

**KLASIFIKASI KANKER PARU-PARU
MENGUNAKAN *DEEP LEARNING* DENGAN ARSITEKTUR DENSENET
PADA *DATASET IQ-QTH/NCCD***

(Skripsi)

Oleh

**NINDI RESTU PEMBAYUN
2117051021**



**FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM
UNIVERSITAS LAMPUNG
BANDAR LAMPUNG
2025**

**KLASIFIKASI KANKER PARU-PARU
MENGUNAKAN *DEEP LEARNING* DENGAN ARSITEKTUR DENSENET
PADA *DATASET IQ-QTH/NCCD***

Oleh

NINDI RESTU PEMBAYUN

Skripsi

**Sebagai Salah Satu Syarat untuk Mencapai Gelar
SARJANA KOMPUTER**

Pada

**Jurusan Ilmu Komputer
Fakultas Matematika dan Ilmu Pengetahuan Alam Universitas Lampung**



**FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM
UNIVERSITAS LAMPUNG
BANDAR LAMPUNG
2025**

ABSTRAK

KLASIFIKASI KANKER PARU-PARU MENGUNAKAN *DEEP LEARNING* DENGAN ARSITEKTUR DENSENET PADA *DATASET IQ-QTH/NCCD*

Oleh

NINDI RESTU PEMBAYUN

Kanker paru-paru merupakan salah satu jenis kanker dengan tingkat kematian tinggi dan sering terdeteksi pada stadium lanjut. Penelitian ini bertujuan untuk mengklasifikasikan citra *CT scan* paru-paru dari *dataset IQ-QTH/NCCD* ke dalam tiga kelas yaitu normal, *benign*, dan *malignant* menggunakan arsitektur *Deep Learning* DenseNet (121, 169, dan 201). Proses klasifikasi citra dilakukan dengan tahapan *preprocessing* seperti normalisasi piksel, perubahan ukuran citra, *one-hot encoding*, serta augmentasi pada data *training* untuk meningkatkan kemampuan generalisasi model. Model dilatih menggunakan pendekatan *transfer learning* dengan *optimizer* SGD, fungsi *loss* *Categorical Crossentropy*, dan *Early Stopping* untuk mencegah *overfitting*. Hasil evaluasi menunjukkan bahwa ketiga varian DenseNet mampu mencapai akurasi di atas 90%, dengan DenseNet-201 menghasilkan performa terbaik berdasarkan nilai *macro F1-score* sebesar 0,81 dan AUC hingga 1,00. Meskipun demikian, kesalahan klasifikasi masih sering terjadi pada kelas *benign* akibat kemiripan visual dengan kelas normal.

Kata Kunci : Kanker Paru-Paru, *Deep Learning*, DenseNet, Klasifikasi Citra, *IQ-QTH/NCCD*.

ABSTRACT

LUNG CANCER CLASSIFICATION USING DEEP LEARNING WITH DENSENET ARCHITECTURE ON THE IQ-QTH/NCCD DATASET

By

NINDI RESTU PEMBAYUN

Lung cancer is one of the deadliest types of cancer and is often diagnosed at an advanced stage. This study aims to classify lung CT scan images from the IQ-QTH/NCCD dataset into three categories normal, benign, and malignant using the DenseNet deep learning architecture (121, 169, and 201). The image classification process involves several preprocessing steps, including pixel normalization, image resizing, one-hot encoding, and data augmentation on the training set to improve model generalization. The models were trained using a transfer learning approach with the SGD optimizer, Categorical Crossentropy loss function, and Early Stopping to prevent overfitting. Evaluation results show that all DenseNet variants achieved over 90% accuracy, with DenseNet-201 delivering the best performance based on a macro F1-score of 0.81 and AUC up to 1.00. However, misclassification still frequently occurred in the benign class due to visual similarity with the normal class.

Keyword: Lung Cancer, Deep Learning, DenseNet, Image Classification, IQ-QTH/NCCD.

**Judul Skripsi : KLASIFIKASI KANKER PARU-PARU
MENGUNAKAN DEEP LEARNING DENGAN
ARSITEKTUR DENSENET PADA DATASET
IQ-QTH/NCCD**

Nama Mahasiswa : Nindi Restu Pembayun

Nomor Pokok Mahasiswa : 2117051021

Program Studi : S1 Ilmu Komputer

Fakultas : Matematika dan Ilmu Pengetahuan Alam



1. Komisi Pembimbing

Dr. rer. nat. Akmal Junaidi, M.Sc.

NIP. 197101291997021001

Dewi Asiah Shofiana, S.Komp., M.Kom.

NIP. 199509292020122030

2. Ketua Jurusan Ilmu Komputer

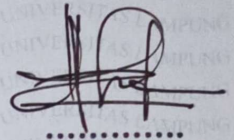
Dwi Sakethi, S.Si., M. Kom.

NIP. 196806111998021001

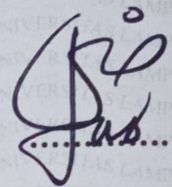
MENGESAHKAN

1. Tim Penguji

Ketua : **Dr. rer. nat. Akmal Junaidi, M.Sc.**

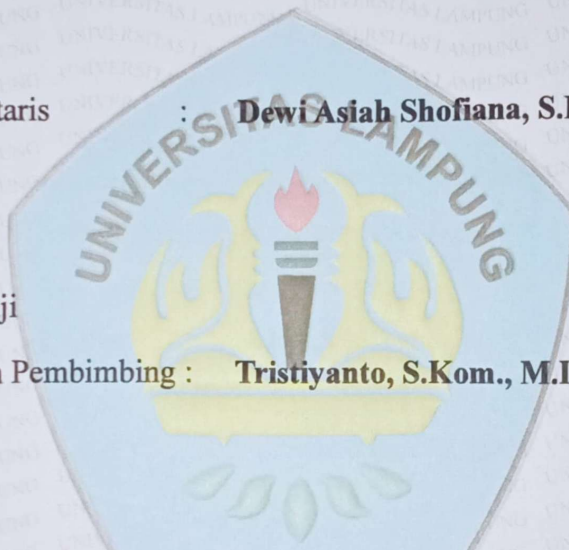
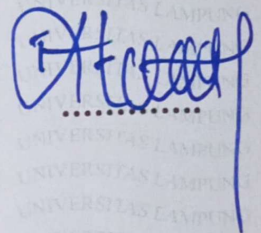


Sekretaris : **Dewi Asiah Shofiana, S.Komp., M.Kom.**



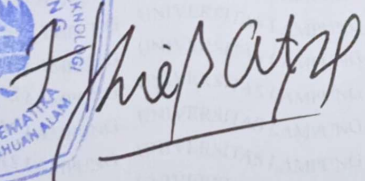
Penguji

Bukan Pembimbing : **Tristiyanto, S.Kom., M.I.S., Ph.D.**



2. Dekan Fakultas Matematika dan Ilmu Pengetahuan Alam




Dr. Eng. Heri Satria, S.Si., M.Si.

NIP. 197110012005011002

Tanggal Lulus Ujian Skripsi: **17 Juni 2025**

PERNYATAAN

Saya bertanda tangan di bawah ini:

Nama : Nindi Restu Pembayun

NPM : 2117051021

Dengan ini menyatakan bahwa skripsi saya yang berjudul **“Klasifikasi Kanker Paru-paru Menggunakan *Deep Learning* dengan Arsitektur DenseNet pada Dataset IQ-QTH/NCCD”** merupakan hasil karya saya sendiri dan bukan karya orang lain. Seluruh tulisan yang tertuang di skripsi ini telah mengikuti kaidah penulisan karya ilmiah Universitas Lampung. Apabila di kemudian hari terbukti skripsi saya merupakan hasil penjiplakan atau dibuat orang lain, maka saya bersedia menerima sanksi berupa pencabutan gelar yang telah saya terima.

Bandar Lampung, 18 Juni 2025



Nindi Restu Pembayun
NPM. 2117051021

RIWAYAT HIDUP



Lahir di Sukaraja Nuban pada tanggal 3 Januari 2003 sebagai anak pertama dari pasangan Bapak Agus Riyadi dan Ibu Diah Lismiati. Penulis menyelesaikan pendidikan dasar di SD Negeri 2 Raman Aji dan selesai pada tahun 2015. Pendidikan menengah pertama ditempuh di MTs Negeri 2 Lampung Timur dan selesai pada tahun 2018, kemudian melanjutkan ke SMA Negeri 1 Kota Gajah dan selesai pada tahun 2021.

Pada tahun 2021 penulis terdaftar sebagai mahasiswa Program Studi S1 Ilmu Komputer di Universitas Lampung melalui jalur Seleksi Nasional Masuk Perguruan Tinggi Negeri (SNMPTN). Selama menjadi mahasiswa, penulis melakukan beberapa kegiatan antara lain:

1. Menjadi Asisten Dosen untuk mata kuliah Pengenalan Pola di Jurusan Ilmu Komputer pada tahun 2024.
2. Melakukan Kerja Praktik di Kantor Badan Pengawasan Keuangan dan Pembangunan (BPKP) Provinsi Lampung periode 2023/2024.
3. Mengikuti Kuliah Kerja Nyata (KKN) pada periode II tahun 2024 di Desa Braja Harjosari, Kecamatan Braja Seleh, Kabupaten Lampung Timur.

MOTTO

*“Allah tidak mengatakan hidup ini mudah. Tetapi Allah berjanji, bahwa
sesungguhnya bersama kesulitan ada kemudahan”*

(QS. Al-Insyirah : 5-6)

*“Allah tidak membebani seseorang melainkan sesuai dengan
kesanggupannya”*

(QS. Al-Baqarah : 286)

*“Tugas kita bukanlah untuk berhasil, tugas kita adalah untuk mencoba. Karena
di dalam mencoba kita akan menemukan dan belajar membangun kesempatan
untuk berhasil”*

(Buya Hamka)

PERSEMBAHAN

Alhamdulillahirabbil'alamiin

Segala puji bagi Allah Subhanahu Wa Ta'ala atas limpahan rahmat dan karunia-Nya, sehingga penulisan skripsi ini dapat diselesaikan dengan baik. Shalawat dan salam senantiasa tercurah kepada Nabi Muhammad Shallallahu 'Alaihi Wasallam, teladan mulia bagi seluruh umat.

Skripsi ini saya persembahkan kepada:

Kedua orang tua saya tercinta, atas doa, kasih sayang, dan dukungan yang tidak pernah surut sepanjang perjalanan studi ini.

Seluruh rekan yang telah menjadi bagian penting dalam proses perjuangan dan penyelesaian skripsi ini.

Diri saya sendiri, sebagai bentuk penghargaan atas ketekunan dan komitmen dalam menyelesaikan pendidikan di Program Studi S1 Ilmu Komputer, Universitas Lampung.

SANWACANA

Segala puji bagi Allah Subhanahu Wa Ta'ala atas limpahan rahmat, taufik, dan hidayah-Nya. Shalawat dan salam senantiasa tercurah kepada Nabi Muhammad Shallallahu 'Alaihi Wasallam, sehingga penulis dapat menyelesaikan skripsi yang berjudul “ **Klasifikasi Kanker Paru-paru Menggunakan *Deep Learning* dengan Arsitektur DenseNet pada Dataset IQ-QTH/NCCD**” dengan baik dan tepat waktu.

Pada kesempatan ini, penulis mengucapkan terima kasih kepada semua pihak yang telah berkontribusi dalam penyusunan skripsi ini, yaitu:

1. Allah Subhanahu wa Ta'ala, atas segala limpahan rahmat, hidayah, dan keridhaan-Nya yang menjadi sumber kekuatan serta kemudahan bagi penulis dalam menyelesaikan penulisan karya ilmiah ini dengan lancar dan tepat waktu.
2. Kedua orang tua penulis, Bapak Agus Riyadi dan Ibu Diah Lismiati, yang senantiasa menjadi sumber doa, semangat, dan kepercayaan dalam setiap langkah penulis. Terima kasih atas kasih sayang dan dukungan tanpa henti hingga skripsi ini dapat diselesaikan.
3. Adik-adik penulis, Maura Hellenia dan Akhtar Attaya El Fatih, atas segala bentuk dukungan, perhatian, dan doa yang telah diberikan selama proses penyusunan skripsi ini.
4. Bapak Dwi Sakethi, S.Si., M.Kom. selaku Ketua Jurusan Ilmu Komputer FMIPA Universitas Lampung.
5. Bapak Tristiyanto, S.Kom., M.I.S., Ph.D. selaku Ketua Program Studi Ilmu Komputer FMIPA Universitas Lampung.

