Dijaya, 2019).

1. ***Resizing***

Tahapan ini terjadi proses normalisasi dimensi citra rontgen dada, yaitu proses pembesaran atau pengecilan dimensi citra rontgen dada menjadi dimensi yang telah ditentukan. Tujuannya, untuk menyamakan dimensi rontgen dada dari tiap citra yang dimasukan, sehingga pada proses ekstrasi citra nanti tidak ada perbedaan dimensi dari matriks data citra rontgen dada. Gambar 2.13 menunjukkan perbedaan citra yang mengalami *resizing*.



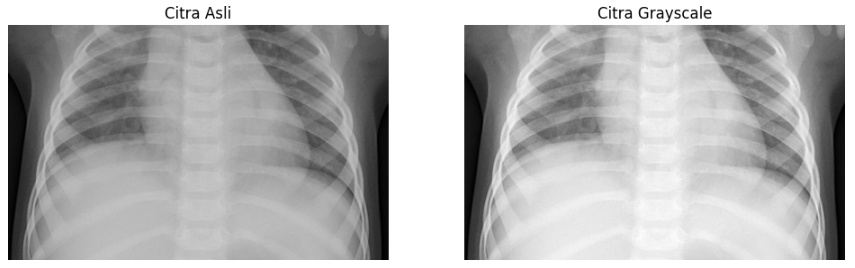
Gambar 2.13 Perbedaan Citra Asli dan Citra Resize

1. **Normalisasi Citra**

Normalisasi citra adalah proses penting dalam pemrosesan citra yang bertujuan untuk mengubah nilai piksel dalam citra agar sesuai dengan skala tertentu atau memiliki distribusi intensitas yang diinginkan. Tujuan normalisasi mencakup stabilisasi kontras untuk meningkatkan ketajaman citra, serta persiapan data untuk model, khususnya dalam konteks *deep learning* atau *machine learning* pada citra, agar data memiliki skala yang seragam dan menghindari masalah divergensi gradien selama pelatihan. Selain itu, normalisasi citra juga bermanfaat untuk perbandingan data dari berbagai sumber, analisis statistik, penanganan kondisi pencahayaan yang tidak konsisten, pengurangan variabilitas dalam data, dan mengurangi dampak perbedaan skala piksel yang bervariasi. Proses ini memiliki aplikasi yang luas dalam meningkatkan kualitas dan konsistensi citra untuk berbagai keperluan analisis dan pemrosesan.

1. **Konversi Citra**

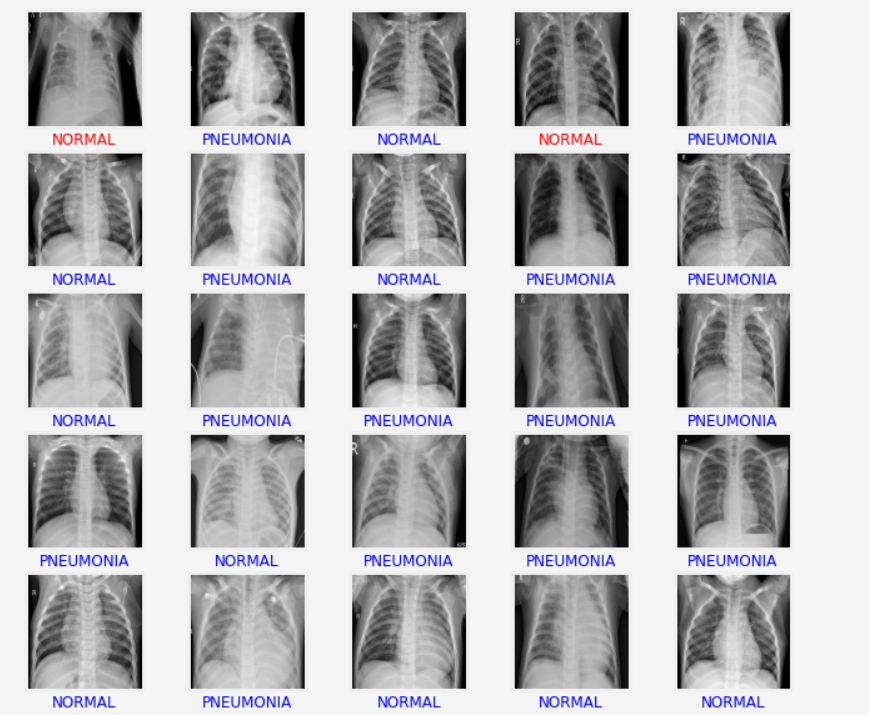
Konversi pada penelitian ini dilakukan dengan mengubah citra berwarna menjadi citra abu-abu, sehingga hanya terdapat dua warna yaitu putih dan hitam. Konversi citra ini dilakukan untuk memudahkan pemrosesan citra pada jaringan CNN, serta untuk mengurangi dimensi masukan pada jaringan. Gambar 2.14 menunjukkan Citra *Grayscale*.



Gambar 2.14 Citra RGB dan Citra Grayscale

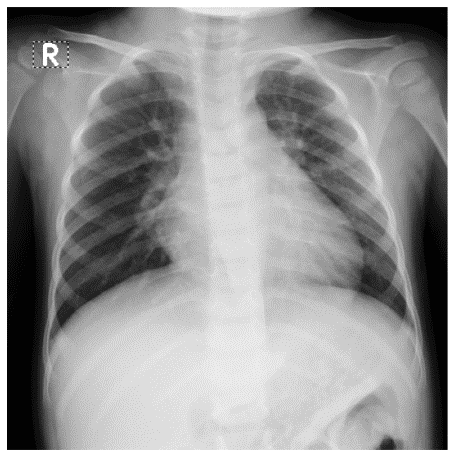
1. ***Image Classification***

Klasifikasi citra (*image classification*) adalah kegiatan mengkategorikan suatu gambar kedalam suatu kategori dari kumpulan kategori-kategori, misalnya seperti yang ditunjukan pada Gambar 2.15 yang mengklasifikasikan antara penyakit penumonia dan normal. Tujuannya adalah agar komputer dapat mengenali objek yang ada pada gambar (Bagas *et al*., 2021). Pada pengklasifikasian yang digunakan pada algoritma *Faster Region Convotional Neural Network* (*Faster* RCNN) menggunakan *bounding box regression* dan *softmax classification.*



Gambar 2.15 Klasifikasi Citra Rontgen Dada Pneumonia & Normal

1. **Citra Rontgen Dada**



Gambar 2.16 Contoh Citra Rontgen Dada

Rontgen dada atau *Chest X-Ray* (CXR) merupakan salah satu aplikasi pencitraan medis yang paling sering digunakan untuk mendeteksi kelainan, terutama tumor pada paru-paru. Proses diagnosis kelainan ini saat ini masih sangat bergantung pada pengamatan visual dokter dalam membaca hasil citra CXR, sehingga penilaian cenderung bersifat subyektif tergantung pada interpretasi masing-masing dokter (Pujitresnani, 2015). Citra Rontgen dapat dilihat pada Gambar 2.16.

