

**PENERAPAN MODEL *TRANSFER LEARNING*  
DENGAN *CONVOLUTIONAL NEURAL NETWORK (CNN)*  
PADA KLASIFIKASI CITRA TUMOR OTAK**

**(Skripsi)**

**Oleh**

**MUHTAROM AHKAM MAULANA**



**JURUSAN MATEMATIKA  
FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM  
UNIVERSITAS LAMPUNG  
BANDAR LAMPUNG  
2024**

## **ABSTRACT**

### **IMPLEMENTATION OF TRANSFER LEARNING MODEL WITH CONVOLUTIONAL NEURAL NETWORK (CNN) IN BRAIN TUMOR CLASSIFICATION**

**By**

**Muhtarom Ahkam Maulana**

Brain tumor is a condition when brain cells grow and develop abnormally in the brain tissue. Detection of brain tumors by doctors can be done by physical examination and manual diagnosis. Manual diagnosis has limitations, namely the possibility of misdiagnosis. The development of artificial intelligence towards computer vision has been applied in the classification of brain tumor images. This research classifies brain tumor images using deep learning, precisely the transfer learning method based on a convolutional neural network (CNN). Pre-trained model used for transfer learning are DenseNet121, InceptionResNetV2, MobileNetV2, NasNetMobile, and ResNet50V2. The dataset comprised 7020 images with four categories: glioma, meningioma, pituitary, and no-tumor obtained from Kaggle. Tests were carried out using several scenarios from the pre-trained model, which used for hyperparameters dropout rate and dense unit which already tuned. Evaluation of the model built using average accuracy, average precision, average sensitivity, and average specificity. The evaluation results showed that the best-performing model achieved an accuracy of 97.70% with a loss of 0.066. These results are illustrated in the confusion matrix, which shows that the model can classify brain tumor images well.

**Keywords:** Classification, Transfer Learning, Brain Tumor, Convolutional Neural Network.

## ABSTRAK

### PENERAPAN MODEL *TRANSFER LEARNING* DENGAN *CONVOLUTIONAL NEURAL NETWORK* (CNN) PADA KLASIFIKASI TUMOR OTAK

Oleh

MUHTAROM AHKAM MAULANA

Tumor otak adalah suatu kondisi di mana sel-sel otak tumbuh dan berkembang secara tidak normal pada jaringan otak. Pendeteksian tumor otak oleh dokter dapat dilakukan dengan pemeriksaan fisik dan diagnosis manual. Diagnosis manual memiliki keterbatasan, yaitu kemungkinan terjadinya kesalahan diagnosis. Perkembangan *artificial intelligence* terhadap *computer vision* saat ini telah diterapkan dalam klasifikasi citra tumor otak. Penelitian ini melakukan klasifikasi citra tumor otak menggunakan *deep learning*, khususnya metode *transfer learning* dengan basis *convolutional neural network* (CNN). Model *pre-trained* yang digunakan untuk *transfer learning* ialah *DenseNet121*, *InceptionResNetV2*, *MobileNetV2*, *NasNetMobile*, *ResNet50V2*. *Dataset* yang digunakan berjumlah 7020 gambar dengan 4 kategori yaitu *glioma*, *meningioma*, *pituitary*, dan *no-tumor* yang diperoleh dari *kaggle*. Pengujian dilakukan dengan menggunakan beberapa skenario dari model *pre-trained*, dengan *hyperparameter dropout rate* dan *dense unit* yang telah di-*tuning*. Evaluasi dari model yang dibangun menggunakan rata-rata akurasi, rata-rata presisi, rata-rata sensitifitas, dan rata-rata spesifisitas. Hasil evaluasi menunjukkan bahwa model dengan performa terbaik mampu mencapai akurasi sebesar 97,70% dengan loss sebesar 0.066. Hasil ini diilustrasikan dalam *confusion matrix*, yang menunjukkan bahwa model tersebut mampu mengklasifikasikan citra tumor otak dengan baik.

**Kata kunci:** Klasifikasi, *Transfer Learning*, Tumor Otak, *Convolutiunal Neural Network*.

**PENERAPAN MODEL *TRANSFER LEARNING*  
DENGAN *CONVOLUTIONAL NEURAL NETWORK (CNN)*  
PADA KLASIFIKASI CITRA TUMOR OTAK**

**Oleh**

**Muhtarom Ahkam Maulana**

**Skripsi**

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

**Pada**

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



**JURUSAN MATEMATIKA  
FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM  
UNIVERSITAS LAMPUNG  
BANDAR LAMPUNG  
2024**

**Judul Skripsi**

**: PENERAPAN MODEL TRANSFER  
LEARNING DENGAN  
CONVOLUTIONAL NEURAL  
NETWORK (CNN) PADA  
KLASIFIKASI TUMOR OTAK**

**Nama Mahasiswa**

**: Muhtarom Ahkam Maulana**

**Nomor Pokok Mahasiswa**

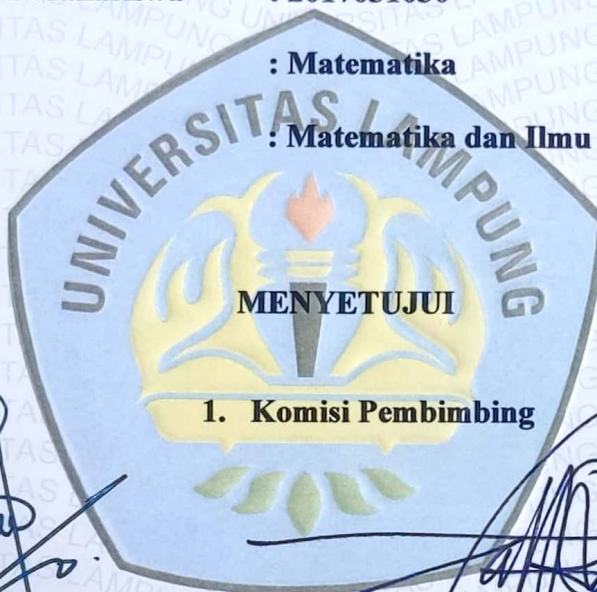
**: 2017031030**

**Jurusan**

**: Matematika**

**Fakultas**

**: Matematika dan Ilmu Pengetahuan Alam**



**1. Komisi Pembimbing**

**Dr. Dian Kurniasari, S.Si., M.Sc.**  
NIP. 196903051996032001

**Favorisen R. Lumbanraja, Ph.D.**  
NIP. 198301102008121002

**2. Ketua Jurusan Matematika**

**Dr. Aang Nuryaman, S.Si., M.Si.**  
NIP. 197403162005011001



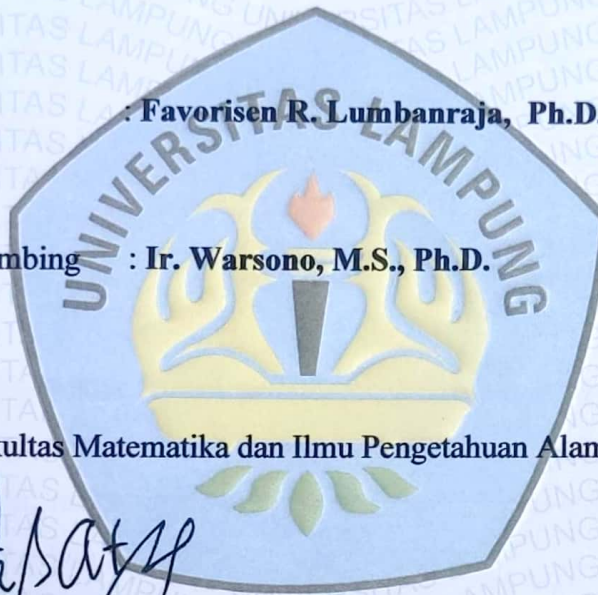
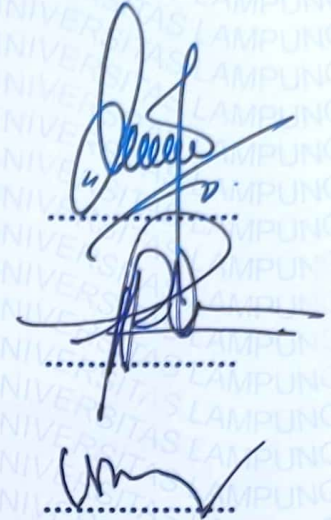
**MENGESAHKAN**

**1. Tim Penguji**

**Ketua : Dr. Dian Kurniasari, S.Si., M.Sc.**

**Sekretaris : Favorisen R. Lumbanraja, Ph.D.**

**Penguji  
Bukan Pembimbing : Ir. Warsono, M.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: 30 April 2024**

## PERNYATAAN

Yang bertanda tangan di bawah ini:

Nama : **Muhtarom Ahkam Maulana**  
Nomor Pokok Mahasiswa : **2017031030**  
Jurusan : **Matematika**  
Judul Skripsi : **PENERAPAN MODEL *TRANSFER LEARNING* DENGAN *CONVOLUTIONAL NEURAL NETWORK (CNN)* PADA KLASIFIKASI TUMOR OTAK**

Dengan ini menyatakan bahwa penelitian ini adalah pekerjaan saya sendiri dan apabila kemudian hari terbukti bahwa skripsi ini merupakan salinan atau telah dibuat orang lain, maka saya bersedia menerima sanksi sesuai dengan ketentuan akademik yang berlaku.

Bandar Lampung, 18 Mei 2024

Penulis,



**Muhtarom Ahkam Maulana**  
NPM. 2017031030

## **RIWAYAT HIDUP**

Penulis bernama Muhtarom Ahkam Maulana, lahir pada tanggal 16 Juli 2002 di Bandar Lampung. Penulis merupakan putra ketiga dari Bapak Rusman dan Ibu Uni Trisnawati.