6. Bapak Dr. rer. nat. Akmal Junaidi, M.Sc. selaku dosen Pembimbing Utama atas kesabaran dan dedikasinya dalam membimbing, memberikan arahan, ilmu, serta koreksi yang sangat berharga selama proses penyusunan skripsi ini.
7. Ibu Dewi Asiah Shofiana, S.Komp., M.Kom. selaku dosen Pembimbing Kedua atas waktu serta bimbingan yang telah diberikan selama proses penyusunan skripsi, yang sangat membantu dalam penyusunan karya ini.
8. Bapak Prof. Admi Syarif, Ph.D, Bapak Tristiyanto, S.Kom., M.I.S., Ph.D. dan Bapak Febi Eka Febriansyah, M.T. selaku dosen Pembahas yang telah memberikan kritik, saran, dan masukan yang sangat berarti bagi penyempurnaan skripsi ini.
9. Bapak/Ibu Dosen dan seluruh Staf Jurusan Ilmu Komputer FMIPA Universitas Lampung yang telah memberikan ilmu, serta membantu dalam berbagai urusan akademik maupun administratif.
10. Faiz Muzaki yang selalu membantu serta memberikan dukungan dan kebersamaan yang tulus di setiap langkah penulis dalam menyelesaikan skripsi ini.
11. Tasya dan Yuna selaku rekan seperjuangan yang selalu membersamai penulis dengan semangat, dukungan, dan kolaborasi selama proses penyusunan skripsi ini.
12. Anggota Grup Whatsapp Sahabat Jannah atas dukungan, doa, dan bantuan yang telah diberikan kepada penulis.

Penulis menyadari bahwa skripsi ini masih memiliki keterbatasan. Oleh karena itu, kritik dan saran yang membangun sangat diharapkan untuk perbaikan di masa mendatang. Semoga karya ini dapat memberikan manfaat bagi semua pihak yang membutuhkan.

Bandar Lampung, 18 Juni 2025

Nindi Restu Pembayun
NPM. 2117051021

DAFTAR ISI

	Halaman
DAFTAR ISI.....	i
DAFTAR TABEL	v
DAFTAR GAMBAR	vi
I. PENDAHULUAN	1
1.1 Latar Belakang	1
1.2 Rumusan Masalah	2
1.3 Batasan Masalah.....	3
1.4 Tujuan Penelitian.....	3
1.5 Manfaat Penelitian	3
II. TINJAUAN PUSTAKA.....	4
2.1 Penelitian Terdahulu.....	4
2.2 Kanker Paru-Paru	6
2.3 <i>CT Scan</i>	7
2.4 <i>Deep Learning</i>	8
2.5 <i>Convolutional Neural Network (CNN)</i>	9
2.5.1 <i>Convolutional Layer</i>	10
2.5.2 <i>Pooling Layer</i>	10
2.5.3 <i>Fully Connected Layer</i>	11
2.6 DenseNet.....	12
2.6.1 <i>Input Layer</i>	14
2.6.2 Lapisan Konvolusi Awal dan <i>Pooling</i>	14

2.6.3	<i>Dense Block</i>	15
2.6.4	<i>Transition Layer</i>	17
2.6.5	<i>Growth Rate (k)</i>	19
2.6.6	<i>Bottleneck Layer</i>	20
2.7	<i>Confusion Matrix</i>	20
2.7.1	<i>Accuracy</i>	21
2.7.2	<i>Precision</i>	21
2.7.3	<i>Recall</i>	21
2.7.4	<i>F1-Score</i>	22
III.	METODE PENELITIAN	23
3.1	Tempat dan Waktu Penelitian.....	23
3.2	Data Penelitian	23
3.3	Alat Pendukung.....	24
3.3.1	Perangkat Keras (<i>Hardware</i>).....	24
3.3.2	Perangkat Lunak (<i>Software</i>)	25
3.4	Alur Kerja Penelitian.....	28
3.4.1	<i>Data Acquisition</i>	28
3.4.2	<i>Data Splitting</i>	29
3.4.3	<i>Data Preprocessing</i>	29
3.4.4	<i>Model Architecture</i> (DenseNet)	30
3.4.5	<i>Model Training</i>	31
3.4.6	<i>Model Testing</i>	32
3.4.7	<i>Model Evaluation</i>	32
IV.	HASIL DAN PEMBAHASAN	33
4.1	<i>Dataset</i>	33
4.2	<i>Preprocessing</i>	33

4.2.1	<i>Resizing Image</i>	33
4.2.2	Normalisasi Piksel.....	34
4.2.3	Augmentasi Data	34
4.3	Data <i>Splitting</i>	36
4.4	Arsitektur Model DenseNet	37
4.4.1	Pemilihan Model DenseNet	37
4.4.2	Modifikasi Arsitektur	37
4.5	Model <i>Training</i>	39
4.5.1	<i>Optimizer</i>	39
4.5.2	<i>Loss Function</i>	40
4.5.3	Metrik Evaluasi	40
4.5.4	<i>Early Stopping</i>	40
4.5.5	Proses Pelatihan Model	41
4.6	Hasil Pelatihan dan Evaluasi Model	41
4.6.1	Evaluasi Model DenseNet-121	42
4.6.2	Evaluasi Model DenseNet-169	44
4.6.3	Evaluasi Model DenseNet-201	47
4.7	Perbandingan Kinerja Antar Model	49
4.8	Analisis <i>Missclassification</i>	51
4.8.1	DenseNet-121.....	52
4.8.2	DenseNet-169.....	54
4.8.3	DenseNet-201.....	56
4.9	Konfigurasi dan <i>Tuning</i> Parameter.....	58
V.	SIMPULAN DAN SARAN	60
5.1	Simpulan	60
5.2	Saran.....	61

DAFTAR PUSTAKA.....	63
LAMPIRAN.....	69

DAFTAR TABEL

Tabel	Halaman
1. Penelitian Terdahulu.....	5
2. <i>Confusion Matrix</i>	20
3. Kelas <i>Dataset IQ-QTH/NCCD</i>	24
4. Sampel Augmentasi.....	36
5. Perbandingan Akurasi, <i>Recall</i> dan <i>Macro F1-Score</i>	51

DAFTAR GAMBAR

Gambar	Halaman
1. <i>CT Scan</i> Paru-Paru.	7
2. <i>Layer Deep Learning</i> (Bhatt et al., 2021).	8
3. Model Lapisan CNN (Alzubaidi et al., 2021).	9
4. <i>Pooling Layer</i> (Putri et al., 2024).	11
5. <i>Fully Connected Layer</i> (Putri et al., 2024).	12
6. Arsitektur DenseNet (Zhou et al., 2022).	13
7. <i>Dense Block</i> (Zhou et al., 2022).	16
8. <i>Transition Layer</i> (Zhou et al., 2022).	18
9. Alur Kerja Penelitian.	28
10. <i>Classifier</i>	38
11. <i>Optimizer</i>	40
12. <i>Early Stopping</i>	41
13. Pelatihan Model	41
14. Grafik Akurasi dan <i>Loss</i> DenseNet-121 pada Data <i>Training</i>	43
15. <i>Confusion Matrix</i> DenseNet-121.	43
16. ROC DenseNet-121.	44
17. Grafik Akurasi dan <i>Loss</i> DenseNet-169 pada Data <i>Training</i>	45
18. <i>Confusion Matrix</i> DenseNet-169.	46
19. ROC DenseNet-169	46
20. Grafik Akurasi dan <i>Loss</i> DenseNet-201 pada Data <i>Training</i>	48
21. <i>Confusion Matrix</i> DenseNet-201.	48
22. ROC DenseNet-201.	49
23. <i>Confusion Matrix</i> Model DenseNet-121.	53

24. <i>Misclassification</i> DenseNet-121.	53
25. <i>Confusion Matrix</i> Model DenseNet-169.	55
26. <i>Misclassification</i> DenseNet-169.	55
27. <i>Confusion Matrix</i> Model DenseNet-201.	56
28. <i>Misclassification</i> DenseNet-201.	57

I. PENDAHULUAN

1.1 Latar Belakang

Kanker paru-paru merupakan salah satu jenis kanker paling mematikan di dunia, dengan tingkat mortalitas yang tinggi terutama karena sering terdiagnosis pada tahap lanjut. Pada tahun 2013, kanker paru-paru menyumbang 13,7% dari total kasus kanker baru di Amerika Serikat dan 27,5% dari seluruh kematian akibat kanker (Kareem et al., 2021). Faktor risiko utama meliputi kebiasaan merokok, paparan polusi udara, serta zat karsinogenik lainnya. Deteksi dini menjadi sangat krusial karena memungkinkan penanganan yang lebih efektif dan dapat meningkatkan harapan hidup pasien. Namun, metode deteksi konvensional seringkali memiliki keterbatasan dalam hal akurasi dan efisiensi waktu.

Seiring perkembangan teknologi, kecerdasan buatan (*Artificial Intelligence*), khususnya *deep learning*, telah menjadi solusi potensial dalam mendukung diagnosis kanker paru-paru. Teknologi ini memungkinkan analisis citra medis secara otomatis dengan tingkat akurasi yang tinggi. Salah satu arsitektur *deep learning* yang populer adalah DenseNet, yang menghubungkan setiap lapisan ke semua lapisan setelahnya, memungkinkan informasi dan fitur mengalir secara lebih efektif. Pendekatan ini membantu mencegah hilangnya informasi penting selama proses pelatihan dan meningkatkan efisiensi dalam pembelajaran model (Sait, 2023).

Beberapa penelitian terdahulu telah menunjukkan keberhasilan penerapan *deep learning* dalam deteksi kanker paru-paru. Misalnya, penelitian oleh Al-Yasriy et al (2020) menggunakan CNN arsitektur AlexNet untuk menganalisis citra *CT scan* pada *dataset IQ-QTH/NCCD* dan memperoleh akurasi 93,548%. *Dataset* tersebut terdiri dari 1190 citra yang dikategorikan ke dalam tiga kelas: normal, jinak, dan

gan. Selain itu, meta-analisis yang dilakukan oleh Wang (2022) mencatat bahwa sensitivitas dan spesifisitas model berbasis *AI* mencapai rata-rata 87%, menjadikannya alat yang menjanjikan untuk deteksi dini kanker paru-paru. Penelitian lain oleh Gayap & Akhloufi (2024) juga menunjukkan bahwa arsitektur lanjutan seperti CNN 3D dan *Vision Transformer* (ViT) dapat meningkatkan akurasi klasifikasi kanker paru-paru pada *dataset* publik seperti LIDC dan LUNA16. Penelitian ini menekankan bahwa model-model ini mampu mengatasi keterbatasan metode tradisional dengan mengenali pola-pola yang kompleks dan memberikan hasil yang lebih akurat.

Berdasarkan hal tersebut, penelitian ini bertujuan untuk mengembangkan model klasifikasi kanker paru-paru berbasis arsitektur DenseNet dengan memanfaatkan *dataset IQ-QTH/NCCD*, yang memiliki tantangan berupa distribusi kelas yang tidak seimbang dan kompleksitas visual pada citra *CT scan*. Penelitian ini mengeksplorasi keunggulan DenseNet dalam mempertahankan fitur penting antar lapisan guna meningkatkan akurasi klasifikasi. Diharapkan model ini tidak hanya memberikan hasil klasifikasi yang lebih akurat, tetapi juga mampu mendeteksi jenis kanker paru-paru sejak tahap awal. Dengan demikian, penelitian ini diharapkan dapat memberikan kontribusi terhadap pengembangan sistem pendukung diagnosis berbasis *AI* dalam dunia medis.

1.2 Rumusan Masalah

Rumusan masalah dalam penelitian ini adalah sebagai berikut :

- a. Bagaimana performa arsitektur DenseNet dalam klasifikasi kanker paru-paru menggunakan *dataset IQ-QTH/NCCD*?
- b. Apakah arsitektur DenseNet mampu menangani kompleksitas *dataset IQ-QTH/NCCD* dan menghasilkan akurasi yang kompetitif?

1.3 Batasan Masalah

Penelitian ini dibatasi oleh beberapa hal, yaitu:

- a. *Dataset* yang digunakan berasal dari Kaggle, yang terdiri dari tiga kelas yaitu normal, jinak, dan ganas. *Dataset* ini mungkin tidak mencakup semua variasi kondisi kanker paru-paru.
- b. Variasi kualitas citra, seperti jumlah irisan per kasus dan resolusi citra, dapat memengaruhi kinerja model klasifikasi.
- c. Penelitian ini hanya berfokus pada klasifikasi kanker paru-paru menggunakan arsitektur DenseNet tanpa melakukan perbandingan dengan metode lain serta tidak mencakup pengembangan aplikasi atau sistem berbasis *Artificial Intelligence* (AI).

1.4 Tujuan Penelitian

Tujuan penelitian ini adalah:

- a. Menganalisis performa klasifikasi kanker paru-paru menggunakan arsitektur DenseNet pada *dataset IQ-QTH/NCCD*.
- b. Mengevaluasi akurasi dan efektivitas DenseNet berdasarkan kestabilan pelatihan dan kemampuan generalisasi terhadap data uji.