Citra Rontgen dada atau yang sering disebut sebagai *Chest X-ray* (CXR) adalah suatu proyeksi radiografi dari dada yang digunakan untuk mendiagnosis berbagai kondisi yang memengaruhi bagian dada, isi, dan struktur di sekitarnya. Proses pengambilan gambar dada menggunakan radiasi terionisasi dalam bentuk sinar-X. Dosis radiasi yang umumnya digunakan pada orang dewasa untuk menghasilkan gambar radiografi ini adalah sekitar 0.06 mSv.

Foto dada digunakan untuk mendiagnosis berbagai kondisi yang melibatkan dinding dada, tulang-tulang di dalamnya, dan struktur lain di dalam kavitas dada, termasuk paru-paru, jantung, dan saluran-saluran besar. Kondisi-kondisi seperti pneumonia dan gagal jantung kongestif sering dapat didiagnosis melalui pemeriksaan gambar dada. Pencitraan ini membantu para profesional medis untuk mengevaluasi dan memahami kondisi pasien dengan memberikan gambaran visual dari organ-organ di dalam dada (Fauzy, 2019).

1. **Python**

Python adalah bahasa pemrograman tingkat tinggi yang mampu menjalankan berbagai instruksi multifungsi secara langsung dengan pendekatan berorientasi objek. Sejumlah perusahaan juga mengembangkan Python secara komersial untuk menyediakan layanan, seperti Anaconda, sebuah aplikasi pemrograman Python yang dilengkapi dengan berbagai alat bantu. Python merupakan salah satu bahasa pemrograman dan penunjang pada perkembangan FOSS dan Linux (Guntara, 2023b).

Python merupakan bahasa pemrograman yang sangat populer dalam bidang data mining dan pemrosesan bahasa alami (*Natural Language Processing*). Python menyediakan beragam pustaka (*library*) yang mendukung pengembangan aplikasi di kedua bidang tersebut, seperti *Scikit-learn*, NLTK (*Natural Language Toolkit*), Gensim, dan sebagainya. Dalam pemrosesan bahasa alami, Python menjadi bahasa pemrograman yang sangat relevan karena kemampuan dan pustaka yang dimilikinya (Guntara, 2023a). Pada penelitian ini di gunakan python 3.9.0.

1. **TensorFlow**

TensorFlow adalah perpustakaan perangkat lunak yang dikembangkan oleh tim Google Brain dan pertama kali dirilis pada tahun 2015 dengan lisensi Apache 2.0. Penggunaan TensorFlow oleh Google tidak hanya terbatas pada riset, tetapi juga melibatkan produksi layanan mereka. TensorFlow dikenal sebagai salah satu perangkat lunak terkemuka dalam pengembangan *machine learning, deep learning*, dan kecerdasan buatan.

Keunggulan TensorFlow mencakup kemampuannya untuk beroperasi lintas platform, mendukung banyak CPU dan GPU. Hal ini memberikan fleksibilitas yang signifikan bagi pengembang dalam menciptakan solusi *deep learning*. TensorFlow dapat diimplementasikan pada berbagai perangkat, termasuk Linux, macOS, Windows, serta perangkat mobile seperti Android dan iOS. Lebih jauh, TensorFlow dapat digunakan dalam berbagai konteks, termasuk aplikasi desktop, *mobile*, server, dan *web* (Aningtiyas, 2020). TensorFlow yang di gunakan pada penelitian ini adalah TensorFlow 2.5.0.

1. **Google Collaboratory**

Google Colaboratory atau lebih dikenal sebagai Google Colab, merupakan produk yang dikembangkan oleh tim Google Research. Ini adalah layanan jupyter notebook berbasis *cloud* yang memungkinkan peneliti untuk melatih model *machine learning* dan *deep learning* menggunakan CPU, GPU, dan TPU. Layanan ini memberikan bantuan kepada peneliti yang memiliki keterbatasan perangkat untuk pengembangan kecerdasan buatan secara gratis(Guntara, 2023a).

Google Colab menyediakan akses gratis untuk penggunaan GPU dan TPU. Diketahui bahwa proses pelatihan model *deep learning* menggunakan komputasi CPU memakan waktu yang cukup lama. Oleh karena itu, kehadiran GPU atau TPU sangat penting, memungkinkan pelatihan model dalam waktu yang lebih singkat. Sebagai contoh, jika melatih suatu model memerlukan waktu berjam-jam dengan menggunakan CPU, pengguna dapat melakukannya dalam beberapa menit atau detik saja dengan menggunakan GPU atau TPU (Guntara, 2023b).

1. ***Rational Unified Process* (RUP)**

Menurut IBM, *Rational Unified Process* (RUP) adalah kerangka proses yang menyediakan simulasi sistem pada industri untuk sistem, *software*, implementasi, dan manajemen proyek yang efektif. RUP adalah salah satu dari sekian banyak proses yang terdapat di dalam *Rational Process Library*, yang memberikan simulasi terbaik untuk pengembangan atau kebutuhan proyek.

Proses RUP merupakan proses pengembangan perangkat lunak yang dilakukan secara *iteratif* (berulang) dan *inkremental* (bertahap dengan progres menaik). Iteratif bisa dilakukan dalam setiap tahap atau *iteratif* tahap pada proses pengembangan perangkat lunak untuk menghasilkan perbaikan fungsi yang *inkremental* (bertambah naik) dimana setiap iterasi akan memperbaiki iterasi berikutnya. Salah satu *unifield process* yang dikenal adalah *Rational Unifield Process* (RUP) (Rahardjo, 2018). Pada Gambar 2.17 adalah alur kerja RUP yang mempunyai beberapa tahapan, yaitu:



Gambar 2.17 Alur Kerja Rational Unified Process

1. *Inception* (permulaan)

*Inception* adalah langkah awal dalam mengidentifikasi dan mengembangkan sistem. Kegiatan pada tahap ini melibatkan analisis sistem yang sudah ada, perumusan sistem target, penentuan arsitektur global target, identifikasi kebutuhan, formulasi persyaratan (fungsional, performansi, keamanan, GUI, dsb.), perumusan kebutuhan pengujian (unit, integrasi, sistem, performansi, fungsionalitas, keamanan, dsb.), pembuatan UML diagram, dan dokumentasi.

