

BAB 2

DASAR TEORI

2.1. KAJIAN PUSTAKA

Dwi Oktaria dan Maharani Sekar Ningrum pada tahun 2017 melakukan penelitian tentang “Pengaruh Merokok dan Defisiensi Alfa-1 Antitripsin terhadap Progresivitas Penyakit Paru Obstruktif Kronis (PPOK) dan Emfisema” menggunakan metode analisis berdasarkan observasi penelitian sebab akibat efek buruk dari merokok. Hasil dari penelitian ini menjelaskan bahwa kebiasaan buruk merokok yang berkepanjangan dan memiliki genetik akan menyebabkan seseorang terjangkit penyakit paru obstruktif kronik terutama emfisema. Aktivitas buruk merokok atau juga terkena polusi udara berbahaya lainnya dapat merusak organ dalam paru yang membuat tingkat elastisitas alveolar paru melemah sehingga pertukaran O₂ dan CO₂ terganggu. Faktor lain yang menyebabkan emfisema berupa genetik yang menurunkan alfa-1 antitripsin, bila defisiensi tersebut mencapai 35% yang mengakibatkan lemahnya proteksi jaringan parenkim yang menyebabkan peluruhan elastase neutrofil. Kondisi terjangkit emfisema diperparah apabila kebiasaan merokok disertai dengan defisiensi alfa-1 antritripsin karena memiliki faktor genetik tersebut [6].

Charless Hatt, Craig Galban, Wassim Labaki, Ella Kazerooni, David Lynch, serta Meilan Han pada tahun 2018 melakukan penelitian mengenai “*Convolutional neural network Based COPD and Emphysema Classification Are Predictive of Lung Cancer Diagnosis*”. Pada penelitian ini, CT scans COPD dan emfisema diklasifikasikan memakai metode *convolutional neural network*. sebanyak 7347 yang diambil dari *National Lung Cancer Screening* (NLCS). Model CNN yang digunakan yaitu CSCNN dan ESCNN yang dimodifikasi penulis, dalam klasifikasi COPD dan emfisema digeneralisasi berdasarkan umur, berat badan, pendidikan, gender, konsumsi rokok dalam satu tahun, dan tahun berhenti merokok. Setelah dilakukan klasifikasi ditemukan bahwa model CSCNN dan ESCNN merupakan prediktor bagus untuk untuk memprediksi kanker paru-paru dari penyakit awal COPD dan emfisema dengan tingkat akurasi yang bagus [2].

Selcuk Yazar pada tahun 2022 melakukan penelitian mengenai “*A Comparative Study on Classification Performance of Emphysema with Transfer Learning Methods in Deep Convolutional neural network*” yang memakai metode *convolutional neural network* dengan dua arsitektur berbeda yaitu model Xception dan model DenseNet121. Citra yang digunakan pada penelitian ini yaitu menggunakan *chest x-ray* 14 berupa data yang disediakan oleh *National Institute of Health (NIH) United State of America*. Dalam terdapat 112.120 gambar *rontgen* anterior dada pada 30.807 pasien amerika. Pengklasifikasian dilakukan pada sebanyak 5.807 *chest x-ray* diagnosis emfisema dari tiga *database* terpisah. Pada penelitian ini penulis mengembangkan aplikasi berbasis pemrograman bahasa *python* untuk menggunakannya sebagai pengklasifikasi emfisema. Setelah dilakukan percobaan didapatkan hasil dari kedua model CNN Xception dan DenseNet121 yang telah dilakukan pelatihan (*training*) sebelum dilakukan pengujian. Xception memperoleh validasi akurasi sebesar 86,44% dan skor AUC 90%. Hasil ini sedikit lebih rendah dibandingkan dengan model DenseNet121 yang memperoleh skor validasi akurasi sebesar 94% dan skor AUC sebesar 95% [7].

Jiashi Zhao, Mengmeng Li, Weili Shi, Yu Miao, Zhengang Jiang, Dan Bai Ji pada tahun 2021 melakukan penelitian tentang “*A Deep Method for Classification of Chest X-ray Images*” yang menggunakan metode *convolutional neural network* dengan arsitektur dasar DenseNet121, namun penulis melakukan modifikasi dengan menambahkan modul setelah *dense block* untuk menambah kemampuan fitur ekstraksi, dan juga menambahkan fungsi *focal loss* untuk mengatasi ketidakseimbangan masalah pada data. Model yang telah dibuat oleh penulis dinamakan *Attention Mechanism_DenseNet*. Sebanyak 112.120 *x-ray* paru bagian depan dari 14 jenis patologis paru dan terutama terdapat emfisema. Setelah dilakukan eksperimen didapatkan hasil bahwa AM_DenseNet efektif dapat mengklasifikasikan 14 jenis patologis paru dengan nilai rata-rata AUC sebesar 0,8537, dan khusus untuk klasifikasi pada emfisema memperoleh nilai AUC sebesar 0,9479 [4].