1.5 Manfaat Penelitian

Adapun manfaat penelitian ini adalah sebagai berikut:

- a. Memberikan kontribusi dalam penerapan model *deep learning*, khususnya DenseNet, untuk klasifikasi kanker paru-paru menggunakan *dataset IQ-QTH/NCCD*.
- b. Mengevaluasi kinerja model DenseNet dalam klasifikasi kanker paru-paru berdasarkan *dataset IQ-QTH/NCCD*, yang dapat menjadi referensi bagi penelitian atau pengembangan lebih lanjut di bidang kesehatan.

II. TINJAUAN PUSTAKA

2.1 Penelitian Terdahulu

Dalam beberapa tahun terakhir, penggunaan *deep learning*, terutama *Convolutional Neural Network* (CNN), semakin banyak dimanfaatkan untuk menyelesaikan berbagai masalah klasifikasi gambar. Salah satu arsitektur CNN yang populer adalah DenseNet, yang dikenal karena kemampuannya meningkatkan efisiensi dan akurasi dalam proses klasifikasi. Beberapa penelitian berikut menggambarkan penerapan arsitektur DenseNet untuk berbagai kebutuhan klasifikasi data visual.

Penelitian berjudul “*DenseNet_HybWWoA: A DenseNet-Based Brain Metastasis Classification*” yang dilakukan oleh Alshammari (2023), bertujuan untuk mengklasifikasikan jenis tumor otak berdasarkan citra MRI. *Dataset* yang digunakan terdiri dari 3064 gambar MRI yang terbagi menjadi tiga kelas tumor, yaitu *glioma*, *meningioma*, dan *pituitary* tumor. Penelitian ini menggunakan model DenseNet yang dioptimalkan dengan algoritma *Hybrid Whale and Water Waves Optimization Algorithm (HybWWoA)*. Dari penelitian ini, didapatkan hasil akurasi sebesar 92.1%, presisi sebesar 98.5%, dan *F1-score* sebesar 97%.

Penelitian selanjutnya yang berjudul “*DenseNet Architecture Implementation for Organic and Non-Organic Waste*” dilakukan oleh Simarmata et al (2023), bertujuan untuk mengklasifikasikan limbah organik dan non-organik menggunakan arsitektur DenseNet. *Dataset* yang digunakan terdiri dari 22,500 gambar limbah yang terbagi dalam dua kelas yaitu organik dan non-organik. Penelitian ini menggunakan beberapa varian DenseNet, yaitu DenseNet121, DenseNet169, dan DenseNet201.

Hasil penelitian menunjukkan bahwa model DenseNet121 memberikan performa terbaik dengan akurasi sebesar 93.1%, presisi sebesar 94.08%, *recall* sebesar 94%, dan *F1-score* sebesar 94.03%.

Selanjutnya, penelitian “*A Deep Learning Using DenseNet201 to Detect Masked or Non-masked Face*” dilakukan oleh Adhinata et al (2021), bertujuan untuk mendeteksi wajah dengan atau tanpa masker menggunakan metode *deep learning* berbasis DenseNet201. *Dataset* yang digunakan terdiri dari 90,000 gambar wajah tanpa masker dan 2203 gambar wajah dengan masker. Penelitian ini menghasilkan akurasi tinggi, yaitu sebesar 99%, dengan *F-measure* sebesar 0.98.

Penelitian terakhir berjudul “*Klasifikasi Penyakit Tanaman Padi Menggunakan Metode CNN Arsitektur DenseNet-121*” yang dilakukan oleh Agustina et al (2024), bertujuan untuk mengidentifikasi penyakit pada tanaman padi berdasarkan gambar daun. *Dataset* yang digunakan terdiri dari 750 gambar daun padi yang terbagi dalam tiga kelas, yaitu *brown spot*, *blast*, dan *bacterial leaf blight*. Penelitian ini menggunakan arsitektur DenseNet121 dengan augmentasi data. Hasil penelitian menunjukkan akurasi terbaik sebesar 99.17% dengan augmentasi, sedangkan tanpa augmentasi mencapai 95%. Penelitian terdahulu yang berkaitan dengan penelitian skripsi ini secara ringkas dapat dilihat pada Tabel 1.

Tabel 1. Penelitian Terdahulu.

Peneliti	Judul	Data	Metode	Hasil
Alshammari (2023)	<i>DenseNet_ HybWWoA: A DenseNet-Based Brain Metastasis Classification</i>	Terdiri dari 3064 gambar MRI, terbagi menjadi 3 jenis tumor (glioma, meningioma, pituitary tumor)	DenseNet + Hybrid Whale and Water Waves Optimization Algorithm (HybWWoA)	Nilai akurasi sebesar 92.1%, presisi 98.5%, <i>F1-Score</i> 97%
Simarmata et al. (2023)	<i>DenseNet Architecture Implementation for Organic and</i>	22,500 gambar sampah, terdapat 2 kelas (organik dan non-organik)	DenseNet121, DenseNet169, DenseNet201	DenseNet121: Akurasi 93.1%, presisi 94.08%, <i>recall</i>

Peneliti	Judul	Data	Metode	Hasil
	<i>Non-Organic Waste</i>			94%, <i>F1-Score</i> 94.03%
Adhinata et al. (2021)	<i>A Deep Learning Using DenseNet201 to Detect Masked or Non-masked Face</i>	90,000 gambar wajah tanpa masker, 2203 gambar wajah dengan masker	DenseNet201	Akurasi 99%, <i>F-measure</i> 0.98
Agustina et al. (2024)	Klasifikasi Penyakit Tanaman Padi Menggunakan Metode CNN Arsitektur DenseNet-121	750 gambar daun padi, 3 kelas: <i>brown spot</i> , <i>blast</i> , <i>bacterial leaf blight</i>	DenseNet 121 + Augmentasi Data	Akurasi terbaik 99.17% dengan augmentasi, 95% tanpa augmentasi

2.2 Kanker Paru-Paru

Paru-paru merupakan organ pada sistem pernapasan yang berperan penting sebagai tempat pertukaran oksigen dan karbondioksida dalam tubuh (Juniati et al., 2021). Namun pada era *modern* sekarang ini banyak orang yang memiliki pola hidup yang bisa dikatakan tidak sehat. Pola hidup yang tidak sehat ini meliputi kebiasaan merokok, makan makanan instan, kurangnya aktivitas berolahraga dan lain sebagainya. Kebiasaan yang tidak sehat ini berdampak pada kesehatan, salah satunya adalah kanker paru-paru.

Dalam penelitian yang dilakukan oleh Joseph & Rotty (2020), kanker paru-paru adalah semua jenis penyakit keganasan yang terjadi di paru-paru. Penyakit ini mencakup keganasan yang berasal langsung dari paru-paru itu sendiri (kanker paru primer) dan yang berasal dari organ lain kemudian menyebar ke paru-paru (kanker paru metastasis). Secara klinis, kanker paru primer biasanya berupa tumor ganas yang berasal dari epitel bronkus, yang sering disebut sebagai karsinoma bronkus.

Kanker paru-paru secara umum terbagi menjadi dua jenis utama: *Small Cell Lung Cancer* (SCLC) dan *Non-Small Cell Lung Cancer* (NSCLC). SCLC, yang lebih

agresif, tumbuh dan menyebar dengan cepat. NSCLC, yang mencakup sekitar 85% dari semua kasus kanker paru-paru, terdiri atas tiga subtype utama: *adenokarsinoma*, *karsinoma sel skuamosa*, dan *karsinoma sel besar* (Sembiring et al., 2023). *Adenokarsinoma* adalah jenis NSCLC yang paling umum, berasal dari sel yang menghasilkan lendir di paru-paru, sering ditemukan pada perokok maupun non-perokok (Sankar et al., 2023).

2.3 CT Scan

Computed Tomography (CT) scan merupakan perangkat pencitraan standar yang mampu mengidentifikasi berbagai karakteristik spesifik dari nodul paru, yaitu benjolan kecil yang muncul di paru-paru dan bisa bersifat jinak maupun ganas, yang terdiagnosis, seperti kelainan pembuluh darah (*arteriovenous fistulae*), *rounded atelectasis*, *fungus balls*, *mucoïd impaction*, dan *infarct*. Selain itu, *CT scan* juga sangat berguna dalam mengevaluasi seluruh rongga dada selama proses penilaian nodul. Pemeriksaan ini dapat membantu mendeteksi kelainan dan menyoroti tanda-tanda penyakit di paru-paru, terutama pada pasien dengan gejala seperti batuk, nyeri dada, dan demam (Nugroho et al., 2023). Contoh *CT Scan* dapat dilihat pada Gambar 1.

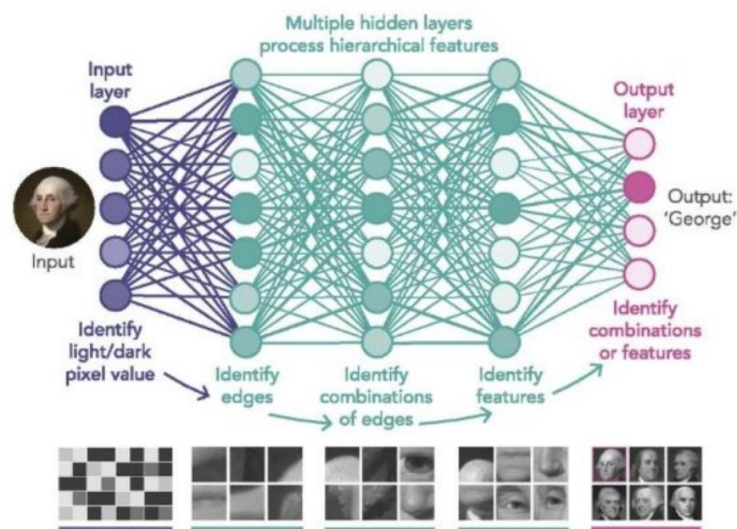


Gambar 1. *CT Scan* Paru-Paru.

2.4 Deep Learning

Deep learning adalah cabang dari pembelajaran mesin yang menggunakan jaringan saraf tiruan untuk meniru cara kerja otak manusia. Teknologi ini dirancang untuk memahami pola dari data yang tidak terstruktur, seperti gambar, suara, atau teks, sehingga sangat efektif dalam menangani masalah-masalah kompleks. Berbeda dengan pembelajaran mesin tradisional yang mengandalkan metode statistik, *deep learning* memanfaatkan *dataset* besar untuk melatih modelnya, memungkinkan hasil yang lebih akurat dan efisien (Sharifani & Amini, 2023).

Salah satu keunggulan utama *deep learning* adalah kemampuannya secara otomatis mengenali pola dan mengekstraksi informasi penting dari data tanpa perlu proses manual yang rumit. Teknologi ini didukung oleh jaringan saraf yang memiliki banyak lapisan tersembunyi, sehingga mampu menangani hubungan yang sangat kompleks. Performanya juga terus meningkat seiring bertambahnya data pelatihan, menjadikannya alat yang sangat efisien untuk menyelesaikan berbagai tantangan di era *big data*. Dengan skalabilitas yang tinggi dan kemampuan untuk mengolah data besar, *deep learning* kini menjadi pilar utama dalam pengembangan kecerdasan buatan modern (Dong et al., 2021). *Layer Deep Learning* dapat dilihat pada Gambar 2 berikut.

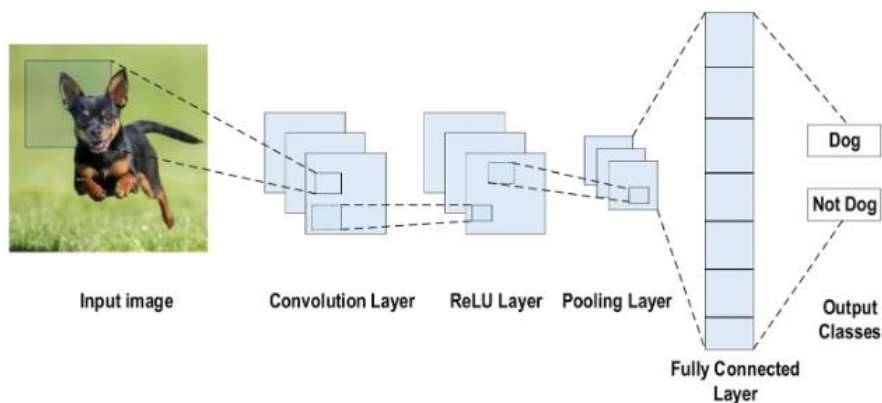


Gambar 2. *Layer Deep Learning* (Bhatt et al., 2021).