Penulis pertama kali menempuh pendidikan di Taman Kanak-Kanak Pembina Cirebon dan Tk Darul Hikam pada tahun 2006-2007. Kemudian menempuh pendidikan Sekolah Dasar di SDN Agung pada tahun 2007-2009 dan berpindah pendidikan di Sekolah Dasar di SDN 2 Beringin Raya 2010 - 2014. Melanjutkan ke Sekolah Menengah Pertama (SMP) di SMPN 14 Bandar Lampung pada tahun 2014-2017. Selanjutnya penulis belajar pada jenjang Sekolah Menengah Atas (SMA) di SMAN 7 Bandar Lampung pada tahun 2017-2020. Pada tahun 2020 penulis terdaftar sebagai mahasiswa Program Studi S1 Jurusan Matematika dan Ilmu Pengetahuan Alam Universitas Lampung melalui jalur SNMPTN.

Pada bulan Januari 2023 penulis melaksanakan Kuliah Kerja Praktik (KP) selama 40 hari di Dinas Ketahanan Pangan, Tanaman Pangan, dan Hortikultura yang beralamat di Jl. ZA. Pagar Alam No.1, Rajabasa, Kec. Rajabasa, Kota Bandar Lampung, Lampung 35144, Indonesia. Kemudian pada bulan Juli 2023 penulis melaksanakan Kuliah Kerja Nyata (KKN) di Yayasan Apik Mandiri dan berbagai daerah lainnya selama 40 hari.



## **KATA INSPIRASI**

“Satu tahun dari sekarang kamu akan berharap telah memulai sesuatu pada hari ini.”

**(Karen Lamb)**

“The reason you are sad inside is because you know you are wasting your potential”

**(Andrew Tate)**

“Every day, it’s important to ask and answer these questions: What’s good in my life? And What needs to be done?”

**(Nathaniel Branden)**

“Tindakan menyalahkan hanya akan membuang waktu. Sebesar apapun kesalahan yang kamu timpakan ke orang lain, dan sebesar apapun kamu menyalahkannya, hal tersebut tidak akan mengubahmu.”

**(Wayne Dyer)**

“Untuk mendapatkan apa yang diinginkan, kau harus bersabar dengan apa yang kau benci.”

**(Imam Ghazali)**

“Yesterday is history, tomorrow is a mystery, but today is a gift.”

**(Master Oogway)**

## **PERSEMBAHAN**

Alhamdulillah, puji dan syukur kepada Allah SWT atas nikmat serta hidayahnya sehingga skripsi ini dapat terselesaikan dengan baik dan tepat pada waktunya.

Oleh karena itu, dengan rasa syukur dan bahagia saya persembahkan rasa terimakasih saya kepada

### **Almh. Ibu dan Bapak**

Terima kasih kepada kedua orangtua saya yang selalu mendukung segala hal yang saya inginkan. Terima kasih atas dukungan yang diberikan dalam bentuk doa, kasih sayang, dan dukungan finansial. Terimalah tanda kecil ini sebagai ungkapan kesungguhan saya untuk membalas segala pengorbanan, keikhlasan, dan jerih payah yang telah kalian lakukan.

### **Dosen Pembimbing dan Pembahas**

Terimakasih kepada dosen pembimbing dan pembahas yang telah memberikan bimbingan, motivasi, serta ilmu berharga selama proses penyusunan skripsi ini.

**Almamater Tercinta Universitas Lampung**

## SANWACANA

Puji syukur penulis haturkan atas kehadiran Allah SWT yang telah melimpahkan rahmat dan karunia-Nya kepada penulis sehingga skripsi dengan judul “***Penerapan Model Transfer Learning Dengan Convolutional Neural Network (CNN) Pada Klasifikasi Tumor Otak***” dapat terselesaikan dengan baik. Shalawat serta salam semoga senantiasa tercurah kepada Nabi Muhammad SAW beserta keluarga, sahabat, dan umatnya.

Pada tahapan penyelesaian skripsi ini dapat diselesaikan tidak terlepas dari bantuan dan kerjasama berbagai pihak. Oleh karena itu, dalam kesempatan ini penulis mengucapkan terimakasih kepada:

1. Ibu Dr. Dian Kurniasari, S.Si., M.Sc. selaku dosen pembimbing I yang senantiasa membimbing dengan sabar, memberi masukan serta saran serta mendukung penulis dalam menyelesaikan skripsi ini. Terima kasih atas bimbingan, masukan, serta dukungan yang telah diberikan selama proses penulisan skripsi ini.
2. Bapak Favorisen Rosyking Lumbanraja, S.Kom., M.Si., Ph.D selaku dosen pembimbing II yang senantiasa memberikan bimbingan, arahan, serta saran sehingga penulis dapat menyelesaikan skripsi ini.
3. Bapak Ir. Warsono, M.S. Ph.D. selaku dosen penguji yang telah memberikan kritik dan saran yang membangun sehingga skripsi ini dapat terselesaikan.
4. Bapak Prof. Drs. Mustofa, M.A., Ph.D., selaku Pembimbing Akademik.
5. Bapak Dr. Aang Nuryaman, S.Si., M.Si. selaku Ketua Jurusan Matematika Fakultas Matematika dan Ilmu Pengetahuan Alam Universitas Lampung.
6. Bapak Dr. Eng. Heri Satria, S.Si., M.Si. selaku dekan Fakultas Matematika dan Ilmu Pengetahuan Alam.

7. Seluruh dosen, staff, karyawan Jurusan Matematika Fakultas Matematika dan Ilmu Pengetahuan Alam Universitas Lampung.
8. Kepada Bapak, Almh. Ibu, dan seluruh Keluarga, saya mengungkapkan rasa terima kasih atas dukungan moral dan materiil yang tak henti-hentinya diberikan.
9. Teruntuk sahabat-sahabat terbaik, Ikram dan Rury, terima kasih atas kehadiran dan dukungan kalian. Semangat dan semangat positif yang kalian berikan membuat perjalanan penelitian ini lebih berwarna.
10. Kepada Arif yang telah memberi bantuan serta dukungan dalam mengerjakan penulisan skripsi.
11. Kepada teman-teman kuliah, Fegy, Hanafi Hilal, Nispril, Winda, Sisil, Yulian Mawar, Nisa, Nadia, Ocha, dan Gustina terima kasih atas dukungan, bantuan, dan waktu yang telah diberikan sejak awal perkuliahan hingga penelitian ini selesai.
12. Untuk teman masa SMP, Yusron, Reza, Wira, Eski, Eko, Yudha, Nanda, Nuril, dan Teman-teman dari masa SMA, seperti Aisyah, Ayu, Aem, Rahman, Syahrul, dan Zaki yang selalu memberikan dukungan dan waktu dalam menyelesaikan penulisan skripsi.
13. Orang-orang baik yang namanya tidak bisa saya sebutkan satu persatu yang telah menjadi teman terbaik penulis.
14. Teman-teman seperbimbingan, Harum, Yazid, Ziyad, Tama, Nanda, Wais, Callista, Salsa, Adel, terimakasih atas doa, motivasi, dukungan, semangat, yang selalu di berikan kepada penulis.

Penulis menyadari masih banyak kekurangan dalam penulisan skripsi ini. Oleh karena itu, penulis mengharapkan masukan serta saran untuk dijadikan pelajaran kedepannya.

Bandar Lampung, 18 Mei 2024

Penulis,

**Muhtarom Ahkam Maulana**  
NPM. 2017031030



## DAFTAR ISI

	Halaman
<b>DAFTAR TABEL</b> .....	xvi
<b>DAFTAR GAMBAR</b> .....	xvii
<b>I. PENDAHULUAN</b> .....	1
1.1 Latar Belakang dan Masalah.....	1
1.2 Rumusan Masalah .....	4
1.3 Tujuan Penelitian .....	4
1.4 Manfaat Penelitian .....	5
<b>II. TINJAUAN PUSTAKA</b> .....	6
2.1 Penelitian Terkait .....	6
2.2 Tumor Otak.....	11
2.3 Klasifikasi.....	12
2.4 Citra Digital .....	13
2.5 <i>Machine Learning</i> .....	15
2.6 <i>Deep Learning</i> .....	18
2.7 <i>Hyperparameter Tuning</i> .....	19
2.8 <i>Convolutional Neural Network</i> .....	21
2.8.1 Lapisan Konvolusi .....	22
2.8.2 Fungsi Aktivasi .....	25
2.8.3 Lapisan <i>Pooling</i> .....	28
2.8.4 Lapisan <i>Fully-Connected</i> .....	29
2.8.5 <i>Dropout Regularization</i> .....	30
2.8.6 <i>Optimizer</i> .....	32

2.8.7 Fungsi Kerugian.....	33
2.9 <i>Transfer Learning</i> .....	34
2.10 Evaluasi Model .....	40
2.10.1 Rata-rata Akurasi .....	41
2.10.2 Rata-rata Presisi.....	41
2.10.3 Rata-rata Sensitifitas .....	42
2.10.4 Rata-rata Spesifisitas .....	43
<b>III. METODOLOGI PENELITIAN</b> .....	44
3.1 Tempat dan Waktu Penelitian.....	44
3.2 Data dan Alat .....	45
3.3 Alur Kerja Penelitian.....	49
<b>IV. HASIL DAN PEMBAHASAN</b> .....	53
4.1 <i>Input Data</i> .....	53
4.2 <i>Pre-processing Data</i> .....	54
4.2.1 Penerapan <i>Grayscale</i> .....	55
4.2.2 Penerapan <i>Gaussian Blur Filter</i> .....	56
4.2.3 Penerapan <i>Binary Threshold</i> .....	57
4.2.4 Penerapan Erosi dan Dilasi .....	58
4.2.5 Pengambilan Kontur .....	59
4.2.6 Penerapan <i>Cropping</i> .....	60
4.2.7 Penerapan <i>Bone Color Map</i> .....	61
4.2.8 Penerapan <i>Resize</i> .....	61
4.3 <i>Splitting Data</i> .....	62
4.4 Augmentasi Data .....	64
4.5 Pemodelan Klasifikasi .....	66
4.5.1 <i>Hyperparameter Tuning</i> Model Klasifikasi.....	67
4.5.2 <i>Model Fitting</i> dengan hyperparameter terbaik .....	68
4.6 Pengujian Model Klasifikasi .....	69
4.6.1 Pengujian Tanpa <i>Transfer Learning</i> .....	70
4.6.2 Pengujian Dengan <i>Transfer Learning</i> .....	73