Erdi Calli, Keelin Murphy, Ernst T. Scholten, Steven Schalekamp, Bram van Ginneken pada tahun 2022 melakukan penelitian tentang “*Explainable emphysema detection on chest radiographs with deep learning*” menggunakan

sistem *deep learning* yaitu ResNet18 untuk secara otomatis mendeteksi emfisema dari citra *x-ray* dada *frontal* dan *lateral*. Rontgen dada *frontal* dan *lateral* 3000 studi dikumpulkan secara retrospektif. Dua ahli radiologi dijelaskan dengan 4 tanda-tanda radiologis emfisema paru diidentifikasi dalam literatur. Dari dua dari tanda-tanda ini dianggap emfisema positif. Dengan menggunakan sistem *deep learning* terpisah untuk gambar depan dan samping, kami memperkirakan keberadaan masing-masing dari keempatnya tanda-tanda visual dan gunakan untuk menentukan kepositifan emfisema. Pada 422 kasus uji, emfisema positif diprediksi dengan AUC 0,924 dan 0,946 menggunakan standar acuan masing-masing ahli radiologi secara terpisah. Dengan menentukan sensitivitas model yang sama dengan ahli radiologi kedua, model ResNet18 yang digunakan penulis memiliki spesifisitas yang sebanding ($p = 0,880$ dan $p = 0,143$ untuk setiap ahli radiologi). Metode ResNet18 sebanding dengan model *black-box* dengan AUCs 0,915 ($p = 0,407$) dan 0,935 ($p = 0,291$). Dari 370 kasus di mana dua ahli radiologi setuju (53 positif), model ResNet18 memperoleh AUC 0,981, sekali lagi dibandingkan dengan model *black-box* AUC sebesar 0,972 ($p = 0,289$). Metode yang diusulkan dapat memprediksi kepositifan emfisema pada citra *x-ray* dada [8]. Berikut tabel kajian penelitian sebelumnya ditunjukkan pada tabel 2.1.

Tabel 2.1 Kajian Penelitian Sebelumnya

No	Nama dan Judul	Jumlah	Metode	Hasil
1	<i>Charles Hatt, 2018 : Convolutional neural network Based COPD and Emphysema Classification Are Predictive of Lung Cancer Diagnosis</i>	CT scan 7347	CNN	cox regresi on 71,4%
2	<i>Selcuk Yazar, 2022 : A Comparative Study on Classification Performance of Emphysema with Transfer Learning Methods in Deep Convolutional neural network</i>	X-ray 5807	Densenet12 1 & Xception	akurasi xceptio n 86,44 % & densen et121 94%
3	<i>Jiashi Zhao, 2021 : A Deep Method for Classification of Chest X-ray Images</i>	X-ray 112120	Attention Mechanism _Densenet	AUC 94,79 %
4	<i>Erdi Calli, 2022 : Explainable emphysema detection on chest radiographs with deep learning</i>	X-ray 3000 frontal, 3000 lateral	ResNet18	AUC 98,1%

No	Nama dan Judul	Jumlah	Metode	Hasil
5	<i>Liyong Peng, 2018 : Classification Of Pulmonary Emphysema In Ct Images Based On Multi-Scale Deep Convolutional neural network</i>	HRCT 102000 0	multi-scale early fusion CNN	akurasi 92.68%
6	<i>Sumedha Singla, 2021 : Improving clinical disease subtyping and future events prediction through a chest CT-based deep learning approach</i>	HRCT 10300	CNN	akurasi 74.68%
7	<i>Zong Li, 2022 : Early detection of COPD based on graph convolutional network and small and weakly labeled data</i>	CT Scan 300	GCN (graph convolution network)	akurasi 77%, AUC 81%
8	<i>Stephen M. Humphries, 2020 : Deep Learning Enables Automatic Classification of Emphysema Pattern at CT</i>	CT Scan 7143	Kombinasi CNN LSTM	cohort (P < 001)

2.2. DASAR TEORI

2.2.1. Emfisema

Salah satu penyakit yang menyerang paru yaitu emfisema. Emfisema merupakan penyakit yang menyerang organ paru-paru dari kebiasaan buruk merokok maupun tidak sengaja menghirup udara yang mengandung polusi udara berbahaya lainnya. Kondisi patologis emfisema berupa udara yang terperangkap dalam organ paru tidak dapat keluar atau melakukan sirkulasi udara karena alveoli pada jaringan parenkim paru rusak, elastisitasnya berkurang. Karakteristik paru-paru yang terkena emfisema yaitu paru-paru membesar namun mengalami sesak nafas, dan bentuk diafragma abnormal. Emfisema terparah yaitu emfisema subkutis yang menimbulkan ketidaknyamanan selain sesak napas yaitu terdapatnya sekumpulan udara di subkutis pada dinding dada yang menjalar ke jaringan lunak wajah sehingga terjadi pembengkakan area leher, dada atas, bahu, hingga wajah [9]. Berikut merupakan gambaran dari citra *x-ray* paru-paru normal dan emfisema yang ditunjukkan pada Gambar 2.1.