2.5 Convolutional Neural Network (CNN)

Convolutional Neural Network (CNN) adalah salah satu metode pembelajaran mendalam (*deep learning*) yang dirancang untuk memproses data dalam bentuk *grid*, seperti citra atau video. CNN meniru cara kerja korteks visual manusia dengan mengenali pola visual melalui serangkaian lapisan yang melibatkan operasi konvolusi. Metode ini menggunakan bobot dan bias yang dapat dipelajari untuk menetapkan tingkat kepentingan pada berbagai fitur citra, seperti bentuk, garis, dan warna, sehingga mampu membedakan satu objek dari yang lain (Bhatt et al., 2021).

Struktur dasar CNN terdiri dari satu atau lebih lapisan konvolusi, lapisan *pooling*, dan lapisan *fully connected*. Struktur ini memungkinkan CNN untuk memanfaatkan data berbentuk *grid*, seperti gambar 2D, dengan cara mengekstraksi fitur-fitur penting dari gambar melalui operasi konvolusi. Operasi ini menggunakan kernel yang bergerak di atas citra *input* untuk mendeteksi pola-pola visual seperti tepi, tekstur, atau bentuk. Lapisan *pooling* kemudian merangkum informasi tersebut dengan mengurangi dimensi data, sementara lapisan *fully connected* bertugas menggabungkan fitur-fitur yang telah diekstraksi untuk menghasilkan prediksi akhir (Memon et al., 2021). Arsitektur CNN dapat dilihat pada Gambar 3 berikut.



Gambar 3. Model Lapisan CNN (Alzubaidi et al., 2021).

2.5.1 Convolutional Layer

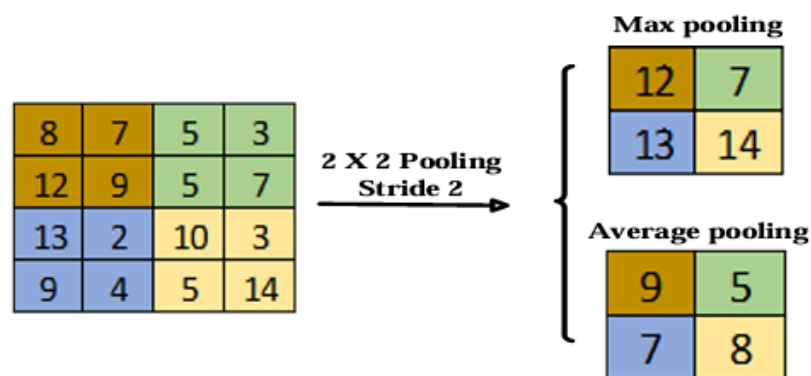
Convolutional Layer merupakan komponen utama dalam arsitektur *Convolutional Neural Network* (CNN) yang bertugas melakukan proses konvolusi pada citra masukan. Lapisan ini menggunakan *filter* untuk mengekstraksi fitur-fitur penting dari citra, seperti tepi, sudut, dan pola tekstur tertentu, yang kemudian menghasilkan peta fitur (*feature map*) atau peta aktivasi (*activation map*). Proses konvolusi ini memungkinkan sistem untuk mengenali pola-pola yang ada pada data citra dengan cara menggeser *filter* secara sistematis di seluruh area citra. Dengan memanfaatkan lapisan ini, CNN mampu menangkap informasi spasial dan hubungan lokal antar piksel yang menjadi dasar dalam memahami struktur data citra (Hanin et al., 2021).

2.5.2 Pooling Layer

Pooling Layer dalam *Convolutional Neural Network* (CNN) adalah lapisan yang berfungsi untuk memperkecil ukuran peta fitur (*feature map*) melalui proses *downsampling*. Proses ini dilakukan dengan menggunakan *filter* berukuran tertentu yang digeserkan pada area peta fitur. Metode *pooling* yang umum digunakan adalah *max pooling*, yang memilih nilai maksimum dari area tertentu, dan *average pooling*, yang menghitung rata-rata nilai dari area tersebut. Tujuan utama dari *Pooling Layer* adalah untuk mengurangi dimensi data sehingga mempercepat proses komputasi dan mengontrol risiko *overfitting* dengan menyederhanakan representasi fitur yang diekstraksi dari citra masukan (Prastowo, 2021).

Pooling Layer juga berfungsi untuk mengurangi dimensi peta fitur secara progresif tanpa kehilangan informasi penting, sehingga membantu mengurangi kompleksitas komputasi dan mengontrol risiko *overfitting*. Operasi *pooling* dilakukan dengan mengolah nilai-nilai piksel pada area tertentu menggunakan

metode statistik, seperti *max pooling*, yang memilih nilai maksimum, atau *mean pooling*, yang menghitung rata-rata nilai dalam area tersebut. Dengan cara ini, *pooling layer* mempertahankan fitur-fitur esensial sambil mengurangi resolusi data, sehingga mempercepat proses komputasi tanpa mengorbankan performa model (Hanin et al., 2021). Operasi sederhana dari *pooling layer* ditunjukkan oleh Gambar 4 berikut.



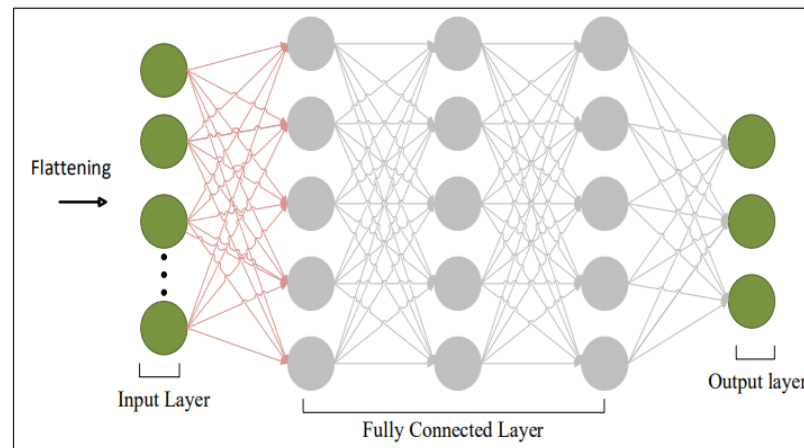
Gambar 4. *Pooling Layer* (Putri et al., 2024).

2.5.3 Fully Connected Layer

Fully Connected Layer adalah lapisan yang menghubungkan semua neuron dari lapisan sebelumnya dengan setiap neuron di lapisan ini, sehingga membentuk koneksi penuh antar neuron. Sebelum memasuki lapisan ini, data dari lapisan sebelumnya diubah menjadi bentuk satu dimensi (*flattening*) untuk memudahkan pemrosesan. Lapisan ini berperan penting dalam menggabungkan fitur-fitur yang telah diekstraksi oleh lapisan sebelumnya untuk menentukan keluaran berdasarkan data pelatihan. Dalam proses klasifikasi, *Fully Connected Layer* bertanggung jawab untuk mengidentifikasi fitur mana yang memiliki korelasi paling kuat dengan kelas tertentu, sehingga menghasilkan prediksi akhir yang akurat (Hanin et al., 2021).

Perbedaan utama antara *Fully Connected Layer* dengan *Convolutional Layer* adalah konektivitasnya. Pada *Convolutional Layer*, neuron hanya terhubung ke

area tertentu dari *input*, sedangkan pada *Fully Connected Layer*, setiap neuron terhubung ke seluruh neuron pada lapisan sebelumnya. Operasi di *Fully Connected Layer* melibatkan penghitungan bobot, bias, dan fungsi aktivasi untuk menghasilkan *output* akhir yang kemudian digunakan untuk klasifikasi atau prediksi lebih lanjut (Wulandari et al., 2020).



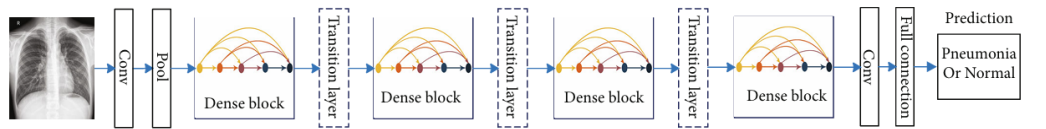
Gambar 5. *Fully Connected Layer* (Putri et al., 2024).

2.6 DenseNet

DenseNet merupakan arsitektur *deep learning* yang diperkenalkan oleh Gao Huang pada tahun 2017 dalam *paper* yang berjudul *Densely Connected Convolutional Networks*. Arsitektur ini berbasis *Convolutional Neural Network* (CNN) yang memanfaatkan koneksi padat antara lapisan-lapisan dalam jaringannya. Berbeda dengan CNN konvensional, di mana setiap lapisan hanya terhubung dengan lapisan berikutnya, DenseNet menghubungkan setiap lapisan ke semua lapisan sebelumnya. Jika suatu jaringan memiliki L lapisan, maka jumlah koneksi langsungnya adalah $L(L+1)/2$, jauh lebih banyak dibandingkan CNN biasa yang hanya memiliki L koneksi (Huang et al., 2017).

DenseNet memberikan beberapa keuntungan, seperti mengurangi jumlah parameter, memperkuat propagasi informasi, serta mendorong penggunaan ulang fitur, sehingga membuat model lebih efisien dan mudah dilatih (Wakili et al., 2022). DenseNet sering digunakan dalam berbagai tugas *computer vision* karena efisiensi

parameter yang lebih baik dibandingkan arsitektur lain. Arsitektur DenseNet dapat dilihat pada Gambar 6 berikut.



Gambar 6. Arsitektur *DenseNet* (Zhou et al., 2022).

Pada Gambar 6 dapat dilihat bahwa DenseNet memiliki beberapa *Dense Block* yang dipisahkan oleh *Transition Layer*. Lapisan ini membantu mengontrol dimensi fitur dan meningkatkan efisiensi komputasi dengan menggunakan 1×1 *Convolution* sebagai *Bottleneck Layer* serta 2×2 *Average Pooling*. Setelah melewati *Dense Block* terakhir, fitur diproses melalui *Global Average Pooling* (GAP) sebelum masuk ke *Fully Connected Layer*. Terakhir, *Softmax Classifier* digunakan untuk mengonversi hasil keluaran menjadi probabilitas untuk setiap kelas, sehingga memungkinkan model membuat prediksi akhir dengan memilih kelas dengan probabilitas tertinggi.

Selain itu, DenseNet memiliki beberapa varian, seperti DenseNet-121, DenseNet-169, dan DenseNet-201, di mana angka-angka tersebut menunjukkan jumlah total lapisan dalam model. Bunde & Danciu (2024), menyatakan bahwa setiap varian DenseNet memiliki jumlah parameter dan konfigurasi blok yang berbeda. Berikut perbedaan antar variannya :

a. DenseNet-121

DenseNet-121 merupakan varian yang memiliki total 121 lapisan dan sekitar 7 juta parameter. Model ini menggunakan konfigurasi *dense block* sebesar 6-12-24-16, yang terdiri dari 58 unit *bottleneck*. Setiap unit *bottleneck* terdiri atas dua buah *convolutional layer*, yaitu 1×1 dan 3×3 *convolution*, sehingga total *convolutional layer* dari *dense block* adalah $58 \times 2 = 116$ layer. Selain itu, model ini memiliki satu lapisan awal *convolution* 7×7 , tiga *transition layer* masing-masing dengan satu *convolutional layer* (1×1), dan satu *fully connected layer* di akhir. Dengan demikian, total keseluruhan lapisan adalah $1 + 116 + 3 + 1 = 121$ lapisan.

b. DenseNet-169

DenseNet-169 memiliki 169 lapisan dan sekitar 14 juta parameter. Konfigurasi bloknya adalah 6-12-32-32, yang menghasilkan total 82 unit *bottleneck*. Dengan dua *convolutional layer* per unit, jumlah total *convolutional layer* pada *dense block* adalah $82 \times 2 = 164$ lapisan. Ditambah satu lapisan *convolution* awal, tiga *transition layer*, dan satu *fully connected layer*, total lapisannya menjadi $1 + 164 + 3 + 1 = 169$ lapisan.

c. DenseNet-201

DenseNet-201 terdiri dari 201 lapisan dengan jumlah parameter sekitar 20 juta dan konfigurasi blok sebesar 6-12-48-32. Jumlah unit *bottleneck* pada varian ini adalah 98, dan jika setiap unit memiliki dua *convolutional layer*, maka total *convolutional layer* dari *dense block* adalah $98 \times 2 = 196$ lapisan. Jika ditambahkan dengan satu lapisan *convolution* awal, tiga *transition layer*, dan satu *fully connected layer*, maka total keseluruhan menjadi $1 + 196 + 3 + 1 = 201$ lapisan.