4.7 Evaluasi Model dan Perbandingan Model Klasifikasi .....	85
4.8 Perbandingan dengan Penelitian Sebelumnya .....	87
<b>V. PENUTUP .....</b>	<b>89</b>
5.1 Kesimpulan.....	89
5.2 Saran .....	90
<b>DAFTAR PUSTAKA.....</b>	<b>91</b>
<b>LAMPIRAN .....</b>	<b>95</b>

## DAFTAR TABEL

Tabel	Halaman
1. Penelitian terkait klasifikasi tumor otak.....	6
2. Data gambar penyakit tumor jenis <i>meningioma</i> .....	45
3. Data gambar penyakit tumor jenis <i>glioma</i> .....	45
4. Data gambar penyakit tumor jenis <i>pituitary</i> .....	46
5. Data gambar penyakit tumor jenis <i>no-tumor</i> .....	46
6. Jumlah gambar tiap kategori .....	53
7. Ilustrasi struktur <i>directory</i> pada dataset .....	54
8. Jumlah kategori setelah dilakukan <i>splitting dataset</i> .....	63
9. Tahapan augmentasi serta parameter yang digunakan .....	65
10. Model yang digunakan untuk klasifikasi citra tumor otak.....	66
11. Pengelompokan <i>hyperparameter</i> yang akan digunakan .....	68
12. Hasil <i>hyperparameter</i> setelah dilakukan <i>hypertuning</i> .....	68
13. Kriteria pengujian yang dipakai untuk model klasifikasi .....	69
14. Performa tiap kategori model CNN .....	72
15. Performa tiap kategori <i>model pre-trained DenseNet121</i> .....	75
16. Performa tiap kategori <i>model pre-trained InceptionResNetV2</i> .....	77
17. Performa tiap kategori <i>model pre-trained MobileNetV2</i> .....	80
18. Performa tiap kategori <i>model pre-trained NasNetMobile</i> .....	82
19. Performa tiap kategori <i>model pre-trained ResNet50V2</i> .....	84
20. Perbandingan performa dari tiap model klasifikasi .....	85
21. Perbandingan model CNN dengan model <i>transfer learning</i> terbaik .....	87
22. Perbandingan performa model dengan penelitian terdahulu .....	87



## DAFTAR GAMBAR

Gambar	Halaman
1. Representasi citra digital dalam 2 dimensi.....	15
2. Perbedaan ML konvensional dan DL.....	19
3. Contoh arsitektur CNN pada klasifikasi gambar .....	22
4. Perhitungan pada setiap langkah lapisan konvolusi .....	24
5. Fungsi aktivasi <i>sigmoid</i> .....	26
6. Fungsi aktivasi <i>tanh</i> .....	26
7. Fungsi aktivasi ReLU .....	27
8. Fungsi aktivasi <i>softmax</i> .....	28
9. Tiga tipe operasi <i>pooling</i> .....	29
10. Lapisan <i>Fully-connected</i> .....	30
11. Contoh penggunaan <i>dropout regularization</i> .....	31
12. Perbedaan <i>traditional machine learning</i> dengan <i>transfer learning</i> .....	35
13. Arsitektur <i>DenseNet121</i> .....	36
14. Arsitektur <i>InceptionResNetV2</i> .....	37
15. Arsitektur <i>MobileNetV2</i> .....	38
16. Arsitektur <i>NasNetMobile</i> .....	39
17. Arsitektur <i>ResNet50V2</i> .....	39
18. <i>Confusion Matrix</i> klasifikasi <i>multiclass</i> .....	40
19. Alur Pengerjaan Penelitian .....	49
20. Penerapan <i>grayscale</i> pada citra MRI otak .....	55
21. Penerapan <i>gaussian blur</i> pada citra MRI otak .....	56
22. Penerapan <i>thresholding</i> pada citra MRI otak .....	57
23. Penerapan <i>erode</i> dan <i>dilate</i> pada citra MRI otak .....	58
24. Pengambilan kontur pada citra MRI otak .....	59
25. Penerapan <i>cropping</i> pada citra MRI otak .....	60

26. Penerapan <i>bone color map</i> pada citra MRI otak .....	61
27. Penerapan <i>resize</i> pada citra MRI otak .....	62
28. <i>Splitting dataset</i> citra MRI otak .....	63
29. <i>Confusion matrix</i> model CNN .....	70
30. Grafik akurasi dan <i>loss</i> Model CNN .....	73
31. <i>Confusion matrix pre-trained DenseNet121</i> .....	74
32. Grafik akurasi dan <i>loss pre-trained DenseNet121</i> .....	75
33. <i>Confusion matrix pre-trained InceptionResNetV2</i> .....	76
34. Grafik akurasi dan <i>loss pre-trained InceptionResNetV2</i> .....	78
35. <i>Confusion Matrix pre-trained MobileNetV2</i> .....	79
36. Grafik akurasi dan <i>loss pre-trained MobileNetV2</i> .....	80
37. <i>Confusion matrix pre-trained NasNetMobile</i> .....	81
38. Grafik akurasi dan <i>loss pre-trained NasNetMobile</i> .....	82
39. <i>Confusion matrix pre-trained ResNet50V2</i> .....	83
40. Grafik akurasi dan <i>loss pre-trained ResNet50V2</i> .....	84
41. Ilustrasi arsitektur model yang digunakan .....	86
42. Perbandingan performa model dengan penelitian terdahulu .....	88

## I. PENDAHULUAN

### 1.1 Latar Belakang dan Masalah

Tumor otak merupakan penyakit yang ditandai dengan pertumbuhan sel yang tidak normal di dalam atau di sekitar otak secara tidak wajar dan tidak terkendali. Tumor otak dibagi menjadi dua yaitu, tumor otak primer dan sekunder. Tumor otak primer merupakan perubahan sel yang tidak normal dan tidak terkontrol yang berasal dari sel otak itu sendiri, sedangkan tumor otak sekunder merupakan tumor yang menyebar ke otak dari kanker tubuh bagian lain. (Sheard dkk., 2020). Jenis tumor otak yang paling umum adalah *glioma*, *meningioma*, dan *pituitary*.

Jumlah kasus tumor otak di dunia terus meningkat setiap tahunnya. Berdasarkan *Global Cancer Observatory* pada tahun 2020, kasus tumor otak di Indonesia menempati urutan ke-4 sebagai penyakit dengan jumlah kasus terbanyak dengan jumlah rata-rata kematian 15.310 kasus dalam lima tahun terakhir. Tumor otak tidak hanya menyerang orang dewasa, tetapi juga anak-anak. Banyak orang yang mengabaikan gejala tumor otak, sehingga diagnosis dan pengobatan menjadi terlambat.

Pendekatan anatomi citra kesehatan merupakan metode yang efektif dan non-invasif yang dapat digunakan untuk mendeteksi keberadaan tumor otak. Pendekatan ini menggunakan teknologi pencitraan untuk menghasilkan gambar dari jaringan dan organ internal tubuh seperti *Computed Tomography Scan* (CT-Scan) dan *Magnetic Resonance Imaging* (MRI) (Heranurweni dkk., 2018). Metode CT-Scan adalah jenis pendekatan anatomi citra kesehatan yang sering digunakan. Metode

CT-Scan menggunakan sinar X untuk menghasilkan profil atenuasi sinar X satu dimensi dari jaringan dan organ internal tubuh. Profil-profil atenuasi yang dihasilkan kemudian direkonstruksi menjadi gambar dua dimensi dan digabungkan menggunakan algoritma rekonstruksi tomografi untuk menghasilkan gambar tiga dimensi yang baik dari struktur internal tubuh. Metode CT-Scan lebih sesuai untuk melihat struktur tulang dan tidak efektif dalam mendeteksi jaringan lunak yang terdapat pada otak. Metode MRI adalah jenis pendekatan anatomi citra kesehatan lainnya yang sering digunakan. Metode MRI menggunakan medan magnet dan gelombang radio untuk menghasilkan gambar dari jaringan dan organ internal tubuh, gambar-gambar tersebut kemudian digabungkan untuk menghasilkan gambar 3D. Metode MRI lebih baik dalam memberikan informasi citra yang lebih dalam, sehingga metode MRI bisa memberikan gambaran informasi yang jelas antara jaringan lunak dan jaringan keras yang terdapat pada otak. Hasil dari MRI dapat memberikan informasi penting bagi dokter untuk melakukan evaluasi dan diagnosis. Metode yang telah dijelaskan di atas memiliki risiko kesalahan dan membutuhkan tahapan serta waktu yang lama. Pengembangan metode alternatif yang cepat dan akurat perlu diteliti untuk membantu dokter dalam mengambil keputusan secara efisien dan mengurangi tahapan yang lama.

Perkembangan *artificial intelligence* (AI) telah berkembang pesat, khususnya bidang *computer vision*. *Computer vision* telah banyak diterapkan pada bidang kesehatan, salah satunya adalah pengenalan pola citra pada penyakit yang berkaitan dengan hasil anatomi citra kesehatan. Pengenalan pola citra dapat dilakukan dengan menggunakan pendekatan tradisional *machine learning* (ML) atau *deep learning* (DL). Salah satu pengenalan pola citra yang akan dibahas kali ini ialah tentang klasifikasi citra tumor otak.