Gambar 2.1 a. Paru-paru Normal. b Paru-paru Emfisema

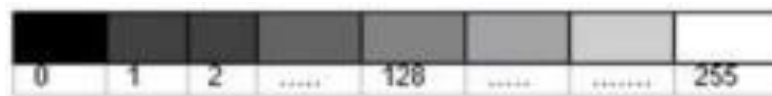
2.2.2. Deep Learning

Deep learning merupakan bagian dari *machine learning* yang dirancang secara kontinu untuk menganalisis dan memproses data atau mengukur nilai data pada setiap lapisan dari *layer* sebelumnya yang berjalan mundur dengan struktur logika pengambilan keputusan yang mirip dengan manusia. *Deep learning* dapat menggunakan banyak *layer* sampai dengan ratusan *layer* [10]. Salah satu kemampuan dari *deep learning* yaitu mengubah fitur buatan tangan dengan algoritma yang efisien untuk pembelajaran *supervised Learning*, *unsupervised learning*, dan *semi-supervised learning*. *Supervised learning* merupakan algoritma yang akan melakukan prediksi dan koreksi terus menerus dari hingga mencapai nilai akurasi yang diinginkan, contohnya seperti regresi linear, *random forest* dan *support vector machine*. Algoritma *unsupervised learning* merupakan kebalikan dari *supervised learning* yang tidak menyajikan *output*, karena hanya berfokus pada memodelkan struktur yang dapat mempelajari data saja, contohnya seperti *k-means clustering* dan algoritma apriori. Algoritma *semi-supervised learning* merupakan kombinasi dari algoritma *supervised* dan *unsupervised*. Algoritma *unsupervised* untuk menemukan struktur model terbaik untuk data *input* dan algoritma *supervised* untuk membuat prediksi data dengan akurasi yang diinginkan. *Deep learning* telah banyak diterapkan pada beberapa bidang seperti deteksi objek, pengenalan objek, mengubah teks ke suara, robotik, dan klasifikasi teks, gambar, maupun video [11].

2.2.3. Citra x-ray

Sinar x-ray adalah salah satu bentuk *electromagnetic wave* yang berupa cahaya serta gelombang bunyi. Panjang gelombang dari *x-ray* pendek memiliki

frekuensi tinggi sehingga memiliki energi besar untuk dapat menembus dan melihat organ tubuh manusia. Mesin *x-ray* memancarkan radiasi kecil dan merekam citra pada plat khusus film. Organ keras seperti tulang akan menerima radiasi besar dibandingkan organ lunak seperti otot, lemak dan lainnya sehingga tulang pada citra akan berwarna putih, udara akan berwarna hitam, serta warna kelabu menunjukkan organ lunak. *X-ray* bermanfaat di dunia medis untuk melihat dan mendiagnosis penyakit organ tubuh manusia seperti penyakit pada paru-paru. Citra *x-ray* merupakan citra yang tergolong pada *greyscale* yang memiliki nilai intensitas tingkat keabuan. Pada Gambar 2.2 citra *greyscale* berupa citra 8-bit yang memiliki tingkat intensitas 0 sampai 255. Warna hitam untuk tingkat paling rendah dan putih tingkat paling tinggi. Setiap kanal hanya memiliki satu nilai piksel [12]. Berikut merupakan tingkat intensitas *greyscale* yang ditunjukkan pada Gambar 2.2.



Gambar 2.2 Tingkat Intensitas *Greyscale*

2.2.4. *Image Processing*

Pengolahan citra atau dengan kata lain *image processing* merupakan suatu kegiatan yang dilakukan komputer untuk melakukan beberapa operasi atau manipulasi dari suatu objek yang berupa gambar. Selain itu untuk mendapatkan suatu informasi dari gambar yang telah diekstraksi atau pula menyempurnakan gambar yang kurang baik. Informasi dari gambar berupa gambar visual dua dimensi matriks gambar. *Input* dari *image processing* dapat berupa gambar dan keluarannya dapat berupa gambar ataupun karakteristik gambarnya. Cara kerja sistem *image processing* berbeda dengan visual manusia karena terbatas *bandwidth* dan *noise*. Sehingga tujuan sebenarnya dari *image processing* bukan hanya untuk memanipulasi gambar menjadi lebih estetik tetapi menyamakan persepsi antara sistem visual komputer dan manusia [13].