2.6.1 Input Layer

Lapisan *input* merupakan tahap awal dalam arsitektur DenseNet, yang menerima citra sebagai masukan. Sebelum masuk ke jaringan, citra terlebih dahulu melalui proses *preprocessing*, seperti *resizing*, normalisasi, dan transformasi lainnya agar sesuai dengan format yang dibutuhkan oleh model. Proses ini memastikan bahwa semua data masukan memiliki ukuran dan skala yang seragam, sehingga model dapat belajar secara optimal tanpa dipengaruhi oleh variasi ukuran atau intensitas piksel pada gambar (Pardede & Putra, 2020).

2.6.2 Lapisan Konvolusi Awal dan Pooling

Setelah tahap *preprocessing*, citra melewati lapisan konvolusi awal yang bertujuan mengekstraksi fitur dasar. Biasanya, lapisan ini menggunakan kernel berukuran 7×7 dengan *stride* 2, diikuti oleh operasi *Batch Normalization* dan

ReLU Activation. Kombinasi ini membantu dalam menangkap pola dasar dalam citra dan menormalkan *output* untuk stabilitas proses pelatihan (Pardede & Putra, 2020).

Setelah lapisan konvolusi pertama, diterapkan operasi *Max Pooling*. Dalam arsitektur DenseNet, umumnya digunakan *Max Pooling* dengan ukuran kernel 3×3 dan *stride* 2. Tujuannya adalah mengurangi beban komputasi sekaligus mempertahankan informasi yang paling menonjol dalam fitur yang telah diekstraksi. Dengan menggunakan operasi *pooling*, jaringan dapat lebih fokus pada fitur utama tanpa kehilangan informasi penting yang berkontribusi dalam klasifikasi (Ashraf et al., 2016).

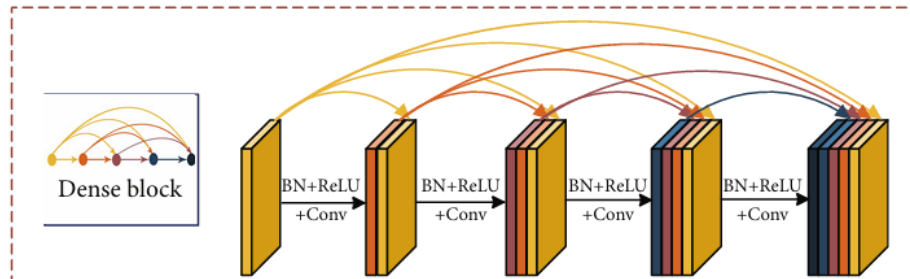
2.6.3 Dense Block

Dense Block merupakan komponen utama dalam arsitektur DenseNet yang dirancang untuk meningkatkan efisiensi pembelajaran fitur dengan cara menghubungkan setiap lapisan ke semua lapisan sebelumnya. Tidak seperti arsitektur konvolusi konvensional yang hanya menerima masukan dari lapisan sebelumnya, setiap lapisan dalam *Dense Block* menerima peta fitur dari seluruh lapisan sebelumnya dan meneruskan hasilnya ke semua lapisan berikutnya melalui operasi *concatenation* (penggabungan peta fitur). Dengan pendekatan ini, jika terdapat L lapisan dalam jaringan, maka akan terbentuk $L(L+1)/2$ koneksi langsung antar lapisan (Huang et al., 2017). Secara matematis, hubungan antar lapisan dalam *Dense Block* dapat dinyatakan dengan Persamaan 1 berikut.

$$x_l = H_l([x_0, x_1, \dots, x_{l-1}]) \quad (1)$$

Dari Persamaan tersebut, x_l adalah keluaran dari lapisan ke- l , $([x_0, x_1, \dots, x_{l-1}])$ menunjukkan penggabungan peta fitur dari lapisan 0 hingga $l-1$, dan H_l adalah fungsi komposit yang mencakup normalisasi *batch*, aktivasi ReLU, dan operasi konvolusi (Bundea & Danciu, 2024). Pendekatan ini tidak hanya mengatasi masalah *vanishing gradient*, tetapi juga meningkatkan efisiensi

pembelajaran dengan memungkinkan model menggunakan kembali fitur yang telah dipelajari di lapisan sebelumnya. Lapisan *Dense Block* dapat dilihat pada Gambar 7.



Gambar 7. *Dense Block* (Zhou et al., 2022).

Gambar 7 merupakan *Dense Block* 5 lapis dengan tingkat pertumbuhan atau *Growth Rate* (k) = 4. Setiap lapisan tersebut juga mengambil semua peta fitur sebelumnya sebagai *input*.

Di dalam penelitian yang dilakukan oleh Faizin et al (2022), setiap kombinasi lapisan *layer* dalam *Dense Block* memakai *batch* normalisasi, aktivasi ReLU serta konvolusi menggunakan filter 3x3. contohnya dapat dilihat pada Gambar 7. Berikut adalah tahapan kombinasi lapisan *layer*-nya:

1. *Batch Normalization* (BN) : *Batch Normalization* ini tahap pertama dari setiap lapisan dalam *Dense Block*. Setiap *input* berupa bentuk matriks piksel citra terlebih dahulu melewati tahap *Batch Normalization*. Pada *Dense Block*, ini terjadi sebelum aktivasi ReLU dan konvolusi. Fungsi dari proses ini bertujuan untuk menormalkan distribusi nilai dalam setiap *batch*, yang membantu mempercepat *konvergensi* dan mengurangi *overfitting* selama proses pembelajaran. Dalam prosesnya, setiap peta fitur dari lapisan sebelumnya terlebih dahulu dinormalisasi agar distribusi nilainya lebih stabil.
2. *ReLU Activation* : *ReLU Activation* dalam *Dense Block* letaknya setelah normalisasi *batch* dan sebelum konvolusi 3x3. Fungsi dari aktivasi ReLU untuk menambahkan non-linearitas ke dalam model, sehingga fitur yang diekstraksi lebih kompleks. Di dalam Faizin et al (2022), fungsi aktivasi

Rectified Linear Units (ReLU) digunakan untuk mengubah nilai x menjadi 0 jika nilai x tersebut bernilai negatif. Jika nilainya tidak kurang dari 0 atau positif, nilai tersebut akan tetap dipertahankan. Fungsi ReLU (*Rectified Linear Unit*) didefinisikan menggunakan Persamaan 2 berikut.

$$f(x) = \max(0, x) \quad (2)$$

Fungsi aktivasi ini bekerja dengan cara seperti berikut:

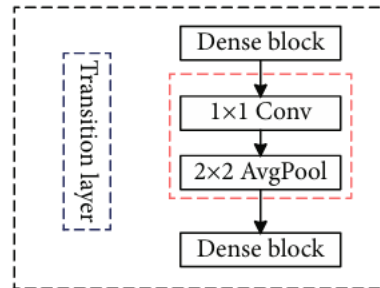
- a. Jika nilai *output* dari *Batch Normalization* negatif, maka diubah menjadi 0.
 - b. Jika nilai *output* nol atau positif, maka nilainya tetap dipertahankan. Dengan mekanisme ini, ReLU membantu jaringan dalam menangkap pola fitur yang lebih kompleks.
3. *3×3 Convolution* : Konvolusi 3×3 pada *Dense Block* terletak setelah aktivasi ReLU yang berfungsi untuk mengesktraksi fitur penting dari citra *input*. Proses dari konvolusi ini adalah data dari aktivasi ReLU dikalikan dengan kernel konvolusi ukuran 3×3 untuk mendeteksi pola fitur. *Output* dari konvolusi adalah peta fitur baru yang akan digunakan di lapisan berikutnya.

2.6.4 Transition Layer

Transition Layer adalah komponen penting yang terletak di antara dua *Dense Blocks* dalam arsitektur DenseNet. Fungsi utama *Transition Layer* adalah mengurangi ukuran peta fitur (*feature-map size*) melalui kombinasi operasi konvolusi 1x1 dan *pooling*. Dengan melakukan *downsampling* pada data, *Transition Layer* membantu mengontrol dimensi peta fitur sehingga menjaga efisiensi komputasi. Pendekatan ini memungkinkan model untuk menangani data berukuran besar tanpa memerlukan sumber daya yang berlebihan, dengan tetap mempertahankan informasi penting yang relevan (Huang et al., 2017).

Selain fungsi-fungsi tersebut, *Transition Layer* juga berperan penting dalam mencegah *overfitting*. Dengan mengurangi jumlah fitur dan resolusi peta fitur melalui operasi konvolusi 1×1 dan 2×2 *average pooling*, *Transition Layer*

membantu mengendalikan kompleksitas model dan jumlah parameter yang berlebihan, sehingga meningkatkan generalisasi model terhadap data baru (Zhou et al., 2022). Lapisan *Transition Layer* dapat dilihat pada Gambar 8 berikut.



Gambar 8. *Transition Layer* (Zhou et al., 2022).

Bedasarkan Gambar 8, setiap *Transition Layer* dalam DenseNet terdiri dari dua komponen berikut :

1. *1×1 Convolution (Bottleneck Layer)* : *1×1 Convolution (Bottleneck Layer)* terletak sebelum operasi *pooling* dan bertindak sebagai langkah pertama dalam *Transition Layer*. Secara teknis, operasi ini dilakukan dengan menerapkan kernel konvolusi berukuran 1×1 pada setiap elemen fitur yang berasal dari *Dense Block* sebelumnya, dengan tujuan mengurangi jumlah saluran (*channels*) fitur tanpa mengubah dimensi spasialnya. Hal ini bertindak sebagai *bottleneck* yang mengoptimalkan komputasi dengan menekan jumlah parameter, sekaligus mencegah lonjakan dimensi akibat konektivitas padat dalam DenseNet.
2. *2×2 Average Pooling* : Setelah proses konvolusi, fitur yang telah dikompresi kemudian melewati *2×2 Average Pooling*, yang berfungsi untuk *downsampling* dengan menggunakan kernel 2×2 dan *stride* 2. Operasi ini mengurangi resolusi peta fitur dengan cara menghitung rata-rata nilai dalam setiap blok 2×2 , sehingga menghemat memori dan mempercepat proses komputasi. Dalam implementasinya, *transition layer* dalam DenseNet menggunakan satu lapisan konvolusi yang diikuti oleh dua lapisan *average pooling* sebagai langkah transisi antara dua *dense*

block. Setelah *dense block* terakhir, dilakukan proses *global average pooling* untuk mereduksi dimensi fitur sebelum akhirnya diteruskan ke *softmax classifier* untuk menghasilkan prediksi akhir.

Dengan adanya *Transition Layer*, DenseNet dapat mempertahankan propagasi informasi yang efisien tanpa mengalami lonjakan dimensi yang berlebihan akibat konektivitas padat dalam *Dense Blocks*. Selain itu, pendekatan ini juga membantu dalam mengurangi jumlah *parameter*, yang membuat model lebih ringan dan efisien dibandingkan dengan arsitektur konvolusional lainnya. Dengan demikian, *Transition Layer* memainkan peran kunci dalam menjaga keseimbangan antara kompleksitas model dan efisiensi komputasi dalam DenseNet.

2.6.5 Growth Rate (k)

Growth Rate (k) dalam arsitektur DenseNet menentukan jumlah fitur baru yang ditambahkan oleh setiap lapisan dalam *Dense Block*. Jika suatu jaringan memiliki L lapisan dan *Growth Rate* sebesar k , maka jumlah total fitur pada lapisan ke- l dapat dihitung menggunakan Persamaan 3 berdasarkan penelitian yang dilakukan oleh (Huang et al., 2017).

$$k_0 + k \times (l - 1) \quad (3)$$

Dengan k_0 merupakan jumlah fitur awal, k adalah *Growth Rate*, dan l merupakan indeks lapisan. Konsep *Growth Rate* ini berbeda dari arsitektur konvensional seperti ResNet, yang menggunakan jumlah fitur tetap di setiap lapisan. Dengan *Growth Rate* yang lebih kecil, DenseNet dapat mengontrol jumlah parameter dalam jaringan, mengurangi redundansi, dan meningkatkan efisiensi komputasi (Huang et al., 2017).

Pada implementasi DenseNet, *Growth Rate* biasanya memiliki nilai antara 12 hingga 48. Nilai yang lebih kecil membuat jaringan lebih ringan dan hemat memori, sedangkan nilai yang lebih besar dapat meningkatkan kapasitas

representasi model, tetapi dengan konsekuensi kebutuhan komputasi yang lebih tinggi.

2.6.6 *Bottleneck Layer*

Untuk lebih meningkatkan efisiensi DenseNet, digunakan *Bottleneck Layer*, yang merupakan konvolusi 1×1 dan ditempatkan sebelum konvolusi 3×3 dalam setiap lapisan *Dense Block* (Huang et al., 2017). Fungsi utama dari *Bottleneck Layer* adalah sebagai berikut.