2. *Elaboration* (perencanaan)

Pada tahap *elaboration* dilakukan perencanaan arsitektur sistem. Pada perencanaan arsitektur sistem dimulai dari melakukan spesifikasi fitur perangkat lunak yang akan digunakan hingga analisis dan desain sistem serta implementasi sistem yang fokus pada arketipe sistem (*prototype*).

3. *Construction* (konstruksi)

Pada tahap *construction* dilakukan pengimplementasian rancangan sistem yang telah dibuat. Pada tahap ini akan fokus pada implementasi perangkat lunak pada kode program serta melakukan pengujian sistem.

4. *Transition* (transisi)

Pada tahap *transition* dilakukan *deployment* perangkat lunak. Pada tahap ini dilakukan evaluasi sistem yang telah dirancang, dianalisis dan dibangun. Tahapan ini bertujuan untuk melakukan pelatihan pada user yang akan menggunakan sistem serta melakukan pengujian sistem apakah sudah memenuhi harapan *user*.

1. ***Confusion Matrix***

*Confusion matrix* adalah metrik yang sangat umum digunakan dalam menangani masalah klasifikasi. Ini dapat digunakan baik untuk klasifikasi biner maupun untuk masalah klasifikasi dengan banyak kelas (Kulkarni *et al*., 2021). *Confusion Matrix* merupakan matriks N x N yang digunakan untuk mengevaluasi kinerja model klasifikasi, dengan N adalah jumlah kelas target. Matriks membandingkan nilai target sebenarnya dengan nilai yang diprediksi oleh model pembelajaran mesin. Ini memberikan pandangan *holistic* tentang seberapa baik kinerja model klasifikasi dan jenis kesalahan apa yang dibuatnya.

**Tabel 2.1 Confussion Matrix**

|  |  |  |  |
| --- | --- | --- | --- |
|  | | *Actual class* | |
| Relevan | Tidak Relevan |
| *Predict Class* | Relevan | *True Positive* (TP) | *False Positive* (FP) |
| Tidak Relevan | *False Negative* (FN) | *True Negative* (TN) |

Jika prediksi bernilai positif dan data bernilai positif maka di masukan dalam kelas *True Positive* (TP). Namun, jika data tersebut diprediksi negatif, maka akan dihitung sebagai *False Negative* (FN). Untuk data negatif yang diprediksi negatif, itu akan dihitung sebagai *True Negative* (TN), tetapi jika data tersebut diprediksi positif, akan dihitung sebagai *False Positive* (FP). Berdasarkan tabel tersebut dapat dihitung nilai *precision, sensitivity/recall, specificity*,*accuracy* dan *F-measure* menggunakan rumus-rumus berikut:

1. ***Precision* (Presisi)**

Nilai *precision* mengukur seberapa banyak prediksi positif yang benar dari total prediksi positif yang dilakukan oleh model. [Rumus *precision*:](https://rey1024.medium.com/mengenal-accuracy-precission-recall-dan-specificity-serta-yang-diprioritaskan-b79ff4d77de8)

1. ***Sensitivity (Sensitivitas* atau *Recall):***

Nilai *recall* mengukur seberapa banyak prediksi positif yang benar dari total data yang sebenarnya positif. [*Recall* dapat dihitung dengan rumus](https://rey1024.medium.com/mengenal-accuracy-precission-recall-dan-specificity-serta-yang-diprioritaskan-b79ff4d77de8):

1. ***Specificity* (Spesifisitas)**

*Specificity* atau Spesifisitas merupakan metrik evaluasi yang menunjukkan seberapa efektif suatu model dalam mengklasifikasikan kelas negatif secara akurat. Adapun rumus *specificity* adalah:

1. ***Accuracy*  (Akurasi):**

Nilai *accuracy*  mengukur seberapa banyak prediksi yang benar dari total data yang diprediksi. *Accuracy*  dapat dihitung dengan rumus:

1. ***F-measure* (*F1 Score*):**

Nilai *F-measure* adalah *harmonic mean* dari *precision* dan *recall*. *F-measure* dapat dihitung dengan rumus:

Keterangan:

TN *= True Negative* TP *= True Positive*

FN *= False Negative* FP = *False Positive*

1. ***Flowchart***

*Flowchart* adalah representasi grafis yang memperlihatkan urutan atau langkah-langkah dalam sebuah program atau prosedur sistem secara logis. *Flowchart* atau sering disebut sebagai bagan alir, merupakan gambaran visual yang menggambarkan urutan algoritma-algoritma dalam suatu program. Diagram ini menyajikan arah aliran program secara visual, memperlihatkan aliran atau jalur eksekusi dari satu langkah ke langkah berikutnya (Yulianeu *et al*, 2022).

*Flowchart* adalah grafik yang menunjukkan langkah-langkah dan urutan prosedur suatu program. Digunakan dalam analisis, perancangan, dan pengkodean untuk memecah masalah menjadi bagian-bagian kecil. *Flowchart* memfasilitasi penyelesaian masalah dan evaluasi lanjutan. Sebagai diagram dengan simbol-simbol grafis, *Flowchart* menggambarkan aliran proses dan langkah-langkah suatu program. Fungsinya membantu pemahaman dan visualisasi proses produksi, mempermudah pengelolaan data melalui diagram alir atau *Flowchart* (Malabay, 2016).

1. ***Graphical User Interface* (GUI)**

GUI atau *Graphical User Interface* adalah antarmuka pengguna yang memungkinkan pengguna untuk berinteraksi dengan perangkat elektronik melalui ikon, menu, dan indikator visual lainnya. GUI pertama kali dikembangkan pada tahun 1981 oleh Alan Kay, Douglas Engelbart, dan sekelompok peneliti lainnya di Xerox PARC. GUI menampilkan objek yang menyampaikan informasi dan mewakili tindakan yang dapat diambil oleh pengguna. Objek-objek tersebut berubah warna, ukuran, atau visibilitas saat pengguna melakukan interaksi seperti mengkliknya. Dalam sistem operasi, GUI digunakan untuk memudahkan pengguna dalam menjalankan perintah seperti membuka, menghapus, dan memindahkan file (Mushtaq, 2021).

1. **Pengujian *Black Box***

Pengujian perangkat lunak memiliki tujuan untuk mendeteksi kegagalan, memungkinkan identifikasi cacat, perbaikan, dan peningkatan kualitas perangkat lunak. Metode pengujian *black box* adalah teknik di mana pengujian dilakukan tanpa memperhatikan struktur internal dari komponen atau sistem. Dalam pengujian *black box*, fokusnya hanya pada hubungan antara *input* dan output. Pendekatan ini memastikan bahwa kebutuhan pelanggan terpenuhi dengan menguji *input* baik yang valid maupun tidak valid (Masripah *et al*., 2019).