Penelitian klasifikasi tumor otak sudah pernah dilakukan sebelumnya. Contoh penelitian yang pernah dilakukan ialah penelitian oleh Akbar dkk (2019). Penelitian tersebut berfokus pada analisis performa model *naïve bayes* dalam deteksi otomatis citra MRI otak manusia. Kumpulan data yang digunakan berjumlah 41 citra otak yang diperoleh melalui *images scrapping*, penelitian ini berhasil



mengklasifikasikan citra otak menjadi dua kelas, normal dan tidak normal, dengan akurasi sebesar 84,17%. Proses analisis melibatkan beberapa tahap, termasuk *pre-processing*, ekstraksi fitur, dan klasifikasi.

Penelitian "Klasifikasi Tumor Otak pada Citra *Magnetic Resonance Imaging* (MRI) dengan metode *support vector machine*" yang dilakukan oleh Febrianti dkk (2020). Penelitian yang dilakukan berfokus untuk mengeksplorasi penggunaan metode *support vector machine* (SVM) dalam mengklasifikasikan citra MRI tumor otak. Kumpulan data yang digunakan berjumlah 200 gambar otak, penelitian ini melibatkan tahap *pre-processing*, ekstraksi fitur tekstur menggunakan *gray level co-occurrence matrix* (GLCM), dan klasifikasi dengan SVM berkernel *radial basis function* (RBF). Hasil menunjukkan bahwa metode SVM mampu mengklasifikasikan citra dengan akurasi memadai dengan nilai akurasi sebesar 76%, presisi sebesar 83%, spesifisitas 81%, dan *error* sebesar 24%. Meskipun berkontribusi pada pengembangan sistem klasifikasi, penelitian ini merekomendasikan penelitian lanjutan untuk meningkatkan variasi data dan mengeksplorasi metode klasifikasi alternatif.

Penelitian terdahulu yang telah dilakukan lainnya ialah tentang klasifikasi tumor otak pada citra MRI yang telah ditingkatkan kontras menggunakan model *convolutional neural network* (CNN) yang dilakukan oleh Das dkk (2019). Penelitian menggunakan kumpulan data yang terdiri dari 3064 citra dengan 3 jenis tumor otak yaitu *glioma*, *meningioma*, dan *pituitary*. Model CNN dilatih dengan citra yang telah diproses menggunakan tahapan *pre-processing*. Tahapan *pre-processing* meliputi perubahan ukuran citra, *gaussian filter*, dan pemerataan histogram. Kumpulan data yang dimiliki dibagi menjadi 3 bagian yaitu data pelatihan, data pengujian, dan data validasi. Model CNN dibangun dan dilatih menggunakan fungsi aktivasi *rectified linear unit* (ReLU), *optimizer adaptive moment estimation* (Adam), dan *dropout rate* 25% dan 40%. Pada tahap pengujian dan evaluasi, model CNN berhasil mencapai nilai rata-rata presisi sebesar 93,3%. Berdasarkan penelitian-penelitian sebelumnya, penelitian ini bertujuan untuk meningkatkan performa klasifikasi citra tumor otak dengan memanfaatkan

perkembangan terbaru dalam bidang CNN yaitu *transfer learning*. *Transfer learning* menggunakan model CNN yang telah dilatih pada kumpulan data yang besar dan kompleks sebelumnya, seperti *ImageNet*. Harapan penggunaan model yang telah terlatih dapat memanfaatkan pengetahuan yang telah terakumulasi dalam representasi fitur yang optimal dari citra, tanpa perlu melatih model dari awal dan mencari bobot secara acak. Berdasarkan uraian latar belakang di atas, maka judul penelitian ini adalah “Penerapan Model *Transfer Learning* dengan *Convolutional Neural Network* (CNN) pada Klasifikasi Citra Tumor Otak”.

## 1.2 Rumusan Masalah

Rumusan masalah dari tujuan penelitian tersebut adalah sebagai berikut:

1. Bagaimana cara membangun model *transfer learning* dengan CNN pada klasifikasi citra tumor otak?
2. Apakah model *transfer learning* dengan CNN dapat meningkatkan performa klasifikasi citra tumor otak dibandingkan dengan model lainnya?
3. Bagaimana nilai rata-rata akurasi, rata-rata presisi, rata-rata sensitifitas, dan rata-rata spesifisitas dari model *transfer learning* dengan CNN pada klasifikasi citra tumor otak?

## 1.3 Tujuan Penelitian

Tujuan dari penelitian ini adalah sebagai berikut:

1. Membangun model *transfer learning* dengan CNN pada klasifikasi citra tumor otak.
2. Mengetahui performa model yang telah dibangun dari model *transfer learning* dengan CNN dengan melihat nilai rata-rata akurasi, rata-rata presisi, rata-rata sensitifitas, dan rata-rata spesifisitas.

## 1.4 Manfaat Penelitian

Manfaat dari penelitian ini adalah sebagai berikut:

1. Dapat dijadikan referensi bagi layanan kesehatan dalam mendeteksi penyakit tumor otak dengan menggunakan model *transfer learning* dengan CNN.
2. Dapat menjadi referensi bagi peneliti selanjutnya dalam melakukan peramalan menggunakan model *transfer learning* dengan CNN dengan jenis data citra yang berbeda.

## II. TINJAUAN PUSTAKA

### 2.1 Penelitian Terkait

Penelitian yang dilakukan memanfaatkan penelitian sebelumnya sebagai sumber referensi dan perbandingan dalam mengevaluasi hasil klasifikasi. Topik penelitian yang dijadikan pembanding adalah *Classifying Brain Tumors on Magnetic Resonance Imaging by Using Convolutional Neural Networks* dan *A Robust and Novel Approach for Brain Tumor Classification Using Convolutional Neural Network*. Secara umum, penelitian ini merangkum beberapa riset yang digunakan sebagai bahan referensi, dengan fokus pada topik-topik seperti klasifikasi citra tumor otak dengan menggunakan CNN dan *transfer learning*. Berikut adalah beberapa metrik evaluasi yang digunakan untuk mengukur kinerja model klasifikasi pada Tabel 1:

- $acc_{avg}$  = Rata-rata akurasi
- $p_{avg}$  = Rata-rata presisi
- $r_{avg}$  = Rata-rata sensitifitas
- $s_{avg}$  = Rata-rata spesifisitas

Tabel 1. Penelitian terkait klasifikasi tumor otak

No	Penelitian	Data	Metode Klasifikasi	Hasil			
				$acc_{avg}$ (%)	$p_{avg}$ (%)	$r_{avg}$ (%)	$s_{avg}$ (%)
1	<i>Analisis Performa</i>	<b>Penyakit tumor Otak</b> Jumlah: 39	<i>Naïve Bayes</i>	84,17	85,03	84,48	84,47



	<i>Algoritma Naïve Bayes Pada Deteksi Otomatis Citra MRI</i> (Akbar dkk., 2019)	Normal: 20 Tidak normal: 19  Sumber: <i>Image scrapping</i>					
2	<i>Klasifikasi Tumor Otak pada Citra Magnetic Resonance Image dengan Menggunakan Metode Support Vector Machine</i> (Febrianti Ainani dkk., 2020)	<b>Penyakit tumor Otak</b> Jumlah: 200 Penyakit Tumor: 100 Normal: 100  Sumber: <i>kaggle</i>	SVM	76,00	83,00	71,00	81,00
3	<i>Brain Tumor Classification Using Convolutional Neural Network</i> (Das dkk., 2019)	<b>Penyakit Tumor Otak</b> Jumlah: 3064 <i>Glioma:</i> 1426 <i>Meningioma</i> : 708 <i>Pituitary:</i> 930  Sumber: <i>Figshare</i>	CNN	94,39	93,33	93	96,65

Berikut adalah *resume* penelitian pada Tabel 1.

a. Penelitian Pertama (Akbar dkk., 2019)

Penelitian ini bertujuan untuk menganalisis performa algoritma *naïve bayes* pada deteksi citra MRI otak manusia. Penelitian ini menggunakan kumpulan data dengan jumlah 39 citra otak yang diperoleh dengan metode *image scrapping* yang kemudian diekstraksi fiturnya. Kumpulan data terdiri atas dua kelas, yaitu normal dan tidak normal. Penelitian ini menggunakan metode *naïve bayes* untuk melakukan klasifikasi citra otak berdasarkan fitur-fitur yang telah diekstraksi. Penelitian ini melalui beberapa tahapan, yaitu *pre-processing*, ekstraksi fitur, pemeringkatan fitur, dan klasifikasi.

Pada tahap *pre-processing*, setiap citra yang diolah akan dilakukan *resize* ukurannya sama rata dengan dimensi 250 x 250 *pixel*. Setelah dilakukan proses *resize*, citra akan diekstraksi fiturnya. Proses ekstraksi fitur dari suatu citra dilakukan dengan proses kompresi agar mengurangi redundansi pada citra. Citra yang telah dikompres akan melalui proses segmentasi lokasi tumor otak dengan memecah citra menjadi dua yang berisi citra otak yang mengandung tumor dan citra yang berisi tempurung otak. Luas segmentasi citra tumor yang diambil dibandingkan dengan citra otak yang tersegmentasi, sehingga dapat dibandingkan masing-masing luasnya. Hasil dari luas citra yang telah tersegmentasi dihitung persentase luasnya, sehingga memunculkan nilai persentase luas tumor dan persentase luas tempurung otak. Pada tahap pemeringkatan fitur, kumpulan data yang telah diperoleh dari ekstraksi fitur kemudian diproses menggunakan metode *cross validasi*, model *naïve bayes*, *laplace correction*, serta *performance* untuk memunculkan hasil perhitungan dari model yang diusulkan. Pada tahap klasifikasi, kumpulan data yang telah melalui pemeringkatan fiturnya kemudian diklasifikasikan dengan algoritma *naïve bayes*. Hasil penelitian ini menunjukkan bahwa algoritma *naïve bayes* dapat mengklasifikasikan citra otak dengan akurasi sebesar 84,17%. Penelitian ini memberikan kontribusi dalam pengembangan sistem deteksi citra MRI otak manusia yang dapat membantu diagnosis penyakit tumor otak.

b. Penelitian Kedua (Febrianti Ainani dkk., 2020)

Penelitian ini membahas tentang penggunaan metode SVM untuk mengklasifikasikan citra MRI yang menunjukkan adanya tumor otak atau tidak. Penelitian ini dilatarbelakangi oleh meningkatnya kasus tumor otak di Indonesia, baik pada orang dewasa maupun anak-anak. Penelitian ini bertujuan untuk mengembangkan sistem klasifikasi tumor otak yang akurat dan efisien dengan menggunakan SVM.