1. Mengurangi jumlah fitur sebelum konvolusi 3×3 , sehingga mengurangi beban komputasi.
2. Meningkatkan efisiensi parameter, karena lebih sedikit fitur yang diproses dalam konvolusi 3×3 .

Dengan adanya *Bottleneck Layer*, jumlah fitur yang diproses dalam konvolusi 3×3 dapat dikurangi menjadi seperempat dari jumlah fitur sebelumnya. Oleh karena itu, DenseNet dengan *Bottleneck Layer* sering disebut sebagai DenseNet-B, yang mampu mengurangi jumlah parameter jaringan secara signifikan tanpa mengorbankan akurasi.

2.7 *Confusion Matrix*

Confusion matrix merupakan suatu metode yang digunakan untuk mengevaluasi performa suatu model dalam klasifikasi data. *Confusion matrix* membantu dalam menghitung akurasi, presisi, *recall*, dan metrik evaluasi lainnya. *Confusion matrix* berbentuk tabel yang memberikan informasi mengenai jumlah prediksi yang benar dan salah untuk setiap kelas dalam suatu model klasifikasi. Bentuk tabel *confusion matrix* dapat dilihat pada Tabel 2.

Tabel 2. *Confusion Matrix*.

Prediksi \ Aktual	Positif	Negatif
Positif	TP (<i>True Positive</i>)	FP (<i>False Positive</i>)
Negatif	FN (<i>False Negative</i>)	TN (<i>True Negative</i>)

Keterangan:

- a. TP (*True Positive*): Model memprediksi positif, dan hasil sebenarnya juga positif.
- b. FP (*False Positive*): Model memprediksi positif, tetapi hasil sebenarnya negatif.
- c. FN (*False Negative*): Model memprediksi negatif, padahal hasil sebenarnya positif.
- d. TN (*True Negative*): Model memprediksi negatif, dan hasil sebenarnya juga negatif.

Kinerja model yang direpresentasikan melalui *confusion matrix* dapat dievaluasi menggunakan sejumlah metrik seperti *accuracy*, *precision*, *recall*, dan *F1-score*.

2.7.1 Accuracy

Akurasi adalah persentase ketepatan model dalam mengklasifikasikan data dengan benar dari seluruh data yang diuji. Akurasi dihitung menggunakan Persamaan 4 berikut:

$$Accuracy = \frac{TP + TN}{TP + FP + FN + TN} \quad (4)$$

2.7.2 Precision

Precision atau presisi adalah proporsi data yang diprediksi positif dengan benar dibandingkan dengan seluruh prediksi positif. Presisi dihitung menggunakan Persamaan 5 berikut:

$$Precision = \frac{TP}{TP + FP} \quad (5)$$

2.7.3 Recall

Recall atau *sensitivity* adalah proporsi data yang benar-benar positif dan berhasil diprediksi sebagai positif oleh model. *Recall* dihitung menggunakan Persamaan 6 berikut:

$$Recall = \frac{TP}{TP + FN} \quad (6)$$

2.7.4 *F1-Score*

F1-score adalah rata-rata harmonik antara *precision* dan *recall*. Nilai *F1-score* berkisar antara 0 hingga 1, di mana nilai 1 menunjukkan performa terbaik. *F1-score* dihitung menggunakan Persamaan 7 berikut:

$$F1-Score = 2 \times \frac{Precision \times Recall}{Precision + Recall} \quad (7)$$

III. METODE PENELITIAN

3.1 Tempat dan Waktu Penelitian

Penelitian ini dilakukan mulai dari bulan Desember tahun 2024 sampai bulan Juni tahun 2025 yang bertempat di Jurusan Ilmu Komputer Fakultas Matematika dan Ilmu Pengetahuan Alam Universitas Lampung, berlokasi di Jalan Prof. Dr. Soemantri Brojonegoro No. 1, Gedung Meneng, Bandar Lampung. Adapun *timeline* penelitian ini dapat dilihat pada Lampiran 1.

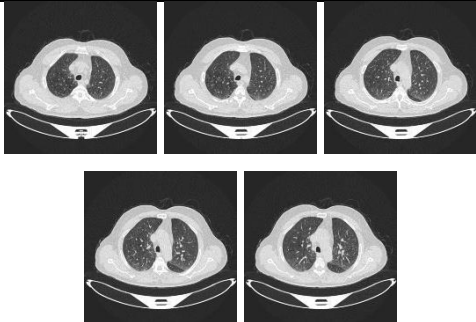
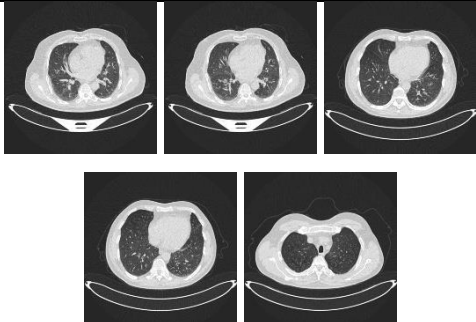
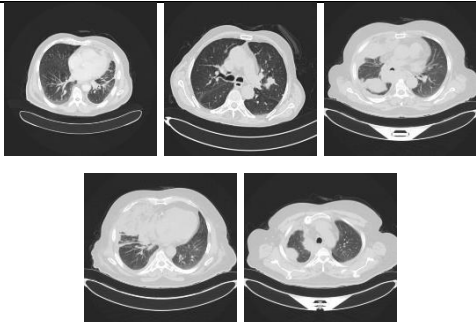
3.2 Data Penelitian

Data yang digunakan dalam penelitian ini merupakan data sekunder yang diambil dari *dataset* berjudul *The IQ-QTH/NCCD Lung Cancer Dataset*, yang tersedia di platform Kaggle dan diunggah oleh Ahmed Hamdallak dengan tautan berikut : (<https://www.kaggle.com/datasets/hamdallak/the-iqothnccd-lung-cancer-dataset/data>). *Dataset* ini dikembangkan oleh para ahli di *Iraq Oncology Teaching Hospital* dan *National Center for Cancer Diseases* pada tahun 2019.

Sampel dalam penelitian ini adalah seluruh citra *CT scan* yang terdapat dalam *dataset IQ-QTH/NCCD*. *Dataset* ini terdiri dari 1097 citra *grayscale* dengan resolusi 512×512 piksel, yang telah dianonimkan dan diproses dari format asli DICOM. *Dataset* ini diklasifikasikan menjadi tiga kategori atau kelas yaitu normal, jinak (*benign*), dan ganas (*malignant*). Terdapat 416 citra untuk kategori normal, 120 citra untuk kategori jinak (*benign*), dan 561 citra untuk kategori ganas (*malignant*).

Citra-citra dalam *dataset* ini merupakan hasil *CT scan* dada pasien dengan berbagai kondisi kesehatan. Data ini berasal dari 110 kasus individu, dengan jumlah *slice* per kasus bervariasi antara 80 hingga 200 citra. Sampel untuk masing-masing kelas dari *dataset* dapat dilihat pada Tabel 3 berikut.

Tabel 3. Kelas *Dataset IQ-QTH/NCCD*.

Kategori / Jumlah Data	Gambar
Normal / 416	
Benign / 120	
Malignant / 561	

3.3 Alat Pendukung

3.3.1 Perangkat Keras (*Hardware*)

Perangkat keras yang digunakan pada penelitian ini adalah sebuah laptop dengan spesifikasi sebagai berikut:

- Nama *Device* : Asus

- b. *Processor* : 12th Gen Intel(R) Core(TM) i5-1235U 1.30GHz
- c. *Installed RAM* : 16.0 GB
- d. *System Type* : 64-bit operating system, x64-based processor

3.3.2 Perangkat Lunak (*Software*)

Perangkat lunak yang digunakan dalam penelitian ini antara lain:

- a. Sistem Operasi : Windows 11
Sistem operasi yang digunakan pada laptop adalah Windows 11.
- b. Bahasa Pemrograman Python
Python adalah bahasa pemrograman tingkat tinggi yang dikenal karena sintaksisnya yang sederhana, mudah dipahami, dan fleksibel. Dalam Python, terdapat beberapa aturan yang harus dipenuhi. Hal ini untuk mengantisipasi terjadinya *error* atau masalah pada program yang dibuat. Python sering digunakan dalam berbagai bidang, seperti pengembangan perangkat lunak, analisis data, kecerdasan buatan, dan lainnya (Ks et al., 2021).
- c. Google Colab
Google Colab (*Google Colaboratory*) adalah platform berbasis *cloud* yang memungkinkan pengguna menulis dan menjalankan kode Python langsung dari *browser* tanpa memerlukan instalasi perangkat lunak tambahan. *Platform* ini mendukung kolaborasi *real-time*, integrasi dengan Google Drive, serta memberikan akses gratis ke GPU dan TPU untuk pengolahan data dan komputasi berat, menjadikannya pilihan populer bagi pengembang, peneliti, dan pelajar (Nazar, 2024).
- d. Google Drive
Google Drive adalah layanan penyimpanan berbasis awan yang memungkinkan pengguna menyimpan berbagai jenis *file*, seperti gambar, video, dokumen, *spreadsheet*, dan presentasi. Layanan ini pertama kali diperkenalkan oleh Google Inc. pada April 2012 dan menyediakan kapasitas penyimpanan gratis sebesar 15 GB. Salah satu keunggulan utama

Google Drive adalah kemudahan aksesnya yang dapat dilakukan kapan saja dan dari perangkat apa saja selama terhubung ke internet.

Google Drive juga mendukung fitur berbagi *file* dengan pengaturan akses yang fleksibel, seperti memberi izin untuk mengedit, berkomentar, atau hanya melihat *file*. Selain itu, *platform* ini dilengkapi teknologi pengenalan karakter optik (OCR) yang memungkinkan pencarian teks dalam gambar atau dokumen PDF. Dengan fitur-fitur tersebut, Google Drive menjadi alat yang andal untuk menyimpan, mengelola, dan berbagi data secara efisien (Safitri & Nasution, 2023).

e. Jupyter Notebook

Jupyter Notebook adalah aplikasi berbasis *website* yang bisa digunakan untuk menulis dan menjalankan kode Python. Selain itu, Jupyter Notebook juga bisa digunakan untuk membuat catatan, menampilkan grafik, dan melakukan visualisasi data secara langsung di *browser* (Nurdiana & Algifari, 2020).

f. TensorFlow

TensorFlow merupakan salah satu *library* untuk *machine learning* (ML) dan *deep learning* (DL) yang dikembangkan oleh Google. TensorFlow menawarkan berbagai macam tugas ML/DL seperti regresi, klasifikasi, dan lainnya. *Library* ini dirancang untuk berjalan pada CPU, GPU, atau bahkan pada sistem operasi *mobile*. TensorFlow juga sangat fleksibel, memungkinkan banyak modifikasi sesuai kebutuhan, tetapi penggunaannya lebih cocok untuk mereka yang sudah terbiasa karena kompleksitasnya (Raihan et al., 2021).

g. Keras

Keras adalah *framework* berbasis Python yang dirancang untuk mempermudah pengembangan model jaringan saraf tiruan, seperti klasifikasi, pengenalan citra, dan tugas *deep learning* lainnya. *Framework* ini menggunakan TensorFlow atau Theano sebagai *backend*, mendukung penggunaan di CPU dan GPU, serta memungkinkan kombinasi berbagai jenis jaringan saraf untuk membangun model yang kompleks. Keras dikenal karena kemudahan penggunaannya, sehingga cocok untuk

eksplorasi dan pembelajaran oleh pengguna dari berbagai tingkat keahlian (Ihsan, 2021).

h. Numpy

NumPy adalah pustaka Python yang dirancang untuk mendukung operasi array dan matriks besar, serta menyediakan fungsi matematika tingkat tinggi untuk memanipulasi array tersebut. NumPy awalnya dikembangkan dari pustaka *Numeric* oleh Jim Hugunin, kemudian diperluas oleh Travis Oliphant pada tahun 2005 dengan memasukkan fitur-fitur dari pustaka *Numarray* (Julianto, 2022).

i. Pandas

Pandas adalah sebuah *library* Python yang bersifat sumber terbuka (*open source*) dan digunakan untuk analisis data. *Library* ini sangat dibutuhkan oleh *programmer* Python untuk mempelajari dan menganalisis data, termasuk dalam analisis statistik dan pengambilan keputusan. Dalam Pandas terdapat dua tipe data utama, yaitu *Series* (satu kolom data) dan *DataFrame* (data yang diatur dalam baris dan kolom) (Taruna & Budi, 2022).

j. Matplotlib

Matplotlib adalah sebuah *library* Python yang digunakan untuk membuat visualisasi data agar lebih interaktif, mudah dibaca, dan mudah dianalisis. *Library* ini mengubah data tabel yang kaku menjadi grafik, diagram, atau bentuk visual lainnya yang mampu memperlihatkan perubahan dan perbedaan data dengan lebih jelas. Hasil visualisasi menggunakan Matplotlib dapat meningkatkan daya tarik data analisis dengan tampilan yang lebih cantik dan informatif (Tamam et al., 2022).