Penguji yang menggunakan pengujian *black box* tidak diberikan akses untuk mengetahui kode sumber dan arsitektur sistem. Pemodelan *black box* disesuaikan dengan tingkat akurasi yang diinginkan dan struktur opsional dipilih untuk memetakan data yang diukur dari sistem, termasuk *input* dan *output*.

Pengujian *black box* memiliki beberapa tujuan yang melibatkan pencarian fungsi yang salah atau hilang, identifikasi kesalahan antarmuka, deteksi kesalahan dalam struktur data atau akses ke database eksternal, pengamatan terhadap kesalahan kinerja, evaluasi inisialisasi dan terminasi, validasi fungsional, penilaian terhadap sensitivitas sistem terhadap nilai *input* tertentu, serta pengetesan batasan suatu data (Parlika *et al*., 2020).

**BAB III  
METODOLOGI PENELITIAN**

1. **Metode Pengumpulan Data**

Beberapa metode yang digunakan dalam pengumpulan data meliputi:

1. Studi Literatur

Peneliti menggali gambaran dan penjelasan pengolahan citra menggunakan metode Jaringan Saraf Tiruan yang lebih lengkap dan spesifik melalui tinjauan literatur terdahulu yang telah berhasil diimplementasikan oleh berbagai sumber. Kajian pustaka dilakukan dengan cara mengumpulkan dan mempelajari literatur terkait pengolahan citra menggunakan *Faster RCNN*, seperti buku teks, makalah ilmiah, dan sumber informasi online.

1. Wawancara

Pengumpulan informasi mengenai gambaran umum, keterangan, alur sistem, dan aspek lainnya yang diperlukan untuk memahami cara kerja sistem. Data yang diperoleh berasal dari interaksi dengan ahli radiologi untuk pembacaan hasil citra rontgen pada dada dan pengumpulan informasi mengenai pneumonia melalui wawancara dengan Ibu Gita Aprilia Laasamana, A.Md. Rad, S.Si bagian radiologi di Rumah Sakit Umum Daerah Kabupaten Konawe Selatan.

1. Dataset

Dalam rangka penelitian, dilakukan pengumpulan dataset yang akan digunakan dalam pelatihan menggunakan metode *Faster Region Convolutional Neural Network* (*Faster* RCNN). Jumlah dataset yang dihimpun mencapai 5.881 citra, yang diperoleh dari National Institutes of Health Clinical Center (URL dataset dapat diakses pada link: https://www.kaggle.com/competitions/rsna-pneumonia-detection-challenge/data) dengan total 5.865 data, Rumah Sakit Umum Daerah Kota Kendari sebanyak 10 data terdiri dari 5 data normal dan 5 data penumonia, Rumah Sakit Umum Daerah Kabupaten Konawe Selatan. Total 14 data yakni 3 data normal dan 11 data pneumonia. Adapun dataset citra rontgen pneumonia dan normal dapat dilihat pada Gambar 3.1, Gambar 3.2, Gambar 3.3, Gambar 3.4, Gambar 3.5, dan Gambar 3.6.



Gambar 3.1 Dataset Citra Rontgen Dada Normal



Gambar 3.2 Dataset Citra Rontgen Dada Pneumonia



Gambar 3.3 Citra Rontgen Normal RSUD Konawe Selatan



Gambar 3.4 Citra Pneumonia RSUD Konawe Selatan



Gambar 3.5 Citra Normal RSUD Kota Kendari



Gambar 3.6 Citra Pneumonia RSUD Kota Kendari

1. **Metode Pengembangan Sistem**

Metode pengembangan sistem adalah pendekatan atau strategi yang digunakan dalam merencanakan, merancang, dan mengimplementasikan sistem informasi atau perangkat lunak. Setiap metode memiliki proses dan langkah-langkah tertentu yang harus diikuti untuk mencapai tujuan pengembangan sistem . Metode pengembangan perangkat lunak yang digunakan dalam penelitian ini adalah *Rational Unified Process* (RUP). RUP adalah salah satu metode pengembangan perangkat lunak yang mengedepankan pendekatan iteratif dan inkremental. Keunggulan RUP juga terletak pada kemampuannya untuk beradaptasi dengan perubahan kebutuhan pengguna atau kebijakan proyek secara efisien melalui pendekatan inkremental. Setiap iterasi menambahkan fungsionalitas baru dan memperbaiki fitur yang sudah ada berdasarkan umpan balik dari pemangku kepentingan. *Rational Unified Process* (RUP) terdiri dari empat tahap utama:

1. *Inception* (Permulaan)

Pada tahap ini, dilakukan identifikasi aplikasi melalui analisis kebutuhan aplikasi. Pada tahap ini dilakukan kajian terhadap penelitian yang terkait dengan informasi jadwal kegiatan dengan menggunakan algoritma *Faster Region Convotional Neural Network*. Tujuan dari tahap ini adalah mengidentifikasi kebutuhan dasar dan memahami konsep dasar penelitian.

1. *Elaboration* (Perluasan/Perencanaan)

Setelah menetapkan ruang lingkup penelitian, tahap *Elaboration* melibatkan perancangan dan analisis sistem dengan menggunakan UML (*Unified Modeling Language*), termasuk pembuatan diagram *use case, activity, class*, dan *sequence*. Seiring dengan itu, analisis sistem pada tahap ini mencakup perancangan tampilan antarmuka (GUI) dari sistem yang akan dikembangkan. Proses ini membantu merinci struktur sistem dan menyusun rencana perancangan antarmuka pengguna yang optimal. Perhatian khusus diberikan pada aspek tampilan sistem untuk memastikan keterpaduan antara fungsionalitas dan desain antarmuka yang efektif.

1. *Construction* (Konstruksi)

Pada tahap *Construction*, proses pengembangan perangkat lunak dilakukan secara menyeluruh. Ini melibatkan pembuatan antarmuka aplikasi, penulisan kode, dan implementasi langkah-langkah seperti akuisisi citra, konversi citra ke *grayscale*, *cropping*, normalisasi ukuran, dan pengenalan pola huruf. Fokus pada tahap ini adalah menghasilkan perangkat lunak yang sesuai dengan fungsinya.

1. *Transition* (Transisi)

Fase *transition* berfokus pada penerapan sistem ke lingkungan pengguna. Sistem diuji untuk memastikan kinerjanya yang baik, dan masalah yang muncul selama pengujian diperbaiki. Dalam penelitian ini, aplikasi digunakan untuk melakukan pengujian dan memastikan bahwa semua aspeknya berfungsi sebagaimana mestinya.