2.2.5. Augmentasi Data

Augmentasi data yaitu proses memanipulasi, memodifikasi citra tanpa kehilangan informasi penting atau dengan kata lain citra atau gambar tersebut sebenarnya citra yang sama sebelum augmentasi. Tahap augmentasi dilakukan untuk memperbanyak dan memvariasikan citra tanpa kehilangan informasi ataupun esensi citra bahkan memperbanyak informasi. Metode dalam augmentasi data yang sering digunakan yaitu *rescale*, *random brightness*, *horizontal flip*, *vertical flip*, dan *rotation* [14].

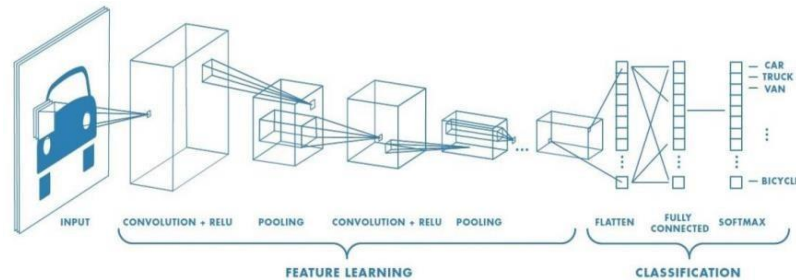
2.2.6. Bahasa Pemrograman Python

Python adalah bahasa pemrograman tingkat tinggi yang ditafsirkan, interaktif dan berorientasi objek, serta dapat digunakan di hampir semua *platform*. Bahasa pemrograman *Python* termasuk mudah untuk dipelajari karena sederhana, memiliki sintaks yang jelas, dan punya pustaka yang luas. Sintaks *python* bersifat *case sensitive* yang berarti membedakan antara huruf kecil dan besar. Identasi pada *python* sangat penting karena bila tidak tepat mengelompokkan blok kode program. Penulisan variabel pada *python* karakter pertama harus huruf atau garis bawah dan selanjutnya berupa huruf, angka atau garis bawah. Penulisan sintaks *statment* sederhana karena hanya dipisahkan titik koma bila terdapat lebih banyak *statment* dalam satu baris dan tidak diakhiri dengan titik koma. Untuk penulisan *string* dipisahkan oleh tanda petik satu, ganda, maupun *triple*. Penulisan komentar diawali dengan tanda pagar. *Python* diciptakan oleh Guido van Rossum di Belanda tahun 1990, bahasa pemrograman *python* sudah banyak digunakan oleh perusahaan besar maupun *developer* untuk mengembangkan berbagai aplikasi *desktop*, *web*, dan lain sebagainya. [15].

2.2.7. Convolutional Neural Network

Convolutional neural network merupakan sebuah metode algoritma *machine learning* yang dikembangkan dari MPL bertujuan untuk mengolah data dua dimensi, serta dimanfaatkan pula untuk deteksi citra dibidang kesehatan (citra jantung, paru-paru, toraks), agrikultur, dan lainnya [16]. CNN termasuk pula kedalam jenis *deep learning* karena memiliki banyak tingkat lapisan jaringan dan

diimplementasikan untuk data gambar. Arsitektur CNN dapat dilatih melalui beberapa tahap dari mulai masukan sampai keluaran citra atau gambar [17]. Lapisan-lapisan dari arsitektur CNN dapat digambarkan pada gambar 2.3 berikut.