Penelitian ini menggunakan kumpulan data gambar otak yang berjumlah 100 data. Kumpulan data dapat diakses pada [website https://www.kaggle.com/datasets/simeondee/brain-tumor-images-dataset](https://www.kaggle.com/datasets/simeondee/brain-tumor-images-dataset) dan disimpan dalam format .JPG dengan ukuran *pixel* yang bervariasi. Penelitian ini melakukan tiga langkah utama, yaitu *pre-processing*, ekstraksi fitur, dan klasifikasi.

Pada langkah *pre-processing*, gambar otak diubah menjadi *grayscale* terlebih dahulu, lalu disaring dengan *gaussian filter*, selanjutnya menerapkan *thresholding*, melakukan operasi morfologi, serta pengambilan kontur. Langkah *pre-processing* yang dilakukan bertujuan untuk menghilangkan *noise*, meningkatkan kontras, dan mendapatkan bentuk tumor otak yang jelas. Pada langkah ekstraksi fitur, gambar otak yang telah di-*cropping* sebelumnya, akan diekstraksi fitur teksturnya dengan menggunakan GLCM. Fitur tekstur yang digunakan adalah *contrast*, *correlation*, *energy*, *dissimilarity*, *angular second moment*, *homogeneity*, dan *entropy*. Langkah ekstraksi fitur bertujuan untuk mendapatkan informasi fitur yang terkandung dalam suatu pola pada gambar otak untuk memudahkan proses klasifikasi. Pada langkah klasifikasi, fitur-fitur yang telah diekstraksi digunakan sebagai *input* untuk SVM. Penelitian ini menggunakan tiga jenis kernel SVM, yaitu linear, RBF, dan polinomial. Hasil dari penelitian ini menunjukkan bahwa metode SVM dengan kernel RBF lebih baik daripada dengan kernel linear dan polinomial. Metode SVM dengan kernel RBF dapat mengklasifikasikan gambar otak yang mengidap penyakit tumor atau tidak dengan akurasi yang cukup baik sebesar 76%, presisi sebesar

83%, spesifisitas sebesar 81%, dan sensitifitas sebesar 71%. Penelitian ini memberikan saran untuk penelitian selanjutnya, yaitu untuk melakukan kombinasi antara metode SVM dengan model lain seperti CNN untuk mendapatkan hasil yang lebih baik.

c. Penelitian Ketiga (Das dkk., 2019)

Penelitian ini bertujuan untuk mengklasifikasikan jenis-jenis tumor otak dalam citra MRI yang telah ditingkatkan kontras dengan menggunakan model CNN yang diimplementasikan oleh peneliti. Penelitian ini menggunakan kumpulan data dari *website figshare* yang dapat diakses pada [https://figshare.com/articles/dataset/brain\\_tumor\\_dataset/1512427](https://figshare.com/articles/dataset/brain_tumor_dataset/1512427) yang berisi 3064 citra MRI otak yang terdiri atas 3 jenis tumor otak, yaitu *glioma*, *meningioma*, dan *pituitary*. Penelitian ini melalui beberapa tahapan, yaitu *pre-processing*, pembagian data, pelatihan, pengujian, dan evaluasi.

Pada tahap *pre-processing*, citra MRI diubah ukurannya menjadi 112 x 112 piksel, lalu citra disaring dengan *gaussian filter* dan ditingkatkan kontrasnya dengan menggunakan *histogram equalization*. Setelah dilakukan *pre-processing*, data dibagi menjadi 3 bagian yaitu pelatihan, pengujian, dan validasi. Kumpulan data dibagi dengan rasio 66,9%, 16,7%, dan 16,3%. Data pelatihan terdiri atas 2051 citra, data pengujian terdiri atas 500 citra, serta data validasi terdiri atas 513 citra. Pada tahap pelatihan, model CNN dilatih dengan data pelatihan yang telah melalui tahap *pre-processing*. Model CNN yang dibuat terdiri dari lapisan input berisi citra yang telah melalui tahap *pre-processing*, dilanjutkan dengan lapisan konvolusi dan fungsi aktivasi ReLU, *max-pooling*, lapisan *dropout*, lapisan *flatten* dan diakhiri dengan lapisan *dense* dengan fungsi aktivasi softmax untuk memprediksi *output*. Model CNN yang dilatih menggunakan fungsi aktivasi ReLU, *optimizer Adam*, *dropout rate* 25% dan 40%, dan fungsi aktivasi *output softmax*. Pada tahap pengujian, model CNN diuji dengan data pengujian. Pada tahap evaluasi, model-model CNN dievaluasi dengan menggunakan metrik akurasi, presisi, sensitifitas, *f1-score*, *support*, dan kurva *receiver operating characteristic (ROC)*. Hasil penelitian

ini menunjukkan bahwa model CNN dapat mengklasifikasikan tumor otak dengan presisi sebesar 88%, 94%, 98% untuk kelas *glioma*, *meningioma*, dan *pituitary*. sensitifitas sebesar 85%, 95%, 99% untuk kelas *glioma*, *meningioma*, dan *pituitary*. f1-score sebesar 87%, 94%, 99% untuk kelas *glioma*, *meningioma*, dan *pituitary*.

## 2.2 Tumor Otak

Tumor merupakan perkembangan dari sel yang tidak biasa dimana sel merupakan bangunan dasar yang menyusun jaringan dan organ pada tubuh. Pada tumor otak, sel yang tidak normal membentuk benjolan yang tumbuh di daerah sekitar otak yang dapat mengganggu fungsi otak (Sheard dkk., 2020). Pada saat tumor otak terjadi, pertumbuhan sel yang tidak diperlukan secara berlebihan menimbulkan penekanan dan kerusakan pada sel-sel lain di otak dan mengganggu fungsi otak bagian tersebut (Yueniwati, 2017). Tumor otak umumnya terdiri dari tumor otak primer dan sekunder. Tumor otak dengan kelompok primer yaitu penyakit tumor yang pertama kali tumbuh dan berkembang pada otak. Tumor ini dapat menyebar ke bagian lain dari sistem saraf, tetapi jarang tumbuh pada bagian tubuh yang lain. Sedangkan tumor otak dengan kelompok sekunder yaitu penyakit dari tumor yang bermula tumbuh pada bagian dari tubuh selain otak (Andre dkk., 2021).

Tingkatan tumor otak dibagi menjadi empat tingkat, yaitu tingkat 1 hingga tingkat 4, berdasarkan perilaku dan karakteristik tumor tersebut. Pengelompokan ini didasarkan pada faktor-faktor seperti lokasi pertumbuhan tumor, tingkat pertumbuhan, dan cara penyebarannya. Tumor otak yang tergolong dalam tingkat 1 dan 2 dianggap sebagai jenis yang tidak bersifat ganas atau jinak. Tumor otak jinak ini memiliki ciri-ciri seperti pertumbuhan yang terbatas, memiliki selubung pelindung, tidak menyebar ke area lain, dan jika diangkat melalui operasi, dapat sembuh sepenuhnya (Ghozali dan Sumarti, 2021). Di sisi lain, tumor otak yang masuk dalam tingkat 3 dan 4 biasanya memiliki potensi ganas dan sering disebut sebagai tumor otak ganas atau kanker otak. Gejala yang muncul akibat tumor otak

sangat bervariasi tergantung pada faktor-faktor seperti lokasi tumor, ukurannya, dan tingkat pertumbuhannya. Tumor otak yang tumbuh dengan lambat mungkin awalnya tidak menimbulkan gejala yang mencolok, dan gejala awalnya dapat bersifat ringan seperti sakit kepala dan kelelahan. Namun, seiring berjalannya waktu, tumor dapat memberikan tekanan pada otak atau mengganggu fungsi otak, yang dapat mengakibatkan munculnya gejala yang lebih serius seperti kejang dan sakit kepala. Penyebab utama dari sebagian besar kasus tumor otak jinak hingga saat ini masih belum diketahui secara pasti. Terdapat beberapa faktor risiko yang dapat meningkatkan kemungkinan seseorang mengembangkan tumor otak, termasuk faktor genetik dan efek samping dari prosedur radioterapi.

### 2.3 Klasifikasi

Klasifikasi adalah proses pengelompokan objek berdasarkan ciri-ciri yang dimilikinya, dan dalam konteks ini, fokus pada pengelompokan data atau konsep ke dalam kelas yang berbeda (Wibawa dkk., 2018). Tujuan utama dari klasifikasi adalah untuk menemukan model atau fungsi yang dapat menggambarkan dan membedakan kelas-kelas tersebut sehingga dapat digunakan untuk memprediksi kelas dari objek pengamatan yang belum diketahui. Metode klasifikasi memiliki peran penting dalam analisis data dan pemahaman pola di dalamnya. Terdapat dua pendekatan umum dalam klasifikasi.