1. **Waktu dan Tempat Penelitian**
   * 1. **Waktu Penelitian**

Waktu pelaksanaan penelitian yang dilakukan berlangsung selama 4 bulan dan dimulai pada bulan Desember 2023 - Maret 2024. Pada Tabel 3.1 adalah rincian kegiatan penelitian yang dilaksanakan.

**Tabel 3.1 Waktu Pelaksanaan Penelitian**

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **No.** | **Jadwal**  **Kegiatan** | **Waktu Penelitian 2023-2024** | | | | | | | | | | | | | | | |
| **Desember** | | | | **Januari** | | | | **Februari** | | | | **Maret** | | | |
| **1.** | *Inception* |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| **2.** | *Elaboration* |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| **3.** | *Construction* |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| **4.** | *Transition* |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |

* + 1. **Tempat Penelitian**

Penelitian tugas akhir ini dilakukan di Laboratorium *Artificial Intelligence*, Jurusan Teknik Informatika, Fakultas Teknik, Universitas Halu Oleo.

1. **Analisis Kebutuhan Sistem**

Analisis sistem adalah langkah yang bertujuan untuk memahami dan mengamati semua komponen yang terkait dalam suatu sistem. Dalam konteks analisis sistem, terdapat beberapa aspek yang dibahas, termasuk analisis kebutuhan fungsional dan non-fungsional. Tujuan dari analisis kebutuhan sistem ini adalah untuk mengidentifikasi permasalahan yang mungkin muncul dalam sistem tersebut. Ini mencakup integrasi aplikasi, termasuk perangkat lunak, serta hasil analisis terhadap keseluruhan sistem dan elemen-elemen penyusunnya.

1. **Analisis Kebutuhan Fungsional**

Analisis kebutuhan fungsional merujuk pada data yang diperlukan agar sistem dapat beroperasi sesuai prosedur. Setelah melalui tahap analisis, kebutuhan-kebutuhan untuk pembangunan sistem akan ditetapkan. Pada tahap analisis kebutuhan fungsional, terdapat beberapa kriteria utama dalam membangun sistem, termasuk analisis kebutuhan *input*, analisis kebutuhan proses, dan analisis kebutuhan output.

Analisis kebutuhan fungsional pada sistem deteksi objek menggunakan citra digital rontgen dada untuk mendeteksi penyakit pneumonia atau normal dengan metode *Faster* R-CNN. Proses ini melibatkan reaksi sistem terhadap masukan sistem dan tugas-tugas yang akan dijalankan oleh sistem tersebut. Adapun pada tahap analisis kebutuhan fungsional sistem telah ditetapkan, mencakup kebutuhan *input*, kebutuhan proses, dan kebutuhan output. Dalam konteks penelitian ini, kebutuhan fungsionalnya dijelaskan sebagai berikut:

1. Kebutuhan *Input*:Sistem deteksi objek membutuhkan citra digital rontgen dada sebagai *input*. Citra ini adalah representasi visual dari objek yang akan dideteksi, seperti gambaran penyakit pneumonia atau kondisi normal.
2. Kebutuhan Proses: Analisis proses pada aplikasi ini bertujuan untuk mengidentifikasi permasalahan yang ada pada sistem. Sistem harus dapat melakukan deteksi objek pada citra masukan yang diberikan oleh pengguna. Proses deteksi objek diimplementasikan menggunakan bahasa pemrograman Python dengan metode *Faster* R-CNN. Penggunaan metode ini dikombinasikan dengan TensorFlow yang diimplementasi dalam *Software Google Colabolatory* akan mendukung proses ini.
3. Kebutuhan *Output*: Output yang dihasilkan oleh sistem adalah tampilan citra dengan objek yang telah terdeteksi dan diklasifikasikan baik dalam kondisi normal ataupun kondisi pneumonia dengan keterangan nama citra yang di*input* oleh pengguna.
4. **Analisis Kebutuhan Non Fungsional**

Analisis kebutuhan non-fungsional adalah tahap dimana pembuat sistem menganalisis sumber daya yang diperlukan untuk konstruksi sistem yang sedang dikembangkan. Pada tahap analisis kebutuhan non-fungsional, proses ini dapat dibagi menjadi dua langkah, yaitu:

1. Kebutuhan Perangkat Keras

Untuk menerapkan rancangan yang telah dijelaskan sebelumnya, dibutuhkan beberapa perangkat keras sebagai sarana untuk mengimplementasikan sistem yang dibangun. Kebutuhan perangkat keras ditunjukkan pada Tabel 3.2.

**Tabel 3.2 Spesifikasi Perangkat Keras**

|  |  |  |
| --- | --- | --- |
| **No** | **Nama Perangkat** | **Spesifikasi** |
| 1 | Laptop | THOSIBA Satellite C840 |
| 2 | *Processor* | Intel(R) Core(TM) i3-3110M CPU @ 2.40GHz 2.40 GHz |
| 3 | Monitor | 14 Inch |
| 4 | Memori | Ram 9 GB, DDR3 *Memory* |
| 5 | *Hardisk* | 500 GB, 240 GB SSD Klevv |

1. Kebutuhan Perangkat Lunak

Perangkat lunak atau *software* yang dibutuhkan ditunjukkan pada Tabel 3.3.

**Tabel 3.3 Spesifikasi Perangkat Lunak**

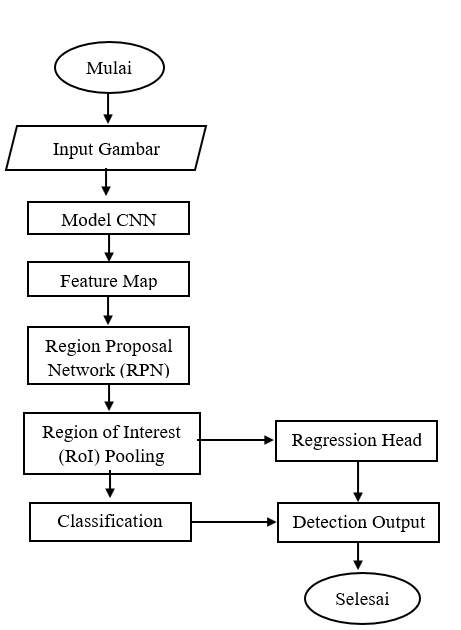
|  |  |  |  |
| --- | --- | --- | --- |
| **No** | **Nama** | **Fungsi** | **Spesifikasi** |
| 1 | Windows | Sistem operasi | Windows 11 Pro |
| 2 | Python | Bahasa Pemrograman | Python 3.9 |
| 3 | TensorFlow | *Software Library* | TensorFlow 2.5 |
| 4 | Google Chrome | *Web Browser* | Versi 120.0.6099.111 |
| 5 | Google Colabolatory | *Code Editor* | - |

1. **Perancangan Sistem**

Perancangan sistem adalah tahapan yang dilakukan dalam penelitian ini. Perancangan sistem ini mencakup serangkaian tahapan yang akan peneliti tempuh sebagai acuan guna mencapai hasil yang diinginkan dan sesuai dengan alur kerja yang telah dibuat.