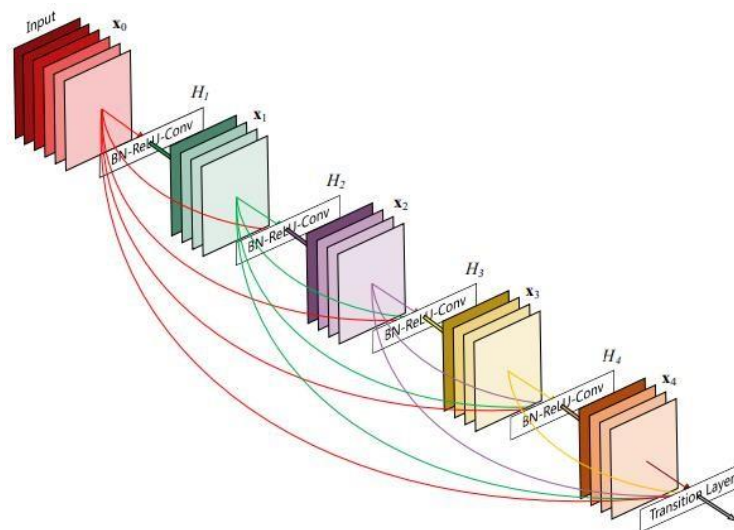


Gambar 2.3 Arsitektur Concolutional Neural Network

Pada gambar 2.3 dapat dilihat bahwa CNN dibagi menjadi dua bagian yaitu *feature learning* dan *clasification* [18]. Berdasarkan kedua bagian tersebut dibagi menjadi beberapa lapisan pada setiap bagian. Pada bagian *feature learning* terdapat lapisan *input* atau masukan untuk citra yang memiliki ukuran tertentu, selanjutnya lapisan *convolution + relu* berfungsi untuk melakukan proses konvolusi dan dalam lapisan ini terdapat kernel 3x3 [19]. Lapisan selanjutnya yaitu *pooling* untuk pengecilan ukuran citra yang memudahkan proses konvolusi selanjutnya [20]. Bagian *classification* terdiri dari lapisan *flatten* yang berfungsi untuk mengubah citra dua dimensi menjadi satu dimensi [21], lapisan kedua yaitu *fully connected layer* bertujuan untuk mentransformasikan dimensi citra agar data citra dapat diklasifikasikan [22]. Lapisan terakhir yaitu *dropout layer* yang berfungsi untuk menghindari terjadinya *overfitting*. *Overfitting* yaitu kondisi dimana data citra memiliki persentasi baik, namun terjadi ketidaksesuaian dengan prediksi. *Overfitting* menghilangkan sementara *hidden layer* dalam jaringan . Dalam model *deep learning* terdapat beberapa metode atau teknik untuk melatih *dataset* yaitu *transfer learning* dan *non-transfer learning*. Metode *transfer learning* yaitu metode yang menggunakan model yang telah dilatih terlebih dahulu (*ImageNet*) terhadap suatu *dataset* berukuran besar yang selanjutnya akan diadaptasi dengan *dataset* yang baru beserta penyesuaian parameternya sedangkan metode *non-transfer learning* merupakan model yang dibuat tanpa digunakan sebelumnya untuk klasifikasi citra.

2.2.8. DenseNet121

Convolutional neural network (CNN) memungkinkan data dalam bentuk banyak *array* diproses seperti 1D (bahasa), 2D (gambar dan suara), 3D (gambar bervolume dan video) yang memiliki piksel warna *red green blue* (RGB) maupun *greyscale*. *Densely Convolutional Network* (DenseNet) menghadirkan konsep yang solid yang menghubungkan semua lapisan/blok secara langsung dengan umpan maju. Untuk setiap level, semua *feature map* level sebelumnya digunakan sebagai *input* sedangkan *feature map* itu sendiri digunakan sebagai *input* untuk semua level berikutnya. Model *dense connection* mempertahankan semua lapisan sebelumnya dengan baik dan tidak perlu mempelajari fitur yang tidak diperlukan [23].



Gambar 2.4 Arsitektur Model DenseNet121

Gambar 2.4 berupa DenseNet-121 terdiri dari empat *dense block*, masing-masing terdiri dari jumlah tertentu. Lapisan konvolusional berisi *batch normalization* (BN), *rectified linear unit* (ReLU) *activation* dan *convolution* filter 3x3 untuk ekstraksi fitur. Setelah tiga *dense block* pertama, *transition layer* ditambahkan terdiri dari BN, 1x1 *convolutional layer* dan 2x2 *average pooling layer*. *Transition layer* digunakan untuk menghubungkan dua *dense block*, mengubah dimensi saluran dan menghitung sampel. Akhirnya model *classification layer* dan hasilnya diperoleh dengan menghubungkan *global average pooling* yang digabungkan *fully connected layer*, dan *softmax activation*. *Pooling layer* terdiri dari sebuah filter dengan ukuran dan *stride* tertentu yang bergeser pada seluruh area

feature map, dan *average pooling* berupa nilai yang diambil rata-rata pada setiap *layer* [20]. Jumlah total lapisan konvolusional dan terhubung penuh dari DenseNet-121 adalah 121 lapisan. Terlihat pada gambar 2.5, Densenet121 terdiri dari 1 lapisan awal (*convolution*, *pooling*), tiga *transition layer*, 1 *classification layer*, empat *dense block* dengan dua kali konvolusi, 1x1 dan 3x3. Sehingga $5 + (6 + 12 + 24 + 16) * 2 = 121$. Sebagai perbandingan model CNN lain memiliki parameter lebih sedikit daripada DenseNet karena ciri khusus fitur lainnya tidak perlu dipelajari. Mengatur *dense connection* memungkinkan model menggunakan kembali fitur untuk mendapatkan lebih banyak informasi aliran, dan operasi regulasi dapat mengatasi masalah kehilangan gradien [23]. Berikut merupakan gambaran lapisan yang ditunjukkan pada Gambar 2.5.