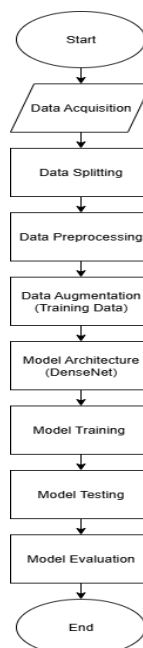
k. Scikit-learn

Scikit-learn, atau yang sering disebut *sklearn* adalah *library open source* yang dirancang untuk mendukung pengembangan *machine learning* dengan bahasa pemrograman Python. *Library* ini menyediakan berbagai algoritma yang sering digunakan dalam *machine learning*, seperti klasifikasi, regresi, dan *clustering*, termasuk algoritma k-means yang populer untuk analisis data dan pengelompokan. Dengan fitur yang

lengkap dan kemudahan penggunaannya, Scikit-learn telah menjadi salah satu *library* terkemuka dan paling banyak digunakan di GitHub, baik oleh peneliti, akademisi, maupun praktisi industri (Rakhmawati et al., 2020).

3.4 Alur Kerja Penelitian

Alur kerja penelitian mencakup serangkaian langkah sistematis yang dirancang untuk mencapai tujuan penelitian. Tahapan-tahapan alur kerja penelitian dijelaskan secara rinci pada Gambar 9.



Gambar 9. Alur Kerja Penelitian.

3.4.1 Data Acquisition

Data yang digunakan dalam penelitian ini diperoleh dari *dataset* publik *IQ-QTH/NCCD*, yang berisi citra *CT scan* paru-paru dengan label klasifikasi ke dalam tiga kondisi medis, yaitu normal, *benign*, dan *malignant*. Setiap citra tersimpan dalam format digital sebagai matriks piksel dan akan diproses lebih lanjut untuk keperluan pelatihan dan pengujian model klasifikasi berbasis *deep learning*.

3.4.2 Data Splitting

Dataset dibagi menjadi tiga *subset* utama secara stratifikasi untuk memastikan evaluasi model yang optimal serta menjaga proporsi kelas pada setiap *subset*. Pembagian ini dilakukan sebagai langkah awal setelah akuisisi data dan sebelum tahap prapemrosesan lebih lanjut, guna mencegah kebocoran data (*data leakage*) dan menjaga integritas proses validasi model. Adapun rincian pembagian data adalah sebagai berikut:

- a. *Training Set* (70%): Digunakan untuk melatih model agar memahami pola-pola dalam data.
- b. *Validation Set* (15%): Digunakan untuk mengevaluasi performa model selama pelatihan guna menghindari *overfitting*.
- c. *Test Set* (15%): Digunakan untuk mengukur performa akhir model setelah pelatihan selesai.

3.4.3 Data Preprocessing

Preprocessing dilakukan untuk menyiapkan data sebelum digunakan dalam pelatihan model. Tahapan ini mencakup beberapa proses penting yang diterapkan pada citra dalam setiap *subset* data (latih, validasi, dan uji) setelah proses pembagian data, sebagai berikut:

- a. *Data Normalization*

Nilai piksel citra yang semula berada dalam rentang 0 hingga 255 dinormalisasi ke dalam rentang yang sesuai dengan kebutuhan model DenseNet, yaitu melalui fungsi *preprocess_input* yang umumnya menskalakan ke rentang $[-1, 1]$. Parameter normalisasi dihitung dari data latih dan kemudian diterapkan secara konsisten pada seluruh *subset* data (latih, validasi, dan uji). Langkah ini bertujuan untuk meningkatkan stabilitas numerik dan mempercepat proses konvergensi saat pelatihan.

- b. *Resizing Image*

Seluruh citra diubah ukurannya menjadi 224×224 piksel agar sesuai dengan dimensi masukan standar pada arsitektur model DenseNet.

c. *One-Hot Encoding*

Label kelas dikonversi ke dalam format vektor biner menggunakan teknik *one-hot encoding*. Representasi ini memungkinkan model memahami bahwa setiap kelas merupakan kategori yang berbeda dan independen dalam klasifikasi multikelas.

d. *Data Augmentation*

Setelah dilakukan pembagian *dataset* secara stratifikasi 70:15:15, jumlah data asli yang masuk ke dalam *training set* adalah sebanyak 768 gambar, yang terdiri dari 393 gambar kelas *Malignant*, 84 gambar kelas *Benign*, dan 291 gambar kelas Normal. Untuk mengatasi ketidakseimbangan kelas, dilakukan proses augmentasi pada data latih. Augmentasi dilakukan terhadap seluruh citra kelas *Benign* sehingga menghasilkan 420 gambar baru, dan sebagian citra kelas Normal yang menghasilkan 145 gambar baru. Kelas *Malignant* tidak mengalami augmentasi karena jumlah datanya telah mencukupi. Dengan demikian, setelah augmentasi, jumlah citra pada *training set* meningkat menjadi 1.433 gambar, yang terdiri dari 768 citra asli dan 565 citra hasil augmentasi. Jumlah data pada *validation set* dan *test set* masing-masing tetap sebanyak 164 dan 165 gambar, tanpa dilakukan augmentasi.

3.4.4 Model *Architecture* (DenseNet)

Model yang digunakan dalam penelitian ini adalah DenseNet (*Densely Connected Convolutional Networks*).

a. Pemilihan Model DenseNet

Penelitian ini menggunakan tiga varian arsitektur DenseNet, yaitu DenseNet-121, DenseNet-169, dan DenseNet-201. DenseNet dipilih karena kemampuannya dalam meningkatkan aliran informasi dan penggunaan ulang fitur (*feature reuse*) melalui koneksi padat antar-lapisan. Karakteristik ini berkontribusi terhadap kinerja klasifikasi yang lebih baik dengan jumlah parameter yang relatif efisien. Seluruh varian model akan diuji dalam proses klasifikasi kanker paru-paru, dan model dengan

performa terbaik berdasarkan akurasi serta metrik evaluasi lainnya akan dipilih untuk dianalisis lebih lanjut.

b. Modifikasi Model

Setiap model DenseNet diinisialisasi dengan bobot prelatih dari *dataset* ImageNet (*transfer learning*) guna mempercepat konvergensi dan meningkatkan performa awal. Model kemudian dimodifikasi pada bagian akhir agar sesuai dengan jumlah kelas dalam *dataset*, yaitu tiga kelas yang terdiri dari normal, *benign*, dan *malignant*. Lapisan atas model asli DenseNet dengan *include_top=False* dihapus, lalu ditambahkan struktur *classifier head* baru yang terdiri atas lapisan *GlobalAveragePooling2D*, *BatchNormalization*, *Dropout* (dengan tingkat *dropout* 0,5), serta dua lapisan Dense masing-masing dengan fungsi aktivasi ReLU dan *softmax* untuk klasifikasi multikelas. Sebanyak 30 lapisan terakhir dari model dasar DenseNet dibuka (*unfrozen*) dan dilatih ulang (*fine-tuned*) bersama dengan lapisan *classifier head* yang baru.

3.4.5 Model Training

Setelah arsitektur model ditentukan, tahap pelatihan dilakukan dengan parameter berikut:

- a. *Optimizer*: Model dilatih menggunakan algoritma *Stochastic Gradient Descent* (SGD) dengan *learning rate* awal sebesar 0,001 dan *momentum* sebesar 0,9.
- b. *Loss Function*: *Loss Function* yang digunakan adalah *Categorical Crossentropy* untuk mengukur selisih antara prediksi dan label asli dalam format *one-hot encoding*.
- c. Metrik Evaluasi: Akurasi, *Precision*, *Recall*, dan *F1-Score* digunakan untuk mengukur performa model.
- d. *Callback* : Beberapa *callback* yang digunakan meliputi *EarlyStopping* untuk menghentikan pelatihan saat *loss* validasi tidak membaik selama 10 *epoch*, *ReduceLROnPlateau* untuk menurunkan *learning rate* ketika *loss*

stagnan selama 5 *epoch*, serta *ModelCheckpoint* untuk menyimpan bobot model terbaik berdasarkan performa validasi.

3.4.6 Model Testing

Setelah pelatihan selesai, model diuji menggunakan data dari *test set*. Citra-citra pada *test set* diberikan kepada model untuk diprediksi labelnya, dan hasil prediksi dibandingkan dengan label asli untuk mengukur performa model terhadap data yang belum pernah dilihat.

3.4.7 Model Evaluation

Evaluasi model dilakukan menggunakan berbagai metrik untuk menilai performa klasifikasi.

- a. *Confusion Matrix*: Menyajikan distribusi antara prediksi benar dan salah untuk setiap kelas, sehingga dapat memberikan gambaran mengenai jenis kesalahan klasifikasi yang dilakukan model.
- b. *Accuracy*: Mengukur persentase prediksi yang benar dari seluruh data.
- c. *Precision, Recall, dan F1-Score*: Metrik ini digunakan untuk mengevaluasi performa klasifikasi pada masing-masing kelas, terutama dalam menangani data yang tidak seimbang.

V. SIMPULAN DAN SARAN

5.1 Simpulan

Penelitian ini bertujuan untuk mengevaluasi performa arsitektur DenseNet dalam mengklasifikasikan citra *CT scan* paru-paru ke dalam tiga kategori: normal, *benign*, dan *malignant*. Tiga varian DenseNet yang diuji dalam penelitian ini adalah DenseNet-121, DenseNet-169, dan DenseNet-201. Berdasarkan hasil pelatihan dan evaluasi terhadap *dataset IQ-QTH/NCCD*, diperoleh beberapa simpulan sebagai berikut:

1. Ketiga varian DenseNet menunjukkan performa klasifikasi yang tinggi dengan akurasi di atas 90% pada data uji. Hal ini menunjukkan bahwa arsitektur DenseNet efektif dalam mengenali pola visual pada citra *CT scan* paru-paru, meskipun tantangan tetap ada pada kelas dengan kemiripan visual tinggi.
2. Di antara ketiga varian, DenseNet-201 memberikan performa terbaik secara keseluruhan dengan akurasi sebesar 91% dan *macro F1-score* sebesar 0.81. Model ini juga mencatatkan jumlah kesalahan klasifikasi paling sedikit, khususnya pada kelas *malignant*, yang merupakan kelas paling krusial dalam konteks deteksi kanker paru-paru.
3. DenseNet-121, sebagai varian paling ringan secara arsitektur, mampu mencapai akurasi tertinggi sebesar 92% dengan *macro F1-score* sebesar 0.80, sekaligus menunjukkan proses pelatihan yang stabil dan efisien secara komputasi. Hal ini menjadikannya alternatif yang sangat layak untuk sistem yang memiliki keterbatasan sumber daya.
4. DenseNet-169 menunjukkan akurasi sebesar 91% dan *macro F1-score* yang setara dengan DenseNet-121, namun tidak memberikan peningkatan signifikan dibanding model yang lebih ringan. Hal ini mengindikasikan bahwa

peningkatan kedalaman arsitektur tidak selalu menjamin peningkatan performa tanpa konfigurasi dan data yang optimal.

5. Hasil analisis kesalahan klasifikasi menunjukkan bahwa sebagian besar kesalahan terjadi antara kelas normal dan *benign*, yang secara visual memang sulit dibedakan. Kesalahan paling kritis berupa *false negative* pada kelas *malignant* paling sedikit terjadi pada DenseNet-201, yang memperkuat posisi model ini sebagai yang paling sensitif terhadap deteksi kanker ganas.

5.2 Saran

Meskipun arsitektur DenseNet telah menunjukkan performa yang baik dalam klasifikasi citra *CT scan* paru-paru, masih terdapat ruang pengembangan, khususnya dalam meningkatkan ketelitian terhadap kelas yang rawan tertukar dan mengurangi kesalahan klasifikasi kritis seperti *false negative* pada kelas *malignant*. Adapun saran untuk penelitian selanjutnya antara lain:

1. Perlu dilakukan perluasan *dataset*, baik dari segi jumlah maupun variasi jenis citra, agar model dapat mengenali fitur visual dengan lebih umum dan mengurangi risiko *overfitting*.
2. Penerapan *preprocessing* lanjutan, seperti segmentasi paru-paru atau peningkatan kontras, dapat membantu model untuk fokus pada area yang lebih relevan, sehingga mengurangi ambiguitas visual antar kelas.
3. Penggunaan visualisasi interpretatif seperti Grad-CAM dapat membantu mengevaluasi area yang menjadi perhatian model saat membuat keputusan, sekaligus memberikan pemahaman yang lebih baik terhadap pola kesalahan klasifikasi.
4. Augmentasi tambahan khusus untuk kelas rawan kesalahan, seperti *benign* dan normal, disarankan guna memperkuat pembelajaran model terhadap pola-pola yang secara visual mirip namun semantik berbeda.
5. Eksplorasi terhadap pendekatan lanjutan seperti *attention mechanism*, *fine-tuning* yang lebih dalam, serta teknik *ensemble* model, dapat meningkatkan sensitivitas dan akurasi, terutama dalam kasus citra medis yang kompleks.