1. **Perancangan Alur Sistem Secara Umum**

Berikut adalah gambaran umum dari alur kerja algoritma untuk deteksi pneumonia pada citra rontgen dada menggunakan algoritma *Faster Region Convolutional Neural Network* (*Faster* R-CNN):

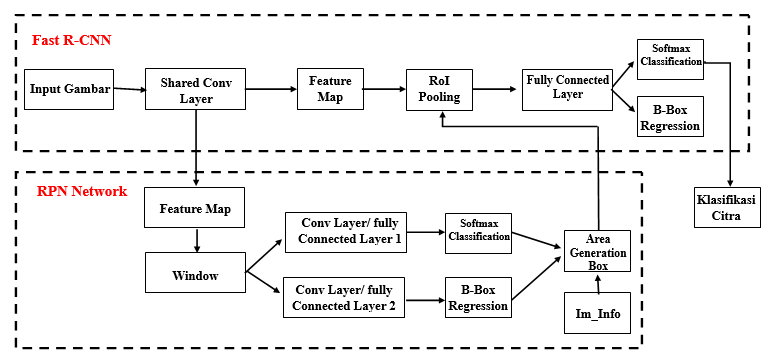


Gambar 3.7 Flowchart Sistem Secara Umum

Berikut penjelasan alur Sistem pada Gambar 3.7:

1. ***Input* Citra Rontgen:** Mulai dengan mengambil citra rontgen sebagai *input*. Citra ini berupa gambar dada yang diambil menggunakan perangkat medis seperti sinar-X.
2. ***Preprocessing* Citra:** Pra-pemrosesan pada citra dilakukan untuk normalisasi intensitas piksel atau *resizing*, peningkatan kontras dam kualitas, segmentasi citra dan operasi lainnya untuk mempersiapkan citra sebelum dimasukkan ke dalam model.
3. **Ekstraksi Fitur dengan *Convolutional Neural Network* (CNN):** Gunakan lapisan-lapisan konvolusi dalam arsitektur CNN untuk mengekstraksi fitur-fitur penting dari citra. Proses ini membantu model untuk memahami pola dan karakteristik pada citra.
4. ***Region Proposal Network* (RPN):** *Region Proposal Network* (RPN) untuk menghasilkan proposal daerah (*region proposals*) yang kemungkinan mengandung objek. RPN membantu dalam mengidentifikasi area pada citra yang memiliki potensi untuk berisi objek tertentu.
5. ***Region of Interest* (RoI) Pooling:** Terapkan *RoI pooling* pada proposal daerah untuk menghasilkan fitur yang tetap berukuran dari setiap proposal. Ini memungkinkan model untuk fokus pada bagian penting dari proposal.
6. **Classification Head**: Lapisan klasifikasi untuk mengidentifikasi apakah setiap proposal daerah mengandung objek pneumonia atau normal. Ini melibatkan klasifikasi biner antara kelas pneumonia dan normal.
7. ***Regression Head*:** Sementara itu,lapisan regresi untuk menghitung *bounding box* yang akurat untuk setiap proposal daerah yang mengandung objek pneumonia. Ini membantu dalam menentukan lokasi objek secara tepat.
8. ***Output* Deteksi:** Penggabungan hasil klasifikasi dan regresi untuk setiap proposal daerah dan hasilkan output akhir berupa kotak pembatas (*bounding boxes)* yang menandai lokasi objek pneumonia yang terdeteksi dan jika tidak maka akan terdeteksi normal.
   * + ***Post-processing*:** Proses pasca pemrosesan seperti *filtering non-maximum suppression* untuk menghilangkan kotak pembatas yang tumpang tindih dan mempertahankan kotak pembatas yang paling relevan.
     + ***Output* Deteksi Pneumonia:** Hasil akhir adalah citra rontgen yang telah ditandai dengan kotak pembatas yang menunjukkan lokasi deteksi pneumonia.
9. **Perancangan Alur Metode *Faster* R-CNN**

Adapun perancangan arsitektur sistem menggunakan metode *Faster Region Convotional Neural Network* dapat dilihat pada Gambar 3.8. Algoritma *Faster* RCNN adalah gabungan dari *Fast* CNN dan RPN *Network*. Citra *Input* akan masuk ke model CNN VGG-16 yang menghasilkan *feature map*, pada RPN *Network* lalu mencari lokasi dan skala yang tepat dari objek yang ingin dideteksi (*sliding window*) yang diteruskan ke *convolutional layer* dan dicocokkan melalui *softmax* dan *B-box regression* diteruskan ke *RoI pooling* lalu dihubungkan ke klasifikasi untuk mendeteksi target objek.



Gambar 3.8 Arsitektur Sistem Faster RCNN

1. **Perancangan Antarmuka**

*User Interface* (UI) adalah antarmuka yang memungkinkan pengguna berinteraksi dengan sebuah aplikasi atau sistem. Dalam konteks deteksi objek, UI berperan sebagai jendela interaktif yang menyajikan informasi visual kepada pengguna. Tampilan depan aplikasi menyajikan objek yang akan dideteksi dan hasil klasifikasinya. Pengguna dapat melihat hasil dari proses klasifikasi yang dibagi menjadi dua kategori, yaitu pneumonia dan normal. Hal ini memberikan pemahaman cepat kepada pengguna tentang status atau kondisi yang terdeteksi oleh sistem. Gambar 3.9 menampilkan tampilan UI secara visual, memberikan representasi grafis dari objek dan hasil klasifikasinya.



Gambar 3.9 Perancangan User Interface

1. **Skema Evaluasi Kinerja Sistem**
2. **Pengujian Metode *Faster Region Convolutional Neural Network***

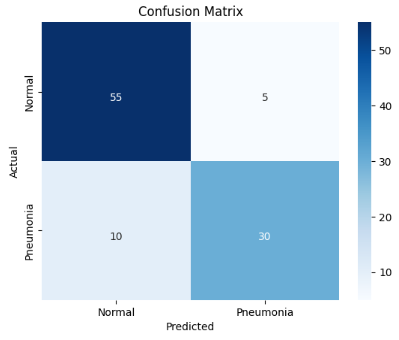
Pada tahap evaluasi model klasifikasi, *confusion matrix* menjadi instrumen kunci yang menyajikan informasi tentang jumlah prediksi yang benar dan yang salah. Dengan matriks ini, kita dapat menghitung metrik-metrik penting seperti *precision, sensitivity/recall, specificity, accuracy* dan *F-measure.* Semakin besar nilainya maka hasil deteksi menjadi semakin akurat.