Layer	Output size	DenseNet-121	DenseNet-169	DenseNet-201	DenseNet-264
Convolutional	112x112	7x7 conv, stride 2			
Pooling	56x56	3x3 max pool, stride 2			
Dense Block (1)	56x56	$\begin{bmatrix} 1x1 \text{ conv} \\ 3x3 \text{ conv} \end{bmatrix} \times 6$	$\begin{bmatrix} 1x1 \text{ conv} \\ 3x3 \text{ conv} \end{bmatrix} \times 6$	$\begin{bmatrix} 1x1 \text{ conv} \\ 3x3 \text{ conv} \end{bmatrix} \times 6$	$\begin{bmatrix} 1x1 \text{ conv} \\ 3x3 \text{ conv} \end{bmatrix} \times 6$
Transition Layer (1)	56x56	1 x 1 conv			
	28x28	2 x 2 average pool, stride 2			
Dense Block (2)	28x28	$\begin{bmatrix} 1x1 \text{ conv} \\ 3x3 \text{ conv} \end{bmatrix} \times 12$	$\begin{bmatrix} 1x1 \text{ conv} \\ 3x3 \text{ conv} \end{bmatrix} \times 12$	$\begin{bmatrix} 1x1 \text{ conv} \\ 3x3 \text{ conv} \end{bmatrix} \times 12$	$\begin{bmatrix} 1x1 \text{ conv} \\ 3x3 \text{ conv} \end{bmatrix} \times 12$
Transition Layer (2)	28x28	1 x 1 conv			
	14x14	2 x 2 average pool, stride 2			
Dense Block (3)	14x14	$\begin{bmatrix} 1x1 \text{ conv} \\ 3x3 \text{ conv} \end{bmatrix} \times 24$	$\begin{bmatrix} 1x1 \text{ conv} \\ 3x3 \text{ conv} \end{bmatrix} \times 32$	$\begin{bmatrix} 1x1 \text{ conv} \\ 3x3 \text{ conv} \end{bmatrix} \times 48$	$\begin{bmatrix} 1x1 \text{ conv} \\ 3x3 \text{ conv} \end{bmatrix} \times 64$
Transition Layer (3)	14x14	1 x 1 conv			
	7x7	2 x 2 average pool, stride 2			
Dense Block (4)	7x7	$\begin{bmatrix} 1x1 \text{ conv} \\ 3x3 \text{ conv} \end{bmatrix} \times 16$	$\begin{bmatrix} 1x1 \text{ conv} \\ 3x3 \text{ conv} \end{bmatrix} \times 32$	$\begin{bmatrix} 1x1 \text{ conv} \\ 3x3 \text{ conv} \end{bmatrix} \times 32$	$\begin{bmatrix} 1x1 \text{ conv} \\ 3x3 \text{ conv} \end{bmatrix} \times 48$
Classification Layer	1x1	7 x 7 global average pool			
		1000D fully-connected, softmax			

Gambar 2.5 Lapisan Arsitektur Densenet121

1. Proses *preprocessing*

Proses ini melibatkan perubahan ukuran menjadi lebih kecil (*resize*) dan seragam secara horizontal dan vertikal, sehingga berjalan lebih cepat dan dengan akurasi yang lebih tinggi selama proses pelatihan dan pengujian. Selama fase *preprocessing*, citra diubah ukurannya menjadi 112 x 112. Data citra harus dinormalisasi menjadi nilai piksel antara 0 sampai 1 dan ditunjukkan pada persamaan 2.1.

$$\text{Rescale} = \text{Image}/255 \quad (2.1)$$

Keterangan :

Rescale = hasil *preprocessing*

Image = nilai *matrix* gambar

2. Padding

Zero padding adalah operasi menambahkan nilai nol ke dalam kolom dan baris matriks untuk membuat matriks gambar menjadi lebih besar dan lebih halus, gambar 2.6 merupakan ilustrasi *zero padding*.



Gambar 2.6 Ilustrasi Zero Padding

3. Convolution layer

Convolutional layer merupakan bagian penting dalam CNN yang tersusun oleh beberapa filter untuk operasi kovolusi dan berfungsi untuk memepelajari fitur dari data citra yang masuk. *Convolution layer* tersusun oleh *neuron* sebagai *receptive field* menjadi *feature maps*. Pada *layer* ini terdapat kernel yang umumnya berukuran 1x1, dan 3x3. Citra pada lapisan ini akan dikonvolusi sesuai ukuran kernel. *Neuron* akan saling terhubung dari *convolution layer* satu dengan yang lainnya. Berikut dibawah berupa persamaan konvolusi.

$$h(a, b) = f(a, b) \times g(a, b) \quad (2.2)$$

Dengan :

$h(a,b)$ = hasil konvolusi gambar

$f(a,b)$ = nilai matriks gambar

$g(a,b)$ = kernel konvolusi

4. Pooling

Pooling berperan mengurangi resolusi peta fitur sehingga mencapai invarian spasial. *Max pooling* sering digunakan pada CNN, cara kerjanya yaitu membagi keluaran lapisan konvolusi menjadi *grid-grid* kecil dan diambil nilai tertingginya pada tiap *grid* untuk menyusun *matrix* gambar yang telah dikurangi. *Global average pooling* yaitu membagi seluruh hasil keluaran tiap *layer* menjadi beberapa *grid* untuk diambil nilai rerata pada tiap *grid* untuk menyusun matriks citra.