1. Pendekatan pertama adalah klasifikasi yang dilakukan secara manual oleh manusia tanpa melibatkan bantuan algoritma cerdas komputer. Pendekatan ini sering digunakan ketika kriteria klasifikasi sangat subjektif atau ketika data tidak tersedia dalam jumlah besar.
2. Pendekatan kedua adalah klasifikasi yang menggunakan bantuan teknologi, khususnya algoritma ML. Algoritma ML seperti *naïve bayes*, SVM, *decision tree*, *fuzzy logic*, dan jaringan saraf tiruan digunakan untuk proses klasifikasi secara otomatis. Pendekatan ini sangat berguna dalam menghadapi era *big data*, yaitu volume data yang sangat besar memerlukan analisis yang kuat dan efisien untuk mengungkap informasi berharga dan

pengetahuan terorganisir. Penerapan algoritma ML dalam klasifikasi juga telah menjadi semakin populer dengan kemajuan teknologi AI (Sihombing dan Yuliati, 2021).

Pada konteks statistika, metode klasifikasi yang umum digunakan adalah analisis diskriminan dan regresi logistik. Namun, dalam era *big data* dan perkembangan teknologi AI, algoritma ML seperti *classification and regression trees* (CART), *random forest*, *naïve bayes*, SVM, dan banyak lainnya menjadi solusi yang lebih efektif dalam melakukan klasifikasi data.

Terdapat 2 jenis klasifikasi yang umum digunakan, yaitu klasifikasi biner dan klasifikasi multi-kelas. Klasifikasi biner merupakan klasifikasi yang memisahkan 2 buah kelas, sedangkan klasifikasi multi-kelas merupakan klasifikasi yang memisahkan lebih dari dua kelas (Tantika dan Kudus, 2022). Klasifikasi biner memisahkan data ke dalam dua kelas yang berlawanan, seperti "Ya" atau "Tidak," "Positif" atau "Negatif." Sementara itu, klasifikasi multi-kelas digunakan ketika terdapat lebih dari dua kelas yang mungkin, seperti klasifikasi jenis bunga ke dalam beberapa spesies yang berbeda.

## 2.4 Citra Digital

Citra digital merupakan sebuah konsep fundamental dalam dunia pemrosesan gambar dan pengolahan visual yang memainkan peran penting dalam berbagai aspek kehidupan. Citra digital adalah cara merepresentasikan dunia visual, seperti gambar dan video, dalam bentuk data yang dapat diolah oleh komputer. Secara teknis, citra digital dapat didefinisikan sebagai sebuah fungsi matematis,  $f(x, y)$ , yang berkaitan dengan koordinat spasial  $x$  dan  $y$  (Susanto, 2019). Citra digital memiliki dimensi yang dapat diukur, seperti  $M$  baris dan  $N$  kolom, yang menunjukkan bahwa citra memiliki resolusi dan ukuran tertentu. Namun, yang membuatnya menjadi "digital" adalah sifat diskrit dari nilai-nilai  $x$ ,  $y$ , dan  $f(x, y)$ . Nilai-nilai ini adalah bilangan berhingga yang memiliki lokasi atau koordinat

tertentu dalam citra. Setiap citra digital tersusun atas sejumlah besar elemen-elemen bilangan berhingga yang memiliki lokasi spesifik dalam citra. Setiap elemen ini sering disebut dengan sebutan yang berbeda, seperti *image element*, *picture element*, *pixel*, atau pel. Pada konteks pemrosesan gambar, istilah "*pixel*" lebih sering digunakan. Piksel adalah unit terkecil dalam citra digital yang memiliki tingkat keabuan atau warna tertentu pada titik koordinatnya. Citra digital dipetakan menjadi elemen-elemen piksel berbentuk matriks 2 dimensi. Setiap piksel-piksel tersebut memiliki angka yang mempresentasikan channel warna. Warna yang terlihat pada citra merupakan kombinasi dari tiga warna dasar, yaitu merah, hijau, dan biru atau biasa dikenal dengan sebutan *red*, *green*, *blue* (RGB). Kombinasi dari tiga warna RGB tersebut menghasilkan warna yang khas untuk piksel yang bersangkutan. Pada citra 256 warna, setiap piksel panjangnya 8 bit. Nilai RGB memiliki rentang nilai antara 0-255. Pilihan skala 256 ini didasarkan pada cara mengungkap 8 digit bilangan biner yang digunakan oleh mesin komputer. Citra digita dapat dituliskan kedalam sebuah Persamaan (2.1) sebagai berikut:

$$f(x, y) = \begin{bmatrix} f(0,0) & f(0,1) & \cdots & f(0, N - 1) \\ f(1,0) & f(1,1) & \cdots & f(1, N - 1) \\ \vdots & \vdots & \ddots & \vdots \\ f(M - 1,0) & f(M - 1,1) & \cdots & F(M - 1, N - 1) \end{bmatrix} \quad (2.1)$$

Berdasarkan rumus di atas, suatu citra  $f(x,y)$  dapat dituliskan ke dalam fungsi matematis seperti Persamaan (2.2) sampai (2.4) sebagai berikut:

$$0 \leq x \leq M - 1 \quad (2.2)$$

$$0 \leq y \leq N - 1 \quad (2.3)$$

$$0 \leq f(x, y) \leq G - 1 \quad (2.4)$$

Dimana :

$M$  = jumlah piksel baris pada *array* citra

$N$  = jumlah piksel kolom pada *array* citra

$G$  = nilai skala keabuan (*grayscale*)

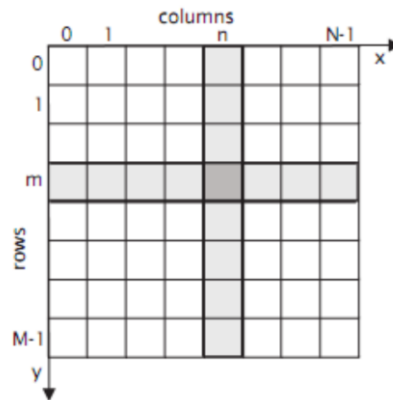
Besarnya nilai  $G$  merupakan perpangkatan dari dua seperti yang terlihat pada Persamaan (2.5) sebagai berikut:

$$G = 2^k \quad (2.5)$$

Nilai  $k$  merupakan bilangan positif. Interval  $(0, G)$  disebut dengan *grayscale*. Besarnya nilai  $G$  tergantung pada proses digitalisasi. Biasanya keabuan 0 (no)



menyatakan intensitas hitam dan 1 (satu) menyatakan intensitas putih. Pada citra 8 bit, nilai  $G$  sama dengan  $2^8 = 256$  warna (derajat keabuan). Representasi citra digital dalam 2 dimensi dapat diilustrasikan pada Gambar 1.



Gambar 1. Representasi citra digital dalam 2 dimensi (Kusumanto, 2011).

Pada konsep matriks berukuran  $M \times N$  ini, citra digital memungkinkan untuk merepresentasikan dunia visual secara terstruktur. Informasi visual, seperti gambar, dapat dipecah menjadi sejumlah besar piksel dengan nilai-nilai yang mewakili warna, kecerahan, atau intensitas di setiap titik. Berdasarkan konsep matriks di atas, komputer dapat menyimpan, mengakses, dan memanipulasi informasi visual dengan mudah.

Citra digital tidak hanya tentang representasi pasif dari gambar atau video. Citra digital membuka pintu bagi berbagai jenis analisis dan pemrosesan gambar yang dapat membantu memahami dan mengambil keputusan berdasarkan informasi visual. Konsep citra digital membentuk dasar dari berbagai aplikasi teknologi modern, seperti pengenalan wajah, kendaraan otonom, pemrosesan citra medis, dan banyak lagi.

## 2.5 Machine Learning

Istilah "*Machine Learning*" atau ML pertama kali diperkenalkan oleh seorang ilmuwan komputer bernama Arthur Samuel pada tahun 1959. Pada awalnya, istilah

ini muncul dalam konteks permainan catur, di mana komputer dapat belajar dan mengembangkan kemampuan untuk bermain catur dengan sendirinya (Joshi, 2020). Namun, seiring berjalannya waktu, konsep ML telah berkembang pesat dan mencakup berbagai bidang dalam ilmu komputer dan teknologi.

Pada dasarnya, konsep ML mengacu pada kemampuan komputer untuk belajar dari data yang ada dan menghasilkan perilaku atau keputusan tanpa perlu diprogram secara eksplisit oleh manusia. Komputer dapat "mengamati" data, mengekstraksi pola, dan membuat prediksi atau keputusan berdasarkan pola-pola yang telah ditemukan. Proses pembelajaran ini melibatkan tiga komponen utama:

1. **Data:** Data adalah bahan bakar utama dalam ML. Komputer menggunakan data sebagai "pengalaman" untuk memahami dan memodelkan fenomena tertentu. Data dapat berupa berbagai jenis informasi, seperti gambar, teks, angka, atau bahkan suara.
2. **Metrik dan Kesalahan:** Untuk memahami sejauh mana komputer telah "belajar," diperlukan suatu metrik atau ukuran kinerja. Metrik ini membantu untuk mengukur sejauh mana prediksi atau keputusan yang dihasilkan oleh komputer cocok dengan kenyataan. Kesalahan adalah perbedaan antara prediksi komputer dan kenyataan, dan algoritma ML berusaha untuk mengurangi kesalahan ini seiring berjalannya waktu.
3. **Umpan Balik (*Feedback*):** Umpan balik adalah elemen kunci dalam pembelajaran. Ketika komputer membuat kesalahan, umpan balik digunakan untuk memperbaiki model atau algoritma ML. Kesalahan digunakan sebagai "pelajaran" agar komputer dapat melakukan lebih baik di masa depan.

Pada konteks teknologi modern, ML telah menjadi landasan penting dalam pengembangan AI. Berbagai algoritma ML digunakan untuk mengatasi masalah kompleks seperti pengenalan wajah, deteksi penipuan, rekomendasi produk, dan banyak lagi. Algoritma-algoritma ML terus berkembang dan semakin canggih seiring dengan peningkatan kekuatan komputasi dan ketersediaan data yang semakin besar. Kemampuan algoritma ML untuk mengekstraksi pengetahuan dari data menjadi salah satu alat paling berharga dalam menghadapi tantangan di era *big*

*data* saat ini. Ciri dari ML adalah adanya proses pembelajaran dan pelatihan. Pelatihan merupakan proses konstruksi model sedangkan pengujian merupakan proses yang digunakan dalam menguji kinerja model pembelajaran yang telah dibangun.