6. Penelitian selanjutnya juga diharapkan dapat melakukan analisis kesalahan secara kuantitatif dan visual yang lebih sistematis, agar pola kelemahan model dapat dipetakan dengan lebih akurat dan menjadi dasar untuk pengembangan arsitektur yang lebih optimal.

DAFTAR PUSTAKA

- Agustina, A., Yanto, F., Budianita, E., Iskandar, I., & Syafria, F. (2024). Klasifikasi Penyakit Tanaman Daun Padi Menggunakan Metode CNN Arsitektur DenseNet-121 dan Augmentasi Data. *Journal Of Information Systems And Informatics Engineering*, 8(1), 124–134. <https://doi.org/10.35145/joisie.v8i1.4256>
- Alshammari, A. (2023). DenseNet_ HybWWoA: A DenseNet-Based Brain Metastasis Classification with a Hybrid Metaheuristic Feature Selection Strategy. *Biomedicines*, 11(5). <https://doi.org/10.3390/biomedicines11051354>
- Al-Yasriy, H. F., Al-Husieny, M. S., Mohsen, F. Y., Khalil, E. A., & Hassan, Z. S. (2020). Diagnosis of Lung Cancer Based on CT Scans Using CNN. *IOP Conference Series: Materials Science and Engineering*, 928(2). <https://doi.org/10.1088/1757-899X/928/2/022035>
- Alzubaidi, L., Zhang, J., Humaidi, A. J., Al-Dujaili, A., Duan, Y., Al-Shamma, O., Santamaría, J., Fadhel, M. A., Al-Amidie, M., & Farhan, L. (2021). Review of deep learning: concepts, CNN architectures, challenges, applications, future directions. *Journal of Big Data*, 8(1). <https://doi.org/10.1186/s40537-021-00444-8>
- Ashraf, K., Iandola, F. N., Han, S., Moskewicz, M. W., Dally, W. J., & Keutzer, K. (2016). SqueezeNet: AlexNet-level accuracy with 50x fewer parameters and <0.5MB model size. *Under Review as a Conference Paper at ICLR 2017*. <https://doi.org/10.48550/arXiv.1602.07360>

- Bhatt, D., Patel, C., Talsania, H., Patel, J., Vaghela, R., Pandya, S., Modi, K., & Ghayvat, H. (2021). Cnn variants for computer vision: History, architecture, application, challenges and future scope. *Electronics (Switzerland)*, 10(20). <https://doi.org/10.3390/electronics10202470>
- Bundea, M., & Danciu, G. M. (2024). Pneumonia Image Classification Using DenseNet Architecture. *Information (Switzerland)*, 15(10). <https://doi.org/10.3390/info15100611>
- Dharma Adhinata, F., Putra Rakhmadani, D., Wibowo, M., & Jayadi, A. (2021). A Deep Learning Using DenseNet201. *Jurnal Informatika*, 9(1), 115–121.
- Dong, S., Wang, P., & Abbas, K. (2021). A survey on deep learning and its applications. In *Computer Science Review* (Vol. 40). Elsevier Ireland Ltd. <https://doi.org/10.1016/j.cosrev.2021.100379>
- Faizin, A., Arsanto, A. T., Lutfi, M., & Musa, A. R. (2022). Deep Pre-Trained Model Menggunakan Arsitektur DenseNet untuk Identifikasi Penyakit Daun Padi. *Jurnal Mahasiswa Teknik Informatika*, 6(2).
- Gayap, H. T., & Akhloufi, M. A. (2024). Deep Machine Learning for Medical Diagnosis, Application to Lung Cancer Detection: A Review. In *BioMedInformatics* (Vol. 4, Issue 1, pp. 236–284). Multidisciplinary Digital Publishing Institute (MDPI). <https://doi.org/10.3390/biomedinformatics4010015>
- Hanin, M. A., Patmasari, R., Yunendah, R., & Fu'adah, N. (2021). Sistem Klasifikasi Penyakit Kulit Menggunakan Convolutional Neural Network (CNN) Skin Disease Classification System Using Convolutional Neural Network (CNN). *E-Proceeding of Engineering*.
- Huang, G., Liu, Z., Van Der Maaten, L., & Weinberger, K. Q. (2017). Densely connected convolutional networks. *Proceedings - 30th IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017, 2017-January*, 2261–2269. <https://doi.org/10.1109/CVPR.2017.243>

- Ihsan, C. N. (2021). Klasifikasi Data Radar Menggunakan Algoritma Convolutional Neural Network (CNN). *Journal of Computer and Information Technology*, 4, 115–121.
- Joseph, J., & Rotty, L. W. A. (2020). Kanker Paru: Laporan Kasus. *Medical Scope Journal*, 2(1), 17–25. <https://doi.org/10.35790/msj.2.1.2020.31108>
- Julianto. (2022). Analisis Investasi dalam Memprediksi Pergerakan Harga Bitcoin dengan Menggunakan Recurrent Neural Network pada Platform Indodax. *Jurnal Ilmiah Rekayasa Dan Manajemen Sistem Informasi*, 8(2), 136–147.
- Juniati, N., Bahri, S., Desimarlina, Y., Robbia, A. Z., Jariah, A., & Fuadi, H. (2021). Comparison of Lung Capacity in Communities in Low Land and High Land. *Jurnal Biologi Tropis*, 21(1), 95–102. <https://doi.org/10.29303/jbt.v21i1.2359>
- Kareem, H. F., AL-Husieny, M. S., Mohsen, F. Y., Khalil, E. A., & Hassan, Z. S. (2021). Evaluation of SVM performance in the detection of lung cancer in marked CT scan dataset. *Indonesian Journal of Electrical Engineering and Computer Science*, 21(3), 1731–1738. <https://doi.org/10.11591/ijeecs.v21.i3.pp1731-1738>
- Ks, I. G. H. W., Sukerta Wijaya, I. W., Agung Pramana S.B, I. D. M., & Aditya, I. K. G. R. (2021). Program Menghitung Banyak Bata pada Ruangan Menggunakan Bahasa Python. *TIERS Information Technology Journal*, 2(1).
- Memon, N., Parikh, H., Patel, S. B., Patel, D., & Patel, V. D. (2021). Automatic land cover classification of multi-resolution dualpol data using convolutional neural network (CNN). *Remote Sensing Applications: Society and Environment*, 22. <https://doi.org/10.1016/j.rsase.2021.100491>
- Nazar, R. (2024). Implementasi Pemrograman Python Menggunakan Google Collab. *Jurnal Informatika Dan Komputer*, 15, 50–56.
- Nugroho, B., Sari Utami, A., Karini Ratnaningsih, D., & Abdurakhman Handis, L. (2023). Perbandingan Gambaran Ct Scan Paru Perokok dan Non Perokok Pasien Kanker Paru. *Jurnal Kesehatan*, 2. <https://rsupsoeradji.id>

- Nurdiana, N., & Algifari, A. (2020). Studi Komparasi Algoritma ID3 dan Algoritma Naive Bayes untuk Klasifikasi Penyakit Diabetes Mellitus. *Infotech Journal*, 6(2).
- Pardede, J., & Putra, D. A. L. (2020). Implementasi DenseNet Untuk Mengidentifikasi Kanker Kulit Melanoma. *Jurnal Teknik Informatika Dan Sistem Informasi*, 6(3). <https://doi.org/10.28932/jutisi.v6i3.2814>
- Prastowo, E. Y. (2021). Pengenalan Jenis Kayu Berdasarkan Citra Makroskopik Menggunakan Metode Convolutional Neural Network. *Jurnal Teknik Informatika Dan Sistem Informasi*, 7(2). <https://doi.org/10.28932/jutisi.v7i2.3706>
- Putri, R. A., Satyawan, A. S., Prihantono, J. A., Linggi, R. S., Gusti, I., Putri, A., Paramita, S., Kadek, N., Iswarawati, E., Akbar, F., Utomo, A., & Korespondensi, P. (2024). Model Deep Learning untuk Klasifikasi Objek pada Gambar Fisheye. *Jurnal Teknologi Informasi Dan Ilmu Komputer (JTIK)*, 11(3), 519–528. <https://doi.org/10.25126/jtiik.2024118047>
- Raihan, M., Allaam, R., & Wibowo, A. T. (2021). Klasifikasi Genus Tanaman Anggrek Menggunakan Metode Convolutional Neural Network (CNN). *E-Proceeding of Engineering*, 8(2), 1153.
- Rakhmawati, N. A., Aletha, D. B., Widiastuti, L. P. G., Laka Kaki, A. S., & Aminullah, A. T. (2020). Klasterisasi Peraturan Daerah di Seluruh Wilayah Jawa dengan Menggunakan Algoritma K-means. *Jurnal Sisfokom (Sistem Informasi Dan Komputer)*, 9(2), 242–246. <https://doi.org/10.32736/sisfokom.v9i2.816>
- Ratna, S. (2020). Pengolahan Citra Digital dan Histogram dengan Python dan Text Editor PyCharm. *Technologia*, 11(3).
- Safitri, K., & Nasution, P. (2023). Analisis Penggunaan Aplikasi Google Drive Sebagai Media Penyimpanan Data. *Jurnal Sains Dan Teknologi (JSIT)*, 3(3), 220–223.

- Sait, A. R. W. (2023). Lung Cancer Detection Model Using Deep Learning Technique. *Applied Sciences (Switzerland)*, 13(22). <https://doi.org/10.3390/app132212510>
- Sankar, V., Kothai, R., Vanisri, N., S, A., & Anandharaj, G. (2023). Lung cancer: A review. *International Journal of Health Sciences and Research*, 13(10), 307–315. <https://doi.org/10.52403/ijhsr.20231042>
- Sembiring, Y. E., Is Effendi, W., Jeswant Dillon, J., Soebroto, H., Jiwangga Suta Winarno, D., Puruhito, P., Gusti Agung Made Adnyanya Putra Putra, I., Nego Okthara Sebayang, A., & Pramesthi Wisnu Bowo Negoro, S. (2023). Lung Cancer: A Literature Review. *Jurnal Respirasi*, 09(03), 246–251. <https://doi.org/10.20473/jr.v9-I.1.2023.246-251>
- Sharifani, K., & Amini, M. (2023). Machine Learning and Deep Learning: A Review of Methods and Applications. *World Information Technology and Engineering Journal*, 10(7). <https://www.researchgate.net/publication/371011515>
- Simarmata, A. M., Salim, P., Waruwu, N. J., & Jessica, J. (2023). Densenet Architecture Implementation for Organic and Non-Organic Waste. *Jurnal Dan Penelitian Teknik Informatika*, 8(4), 2444–2449. <https://doi.org/10.33395/sinkron.v8i4.12765>
- Tamam, Moh. B., Anwari, & Hozairi. (2022). Visualisasi Data Penyebaran Covid 19 di Indonesia dan Malaysia. *Jurnal SimanteC*, 11(1).
- Taruna, G. K., & Budi, S. (2022). Penerapan Data Science pada Dataset Olympics. *Jurnal Strategi*, 4, 2443–2229.
- Wakili, M. A., Shehu, H. A., Sharif, M. H., Sharif, M. H. U., Umar, A., Kusetogullari, H., Ince, I. F., & Uyaver, S. (2022). Classification of Breast Cancer Histopathological Images Using DenseNet and Transfer Learning. *Computational Intelligence and Neuroscience*, 2022, 31. <https://doi.org/10.1155/2022/8904768>

Wang, L. (2022). Deep Learning Techniques to Diagnose Lung Cancer. *Cancers*, 14(22). <https://doi.org/10.3390/cancers14225569>

Wulandari, I., Yasin, H., & Widiharhi, T. (2020). Klasifikasi Citra Digital Bumbu dan Rempah dengan Algoritma Convolutional Neural Network (CNN). *JURNAL GAUSSIAN*, 9(3), 273–282. <https://ejournal3.undip.ac.id/index.php/gaussian/>

Zhou, T., Ye, X., Lu, H., Zheng, X., Qiu, S., & Liu, Y. (2022). Dense Convolutional Network and Its Application in Medical Image Analysis. *BioMed Research International*, 2022. <https://doi.org/10.1155/2022/2384830>