5. Batch Normalization

Batch normalization merupakan operasi dengan menyamakan distribusi tiap *input layer* yang berubah parameternya selama proses *training*. *Batch normalization* berguna untuk meningkatkan *learning rates* dan mempercepat proses *training*. Berikut persamaan *batch normalization*:

Mean

$$\mu_B = \frac{1}{m} \sum_{i=1}^m x_i \quad (2.3)$$

Variance

$$\sigma^2 = \frac{1}{m} \sum_{i=1}^m (x_i - \mu_B)^2 \quad (2.4)$$

Normalisasi

$$\hat{x} = \frac{x_i - \mu_B}{\sqrt{\sigma^2 + \epsilon}} \quad (2.5)$$

Scale dan shift

$$y_i = \hat{x} + Q \quad (2.6)$$

Keterangan :

μ_B = rerata nilai *batch*

σ_B = standar deviasi *batch* kecil

m = total *dataset*

\hat{x} = *zero-centered* dan *input normalization*

ϵ = variabel komplemen (8-10)

y_i = skala keluaran dan pergeseran dari perhitungan normalisasi *batch*

$\gamma = scale = 1$

$\beta = shift = 0$

6. Aktivasi *ReLU*

ReLU (Rectified Linear Units) digunakan sebagai fungsi aktivasi pada jaringan syaraf tiruan untuk mengubah nilai x menjadi 0 saat nilai x negatif dan sebaliknya nilai x tetap jika nilainya lebih besar atau sama dengan 0. Ditulis persamaan 2.7 seperti dibawah ini.

$$f(x_i) = \max(0, x_i) \quad (2.7)$$

$f(x_i)$ = aktivasi *ReLU*

x_i = nilai *matrix* gambar

7. *Softmax Activation*

Aktivasi *softmax* memiliki fungsi menghitung distribusi probabilitas pada vektor bilangan riil, sehingga menghasilkan nilai keluaran antara 0 sampai dengan 1. *Softmax* ini diterapkan pada lapisan *layer* terakhir pada model DenseNet121.

$$f(x_i) = \frac{\exp(x_i)}{\sum_j \exp^{x_j}} \quad (2.8)$$

$f(x_i)$ = aktivasi *softmax*

(x_i) = kelas ke- i (*2 class*)

j = nilai dari vektor

2.2.9. *Cross Entrophy Loss Function*

Loss function memiliki fungsi untuk mengetahui *error* dari suatu model yang diharapkan bernilai rendah. *Cross entrophy* ini menghitung probabilitas dua distribusi variabel acak. Salah satu *cross entrophy* yaitu *binary cross-entrophy* (BCE) dan *categorical cross-entrophy* untuk multi kelas maupun biner. BCE *loss* digunakan untuk klasifikasi dua kelas (biner) [13][24]. Persamaan matematika dari *cross-entrophy* sebagai berikut.

$$loss(x, y) = - \sum x \cdot \log (y) \quad (2.9)$$

x = nilai aktual
y = nilai prediksi

2.2.10. Optimizer

Optimizer merupakan suatu metode untuk menentukan bobot paling optimal atau terbaik dari proses iterasi. Bobot optimal dari iterasi didapatkan dari meminimalisasi kesalahan atau *loss* dan memaksimalkan akurasi sehingga didapat prediksi yang memiliki akurasi tinggi [25]. Berikut beberapa *optimizer* yang dipakai pada penelitian ini:

1. Adam (*adaptive moment estimation*) adalah penggabungan berasal RMSprop dan *stochastic gradien descent* (SGD). Metode optimasi ini dirancang untuk *deep neural network training* yang memanfaatkan *learning rate* adaptif. *Optimizer* Adam menggunakan estimasi gradien ketika pertama (*beta_1*) dan kedua (*beta_2*) agar mengadaptasi *learning rate* ke setiap bobot *neural network* [26].
2. SGD (*Stochastic Gradien Descent*) merupakan metode yang digunakan agar bobot setiap parameter diperbaiki. Parameter yang diperbaiki atau ditingkatkan yaitu nilai dari perhitungan kesalahan atau *loss* setiap *training* [27].