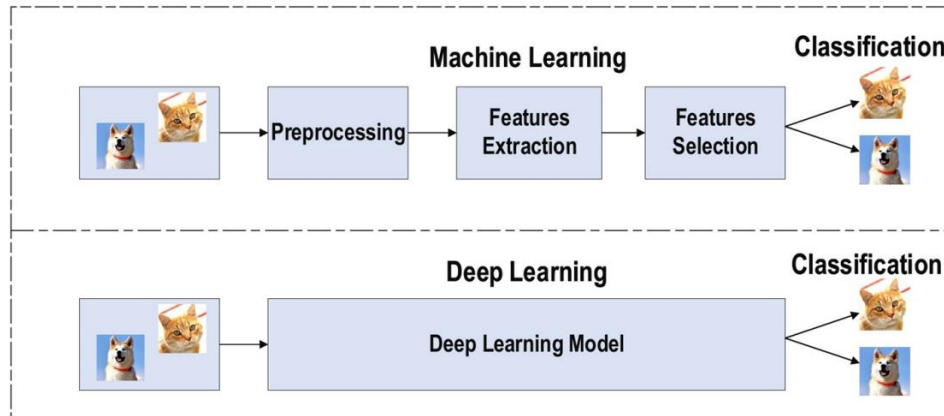
Algoritma ML memiliki 3 teknik pembelajaran, yaitu:

1. *Supervised learning* (pembelajaran terarah) adalah algoritma ML dengan model belajar dari data yang sudah memiliki label atau jawaban yang jelas. Pada *supervised learning*, hasil yang diharapkan sudah diketahui dan proses pembelajaran akan berhenti ketika algoritma telah mencapai hasil tersebut. *Supervised learning* biasanya digunakan untuk membuat prediksi dan klasifikasi.
2. *Unsupervised learning* (pembelajaran tidak terarah) adalah jenis pembelajaran mesin dengan algoritma menerima data tanpa label dan belajar untuk menemukan pola atau kelompok dalam data. Tujuan dari teknik pembelajaran tidak terarah adalah untuk mengelompokkan data yang memiliki karakteristik hampir sama dalam suatu area tertentu dan tidak memerlukan target output. *Unsupervised learning* cocok digunakan untuk berbagai macam tugas, termasuk clustering, deteksi anomali, dan reduksi dimensi.
3. *Reinforcement learning* adalah jenis pembelajaran mesin dengan model belajar melalui interaksi dengan lingkungan. Pada *reinforcement learning*, model membuat tindakan atau keputusan dalam suatu lingkungan dan menerima umpan balik berupa penghargaan atau hukuman berdasarkan tindakannya. Tujuannya adalah untuk memaksimalkan jumlah penghargaan yang diterima oleh model seiring waktu dengan menjalani serangkaian tindakan. *Reinforcement learning* sering digunakan dalam konteks pengambilan keputusan, pengendalian, dan permainan komputer.

## 2.6 Deep Learning

*Deep learning* yang merupakan subset dari ML, mengambil inspirasi dari pola-pola pengolahan informasi yang ditemukan dalam otak manusia. Berbeda dengan ML tradisional, DL tidak bergantung pada aturan yang dirancang oleh manusia (Alzubaidi dkk., 2021). Metode DL memanfaatkan kumpulan data yang sangat besar untuk memetakan input ke label-label tertentu. Metode DL dibangun menggunakan sejumlah lapisan algoritma yang dikenal sebagai jaringan saraf tiruan. Setiap lapisan memberikan interpretasi yang unik terhadap data *input*.

Pendekatan ML konvensional biasanya melibatkan tahapan-tahapan berurutan, termasuk *pre-processing*, ekstraksi fitur, seleksi fitur, pembelajaran, dan klasifikasi. Seleksi fitur memegang peran penting dalam kinerja ML, karena seleksi fitur yang bias dapat mengakibatkan diskriminasi kelas yang salah. Metode DL mengotomatisasi pembelajaran rangkaian fitur untuk berbagai tugas, menyederhanakan proses tersebut. Metode DL sangat baik dalam mencapai pembelajaran dan klasifikasi dalam satu langkah. Metode DL berbeda dari ML konvensional dalam bagaimana proses pembelajaran yang dilakukan. Metode DL tidak memerlukan aturan atau algoritma yang telah dirancang sebelumnya, DL menggunakan pendekatan yang lebih fleksibel. Metode DL diberi data yang telah dilabeli, dan kemudian secara otomatis mencari pola dan hubungan dalam data tersebut. Selama proses pembelajaran, DL dapat menyesuaikan diri dengan data baru yang belum pernah dilihat sebelumnya. Perbedaan antara ML dan DL dapat diilustrasikan pada Gambar 2.



Gambar 2. Perbedaan ML konvensional dan DL (Alzubaidi dkk., 2021).

Salah satu ciri khas pada metode DL adalah penggunaan banyak lapisan atau yang sering disebut sebagai "*deep*" dalam jaringan neural. Kemampuan tersebut sangat berguna untuk memahami dan menganalisis data yang kompleks seperti pengenalan wajah, pengenalan suara, analisis teks, pengolahan gambar, dan bahkan pemecahan masalah yang lebih kompleks seperti mobil yang dapat dikendarai secara otomatis. Salah satu algoritma DL adalah model CNN. Model CNN termasuk ke dalam kategori *deep learning* karena memiliki arsitektur jaringan saraf tiruan yang sangat dalam, dengan banyak lapisan yang secara hierarkis mengenali fitur-fitur visual dari data gambar

## 2.7 Hyperparameter Tuning

*Hyperparameter tuning* dalam konteks ML merupakan tahap yang cukup penting dalam pengembangan model. *Hyperparameter tuning* merupakan penyesuaian *hyperparameter* sebelum memulai proses pelatihan model. Parameter-parameter ini, yang dikenal sebagai *hyperparameter*, melibatkan aspek seperti *learning rate*, *epoch*, fungsi aktivasi, *batch size*, atau bahkan *dense unit* dalam jaringan saraf (Elgeldawi dkk., 2021). Pengaturan yang tepat dari *hyperparameter* memiliki dampak yang signifikan pada kualitas dan kinerja model. Pengaturan *hyperparameter* yang baik akan mengakibatkan model dapat belajar secara optimal serta menghasilkan prediksi yang akurat. Penentuan nilai yang optimal untuk

*hyperparameter* bisa menjadi tugas yang menantang. Kesalahan dalam pengaturan *hyperparameter* dapat mengakibatkan model mengalami *overfitting* dan *underfitting*. Model mengalami *overfitting* berarti bahwa model belajar terlalu baik pada data pelatihan tetapi gagal dalam menggeneralisasi hasil dengan baik pada data baru. Model mengalami *underfitting* berarti model gagal belajar dengan baik dari data pelatihan data pengujian.

Proses *hyperparameter tuning* melibatkan serangkaian eksperimen dengan tujuan untuk menemukan kombinasi *hyperparameter* terbaik agar menghasilkan kinerja model yang optimal. *Hyperparameter tuning* adalah langkah penting dalam pengembangan model ML yang bertujuan untuk memaksimalkan potensi model. Proses *tuning* yang teliti memberikan model terbaik dalam suatu tugas yang diberikan. Proses *hyperparameter tuning* dapat dilakukan dengan beberapa metode, berikut merupakan metode *hyperparameter tuning*:

1. *Grid Search*:

Metode *grid search* merupakan salah satu metode yang paling sederhana dalam mencari kombinasi *hyperparameter* terbaik dengan menentukan nilai yang mungkin untuk setiap *hyperparameter* yang ingin diatur. Semua kemungkinan kombinasi dari nilai-nilai tersebut akan dilatih pada model. Model yang telah dilatih akan dievaluasi untuk setiap kombinasi *hyperparameter*. Hasil evaluasi yang didapat akan dibandingkan, dan dipilih kombinasi *hyperparameter* yang memberikan kinerja terbaik. Keuntungan dari *grid search* adalah bahwa metode ini mencoba semua kombinasi yang mungkin, sehingga dapat yakin menemukan yang terbaik. Namun, metode *grid search* dapat menjadi sangat mahal dalam hal waktu dan sumber daya komputasi jika memiliki banyak *hyperparameter* atau nilai yang berbeda.

2. *Random Search*:

*Random search* merupakan metode yang lebih efisien daripada *grid search*. Pada *random search*, himpunan nilai yang mungkin untuk setiap *hyperparameter* tetap ditentukan, tetapi daripada mencoba semua kombinasi, metode *random search* memilih secara acak kombinasi-

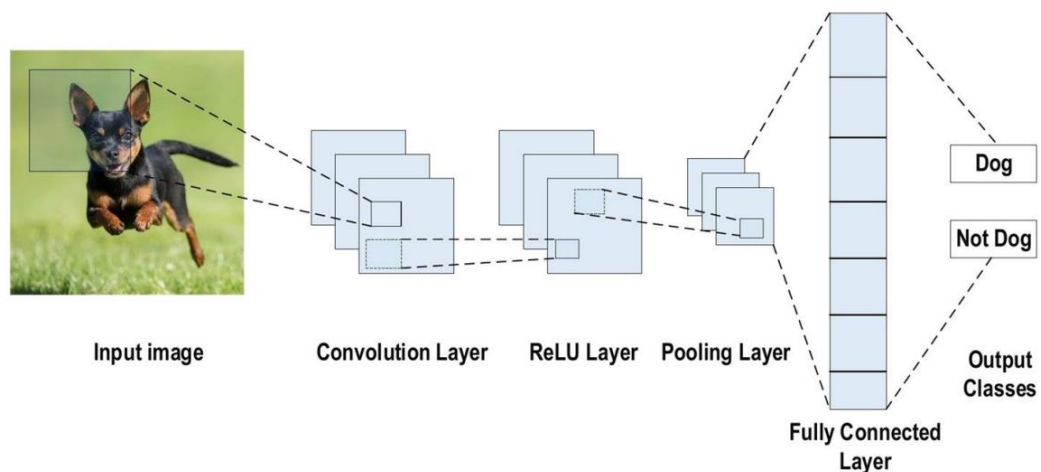
kombinasi tersebut untuk diuji. Metode *random search* menghemat waktu karena tidak perlu mencoba setiap kombinasi. Pada jumlah percobaan yang sama, *random search* dapat menghasilkan hasil yang sama baiknya dengan *grid search*.