**Tabel 3.4 Contoh Tabel Confusion Matrix**

|  |  |  |
| --- | --- | --- |
|  | Relevan | Tidak Relevan |
| Relevan | *30* | *5* |
| Tidak Relevan | *10* | *55* |

Berdasarkan nilai yang diperoleh pada Tabel 3.4 dapat dihitung *precision, sensitivity/recall, specificity, accuracy* dan *F-measure* sebagai berikut:

Berdasarkan hasil perhitungan diperoleh nilai precision = 87%, sensitivity/recall = 75%, specificity = 91%, accuracy = 85% dan F-measure = 80%. Dengan demikian hasil deteksi memiliki keakuratan yang baik. Dari hasil perhitungan tersebut dapat dilihat pada Gambar 3.10 Dalam menghasilkan grafik confusion matrix.



Gambar 3.10 Contoh Grafik Confusion Matrix

Adapun untuk pengujian lebih lanjut guna mencari nilai akurasi terbaik dapat dilakukan dengan beberapa skema skenario pengujian dalam yang tepat dan berbagai kemungkinan. Pada Tabel 3.5 merupakan hasil pengujian yang dilakukan pada penelitian berjudul Pneumonia and COVID-19 Classication in Chest Xrays Using Faster Region-Based Convolutional Neural Networks (Faster R-CNN) dengan hasil nilai akurasi tertinggi yang terdapat pada skenario 2 dengan panjang epoch 250 (Farhat et al., 2022).

**Tabel 3.5 Hasil Skenario Pengujian Metode Faster RCNN**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **No** | **Skenario** | **Jumlah Data** | | **Panjang *epoch*** | ***Accuracy %*** |
| **Training** | **Testing** |
| 1 | 1 | 18355 | 100 | 50 | 0.73 |
| 250 | 0.78 |
| 400 | 0.51 |
| 2 | 2 | 18555 | 140 | 50 | 0.2214 |
| 250 | 0.7642 |
| 400 | 0.3214 |
| 3 | 3 | 18355 | 100 | 50 | 0.51 |
| 250 | 0.48 |
| 400 | 0.51 |
| 4 | 4 | 18555 | 100 | 50 | 0.46 |
| 250 | 0.46 |
| 400 | 0.45 |
| 5 | 5 | 18431 | 18 | 50 | 0.1666 |
| 250 | 0.25 |
| 400 | 0.3333 |
| 6 | 6 | 18631 | 18 | 50 | 0.2777 |
| 250 | 0.1666 |
| 400 | 0.2222 |
| 7 | 7 | 1822 | 300 | 50 | 0.29333 |
| 250 | 0.31333 |
| 400 | 0.3266 |

1. **Pengujian Fungsionalitas Perangkat Lunak**

Pengujian fungsionalitas sistem ini dilakukan menggunakan *black box testing* untuk deteksi penyakit pneumonia menggunakan citra rontgen dada dengan algoritma *Faster* R-CNN melibatkan pengujian terhadap fungsionalitas eksternal tanpa mengetahui implementasi internal. Pengujian *black box* bertujuan untuk memastikan bahwa sistem berfungsi sesuai dengan spesifikasi dan memenuhi kebutuhan fungsional pengguna. Pengujian ini tidak memerlukan pengetahuan tentang sumber kode tetapi hanya berfokus pada masukan dan keluaran yang diharapkan. Tabel 3.6 adalah beberapa skenario pengujian fungsionalitas perangkat lunak menggunakan *black box testing*:

**Tabel 3.6 Pengujian Black Box**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No** | **Masukkan** | **Aksi** | **Output Yang Diharapkan** | **Kesimpulan** |
| 1 | *Input* Gambar | Menekan tombol *button* Input gambar | Sistem menampilkan gambar input | Sesuai |
| 2 | Klasifikasi | Menekan tombol *button* klasifikasi | Sistem menampilkan *output* | Sesuai |

# **DAFTAR PUSTAKA**

Aningtiyas, P.R., 2020, Pembuatan Aplikasi Deteksi Objek Menggunakan TensorFlow Object Detection API Dengan Memanfaatkan SSD MobileNet V2 Sebagai Model Pra - Terlatih, *Jurnal Ilmiah Komputasi*, 19, 3, 421–30.

Basyir, M.A, 2021, Penerapan Metode Convolutional Neural Network Dengan Arsitektur Efficiennet-B4 Untuk Klasifikasi Penyakit Pneumonia, *Skripsi*, Pekanbaru, Universitas Islam Negeri Sultan Syarif Kasim.

Charli, F. *et al.*, 2020, Implementasi Metode Faster Region Convolutional Neural Network (Faster R-CNN) Untuk Pengenalan Jenis Burung Lovebird, *Journal of Information Technology Ampera*, 1, 3, 1–12.

Dijaya, R., 2019, *Buku Ajar Citra Digital*, UMSIDA Press, Sidoarjo.

Direktorat Pencegahan dan Pengendalian Penyakit Menular Kementrian Kesehatan, 2022, *Rencana Aksi Kegiatan 2020-2024 (Revisi)*, Direktorat Pencegahan Dan Pengendalian Penyakit Menular Kementrian Kesehatan, Jakarta.

Farhat, H.J. *et al.,* 2022, Pneumonia and COVID-19 Classification in Chest X-Rays Using Faster Region-Based Convolutional Neural Networks (Faster R-CNN), *Research Square*, 2, 4, 1–11.

Fauzy, A.R., 2019, Implementasi Metode Region Convolutional Neural Network Dalam Mendiagnosa Anomali Pneumonia Pada Foto Thorax, *Skripsi*, Bandung, Universitas Telkom Indonesia.

Fitrianingsi *et al*., 2022, *Profil Rumah Sakit Umum Daerah Kota Kendari*, Rumah Sakit Umum Daerah Kota Kendari, Kendari.

Guntara, R.G., 2023a, Pemanfaatan Google Colab Untuk Aplikasi Pendeteksian Masker Wajah Menggunakan Algoritma Deep Learning YOLOv7, *Jurnal Teknologi Dan Sistem Informasi Bisnis*, 05, 01, 55–60.