2.2.11. Learning rate

Learning rate adalah parameter *gradient descent* yang digunakan dengan tujuan memperbaharui bobot parameter yang akan dilatih setiap kali iterasi atau pengulangan selesai. Kisaran nilai konstanta (*alfa*) berada antara nol (0) dan maksimal satu (1). Tingkat pembelajaran mempengaruhi kinerja jaringan dalam waktu yang dibutuhkan untuk mencapai tujuan yang optimal. Ada berbagai tingkat pembelajaran selama proses pelatihan. Ini menyebabkan lebih sedikit pengulangan atau pengulangan. Ini menyebabkan pola putus karena kurangnya pemahaman [28].

2.2.12. Epoch

Epoch merupakan banyaknya jumlah iterasi atau perulangan yang diawali dari pertama hingga terakhir. Satu kali *epoch* melatih seluruh *training* untuk

melatih *neural network*. Bila terlalu banyak biasanya akan dibagi dalam beberapa satuan atau *batch* [29].

2.2.13. Klasifikasi dan Validasi

Klasifikasi citra melibatkan pengelompokan citra berdasarkan tingkat atau kelas yang sama dengan mempertimbangkan bentuk, pola, dan warna yang serupa sebagai sesuatu yang spesifik. Dalam konteks penyakit paru-paru, klasifikasi citra digunakan untuk mengelompokkan atau membedakan jenis penyakit melalui citra *x-ray* paru-paru. Untuk mendapatkan nilai dari klasifikasi citra paru-paru tersebut, dilakukan perhitungan piksel pada setiap kanal gambar menggunakan metode tertentu untuk melakukan pengklasifikasian citra..

Validasi merupakan salah satu tindak pembuktian dengan cara mengetahui nilai dari proses perhitungan akurasi, sensitivitas, dan spesifitas. Pada pembuktian tersebut menggunakan alat pengukuran yaitu *performance matrix* dengan memanfaatkan hasil dari *confusion matrix* yang di dalamnya terdapat beberapa parameter seperti TP (*true positive*), TN (*true negative*), FP (*false positive*), FN (*false negative*).

a. Accuracy

Akurasi berupa persentase tingkat kedekatan dari nilai sebenarnya dari hasil identifikasi pengujian.

b. Precision

Presisi berupa evaluasi sistem dalam menemukan peringkat yang paling relevan, dan didefinisikan sebagai presentase dokumen seberapa baik hasil pengukuran dari mengklasifikasikan citra positif terdampak penyakit.

c. Recall

Recall mengevaluasi kemampuan sistem untuk menemukan semua item yang relevan dari koleksi dokumen dan didefinisikan sebagai presentase rasio prediksi positif dari citra aktual positif terdampak penyakit [30].

2.2.14. Confusion matrix

Confusion matrix merupakan sebuah alat bantu pengukuran untuk mengukur tingkat kebenaran dari suatu proses klasifikasi apapun khususnya citra.

Confusion matrix berupa matriks atau tabel yang memuat hasil klasifikasi data yang tepat maupun hasil klasifikasi yang tidak tepat. Berikut merupakan tabel *confusion matrix* yang ditunjukkan pada tabel 2.2.

Tabel 2.2 Confusion Matrix

<i>Actual Class</i>	<i>Positive</i>	<i>True Positive (TP)</i>	<i>False Positive (FP)</i>
	<i>Negative</i>	<i>False Negative (FN)</i>	<i>True Negative (TN)</i>
		<i>Positive</i>	<i>Negative</i>
		<i>Predicted Class</i>	

Penjelasan singkat matriks dari tabel 2.2 diatas meliputi :

1. *True Positive (TP)*

True positive berupa jumlah data *actual class positive* dan pada *predicted class positive* pula. Pada matriks TP dinyatakan bahwa pada saat aktual menunjukkan benar dan setelah dilakukan pengetesan menunjukkan benar juga.

2. *True Negative (TN)*

True negative berupa jumlah data *actual class negative* dan pada *predicted class negative* pula. Pada matriks TN dinyatakan bahwa pada saat aktual menunjukkan tidak dan setelah dilakukan pengetesan menunjukkan tidak juga.

3. *False Positive (FP)*

False positive berupa jumlah data *actual class negative* dan pada *predicted class positive*. Pada matriks FP dinyatakan bahwa pada saat aktual menunjukkan tidak dan setelah dilakukan pengetesan menunjukkan benar, atau dengan kata lain dibenarkan karena suatu hal.

4. *False Negative (FN)*

False Negative berupa jumlah data *actual class positive* dan pada *predicted class negative*. Pada matriks FN dinyatakan bahwa pada saat aktual menunjukkan benar dan setelah dilakukan pengetesan menunjukkan salah, atau dengan kata lain disalahkan karena suatu hal [31].