Guntara, R.G., 2023b, Visualisasi Data Laporan Penjualan Toko Online Melalui Pendekatan Data Science Menggunakan Google Colab, *Jurnal Ilmiah Multidisiplin,* 02, 06, 1–10.

Hanggara, N.R., Kumalasari, N.R., & Kasih, P., 2021, Penerapan Content Based Image Retrieval Untuk Pengenalan Jenis Ikan Koi, *Seminar Nasional Inovasi Teknologi*, 05, 01, 1–6.

Hanun, N., Sarosa, M., & Andrie Asmara, R., 2023, Pemanfaatan Algoritma Faster R-CNN ResNet-101 Untuk Deteksi Potongan Tubuh Manusia, *Jurnal Elektronika Dan Otomasi Industri*, 10, 1, 94–103.

Hastuti, E., 2023, Pneumonia, Direktorat Jenderal Pelayanan Kesehatan, Kementerian Kesehatan Republik Indonesia, 21 August 2023, <https://yankes.kemkes.go.id/view_artikel/2709/pneumonia>, diakses 15 Desember 2023.

Hatim, F., 2022, World Pneumonia Day 2022, 29 December 2022, https://yankes.kemkes.go.id/view\_artikel/1997/world-pneumonia-day-2022, diakses 7 Desember 2023.

Hermawati, F.A., 2021, Sistem Deteksi Keganasan Kanker Paru-Paru Pada CT Scan Dengan Metode Mask Region-Based Convolutional Neural (Mask R-CNN). *Konferensi Nasional Ilmu Komputer (KONIK)*, 05, 01, 195-197.

Honainah, 2022, Penerapan Metode Faster Region Convolutional Neural Network (Faster R-CNN) Untuk Deteksi Otomatis Interaksi Laki-Laki Dan Perempuan, *Nusantara Journal of Computers and Its Applications,* 7, 1, 1–10.

Imantiyar, R., 2021, Pengaruh Bias Dataset Terhadap Performa Akurasi Deteksi Objek Pada Arsitektur Efficientdet-Lite, *Skripsi*, Yogyakarta, Universitas Islam Indonesia.

Jumakil, & Lymbran Tina, 2019, Perkembangan Kasus Pneumonia Pada Balita Di Provinsi Sulawesi Tenggara Tahun 2017, *Preventif Jurnal*, 3, 2, 2-4 .

Labbay, H., 2020, Aplikasi Perbaikan Kualitas Citra Digital Dengan Metode Perataan Histogram Dan Metode Fuzzy Filtering, *Skripsi*, Medan, Universitas Pembangunan Panca Budi.

Lin, J. & Irsyad, H., 2021, Klasifikasi Pneumonia Pada Citra X-Rays Paru-Paru Menggunakan GLCM Dan LVQ, *Jurnal Algoritme,* 1, 2, 84–94.

Malabay, 2016, Pemanfaatan Flowchart Untuk Kebutuhan Deskripsi Proses Bisnis, *Jurnal Ilmu Komputer,* 01, 01, 20-21.

Marpaung, F. *et al.*, 2022, *Computer Vision Dan Pengolahan Citra Digital*, Edisi 1, Pustaka Aksara, Surabaya.

Megawan, S. & Lestari, W.S. 2020, Deteksi Spoofing Wajah Menggunakan Faster R-CNN Dengan Arsitektur Resnet50 Pada Video, *Jurnal Nasional Teknik Elektro Dan Teknologi Informasi*, 09, 01, 1-4.

Muharom, B., Hidayat, H. and Putra, R.E., 2019, Penerapan CNN Dengan Filter Gabor Sebagai Feature Extractor Untuk Content-Based Image Retrieval, *Journal of Informatics and Computer Science,* 01, 01, 4-6.

Ordiyasa, I.W. and Bintang, A., 2022, DeePNeu: Robust Detection of Pneumonia Symptoms Using Faster R-CNN, *International Journal of Informatics and Computation (IJICOM)*, 4, 1, 1–12.

Prasetyo, B.D., 2020, Klasifikasi Citra X-Ray Paru-Paru Anak Pneumonia Dan Non-Pneumonia Menggunakan Metode Segmentasi Dan Deteksi Tepi, *Skripsi*, Yogyakarta, Universitas Islam Indonesia.

Pujitresnani, A., 2015, Pengolahan Citra Radiologi Sinar-X Menggunakan Metode Active Contour Untuk Identifikasi Kelainan Pada-Paru, *Thesis*, Jakarta, Universitas Indonesia.

Septian, I. and Septanto, H, 2022, Pengembangan Model Pendeteksian Gambar Alat Musik Dengan Metode Faster R-CNN Dengan Library Keras, *KALBISIANA : Jurnal Mahasiswa Institut Teknologi Dan Bisnis Kalbis*, 8, 1, 1–11.

Shibly, K.H. *et al.*, 2020, COVID Faster R-CNN: A Novel Framework to Diagnose Novel Coronavirus Disease (COVID-19) in X-Ray Images, Department of Computer Science and Engineering (CSE), *Informatics in Medicine Unlocked,* 01, 03, 1-9 .

Sibuea, F., Hardhana, B. and Widiantini, W., 2022, *Profil Kesehatan Indonesia*. Kementerian Kesehatan Republik Indonesia, Jakarta.

Steno, P. *et al.,* 2021, A Novel Enhanced Region Proposal Network and Modified Loss Function: Threat Object Detection in Secure Screening Using Deep Learning, *Journal of Supercomputing*, 77, 8, 40–69.

Tiku, S. *et al.* (2023) *Profil Kesehatan Sulawesi Tenggara Tahun 2022*. Dinas Kesehatan provinsi Sulawesi Tenggara, Kendari.

Wibi B.N, H., Mailoa, E. and Purnomo, H.D., 2021, Deteksi Buah Untuk Klasifikasi Berdasarkan Jenis Dengan Algoritma CNN YOLOv3, *JURNAL RESTI: Rekayasa Sistem Dan Teknologi Informasi,* 04, 03, 76–81.

Yao, S. *et al.*, 2021, Pneumonia Detection Using an Improved Algorithm Based on Faster R-CNN, *Computational and Mathematical Methods in Medicine*, 01, 01, 1–13.

Yopento, J. and Coastera, F., 2022, Identifikasi Pneumonia Pada Citra X-Ray Paru-Paru Menggunakan Metode Convolutional Neural Network (CNN) Berdasarkan Ekstraksi Fitur Sobel, *Jurnal Rekursif,* 10, 01, 40–47.

Yulianeu, A., and Oktamala, R., 2022, Sistem Informasi Geografis Trayek Angkutan Umum Di Kota Tasikmalaya Berbasis Web, *Jurnal Teknik Informatika*, 10, 02, 127-128.