### 3. *Bayesian Optimization*:

*Bayesian optimization* adalah metode yang memanfaatkan informasi dari percobaan sebelumnya untuk menentukan kombinasi *hyperparameter* yang optimal. Metode ini menggunakan pendekatan berbasis *bayesian* untuk mengoptimalkan *hyperparameter*. Metode *bayesian optimization* menggunakan konsep *gaussian process* untuk memprediksi kombinasi *hyperparameter* terbaik berdasarkan hasil-hasil percobaan sebelumnya.

## 2.8 *Convolutional Neural Network*

Bidang DL memiliki algoritma yang terkenal dan sering digunakan yaitu model CNN. Keunggulan utama dari model CNN dibandingkan dengan pendahulunya adalah kemampuannya untuk secara otomatis mengidentifikasi fitur-fitur yang relevan tanpa adanya pengawasan manusia. Model CNN telah banyak diterapkan dalam berbagai bidang yang berbeda, termasuk *computer vision*, pemrosesan suara, dan pengenalan wajah. Secara umum, CNN terdiri dari beberapa lapisan yang berurutan. Lapisan pertama adalah lapisan konvolusi, yang bertugas untuk mendeteksi berbagai fitur pada gambar, seperti tepi, sudut, dan tekstur. Lapisan konvolusi ini diikuti oleh lapisan *sub-sampling* atau *pooling*, yang berfungsi untuk mereduksi resolusi gambar dan mengurangi jumlah parameter yang digunakan. Model CNN biasanya memiliki lapisan sepenuhnya terhubung (*fully connected*) yang berfungsi untuk menggabungkan hasil-hasil dari lapisan sebelumnya dan menghasilkan output akhir berupa klasifikasi gambar. Contoh arsitektur CNN untuk klasifikasi gambar dapat diilustrasikan pada Gambar 3.



Gambar 3. Contoh arsitektur CNN pada klasifikasi gambar (Alzubaidi dkk., 2021).

Penerapan model CNN tidak terbatas hanya pada bidang *computer vision*, tetapi juga telah merambah ke berbagai bidang lainnya, seperti pemrosesan suara, pengenalan wajah, dan analisis teks. Keberhasilan CNN dalam memahami data gambar telah menginspirasi penggunaannya dalam berbagai aplikasi yang mencakup deteksi objek, segmentasi gambar, pengenalan pola, hingga pengembangan teknologi kendaraan otonom. Model CNN menjadi salah satu fondasi penting dalam perkembangan teknologi AI yang semakin meluas dan memiliki potensi besar dalam memecahkan berbagai masalah kompleks di berbagai bidang.

### 2.8.1 Lapisan Konvolusi

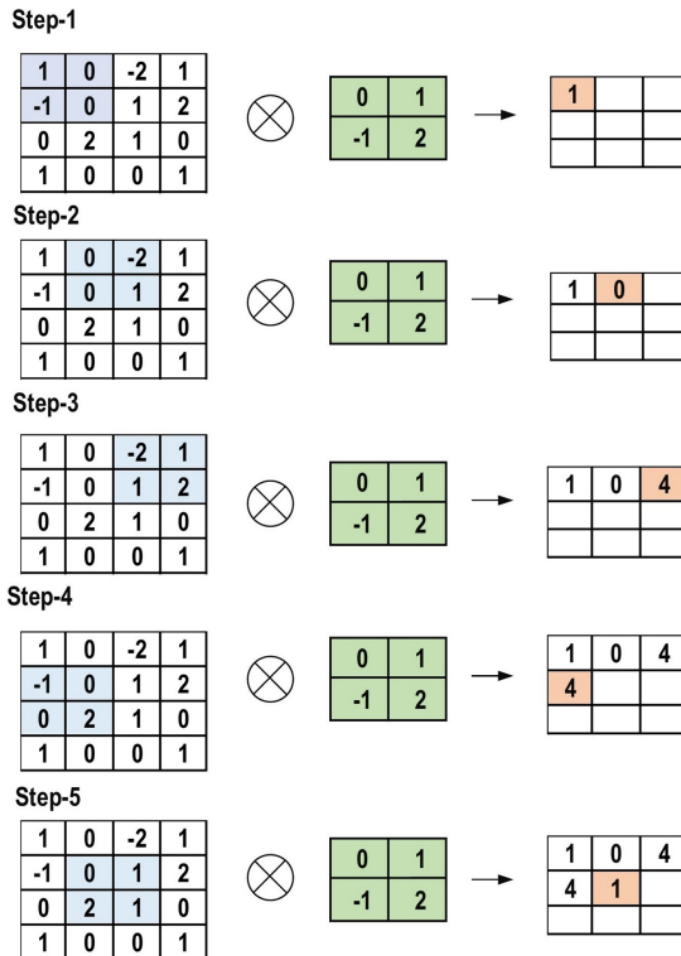
Arsitektur model CNN terdiri dari beberapa lapisan atau blok bangunan (*building blocks*) yang memiliki fungsi masing-masing. Komponen paling signifikan dalam arsitektur CNN adalah lapisan konvolusi (Alzubaidi dkk., 2021). Lapisan konvolusi terdiri dari kumpulan filter konvolusi. Citra masukan, yang diwakili sebagai matriks berdimensi  $M \times N$ , dikonvolusi dengan filter-filter konvolusi untuk menghasilkan peta fitur keluaran.

1. Definisi Kernel: Kernel adalah kisi berisi angka-angka atau nilai-nilai diskrit. Kernel memiliki dimensi lebar, dan tinggi. Ketika kernel



diinisialisasi, setiap elemennya diberikan nilai awal. Setiap nilai disebut sebagai bobot kernel. Pada awal pelatihan CNN, angka-angka acak diberikan sebagai bobot kernel. Pada setiap posisi di gambar, terjadi operasi *dot product* antara bagian gambar pada posisi tersebut dengan kernel yang digunakan. Peta aktivasi didapatkan dengan menggeser kernel ke berbagai posisi pada gambar. (Alzubaidi dkk., 2021).

2. Operasi Konvolusi: operasi konvolusi adalah penggeseran filter kecil di atas gambar, dan melakukan operasi *dot product* piksel demi piksel yang menghasilkan gambar baru dengan mengungkapkan informasi baru tentang gambar asli. Ilustrasi dari penerapan operasi konvolusi dengan menggunakan contoh citra *grayscale* 4x4 dengan kernel berbobot acak berukuran 2x2. Pertama, kernel ini digeser ke seluruh citra secara horizontal dan vertikal. Selanjutnya, produk titik antara citra masukan dan kernel ditentukan, di mana nilai-nilai yang sesuai dikalikan dan kemudian dijumlahkan untuk menciptakan nilai skalar tunggal yang dihitung secara bersamaan. Seluruh proses ini diulangi hingga tidak ada lagi pergeseran yang mungkin. Nilai-nilai produk titik yang dihitung mewakili peta fitur keluaran (Alzubaidi dkk., 2021). Perhitungan pada setiap langkah lapisan konvolusi dapat diilustrasikan pada Gambar 4.



Gambar 4. Perhitungan pada setiap langkah lapisan konvolusi (Alzubaidi dkk., 2021).

Gambar 4 mengilustrasikan perhitungan utama yang dilakukan pada setiap langkah. Pada gambar ini, warna hijau muda mewakili kernel 2x2, sementara warna biru muda mewakili area citra masukan yang berukuran serupa. Keduanya dikalikan, lalu hasil akhir setelah penjumlahan nilai produk yang dihasilkan (ditandai dengan warna oranye muda) mewakili nilai entri pada peta fitur keluaran. *Stride* satu digunakan pada *kernel*. Memungkinkan untuk menggunakan nilai *stride* yang berbeda. Selain itu, penggunaan *padding* sangat penting untuk menentukan informasi ukuran batas terkait citra masukan. Sebaliknya, fitur sisi batas bergerak dengan cepat. Penerapana *padding* mengakibatkan ukuran citra masukan akan meningkat, dan sebagai hasilnya, ukuran peta fitur keluaran juga akan

meningkat. Ukuran spasial dari *output* lapisan konvolusi dapat dihitung berdasarkan Persamaan (2.6) sebagai berikut:

$$\frac{N - F + 2P}{S} + 1 \quad (2.6)$$

Dimana

$N$  : ukuran spasial dari gambar *input*

$F$  : ukuran spasial dari filter

$P$  : jumlah penambahan angka (umumnya 0) untuk menyesuaikan ukuran gambar

$S$  : besaran pergeseran filter di setiap proses komputasi

### 2.8.2 Fungsi Aktivasi

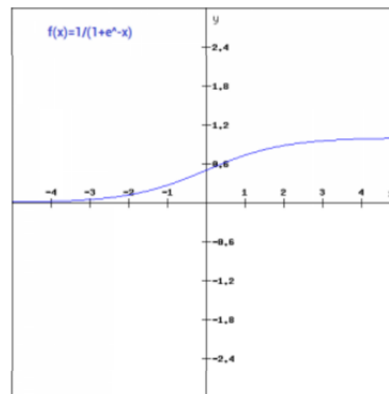
Fungsi aktivasi merupakan komponen inti dalam seluruh jenis jaringan saraf, termasuk dalam arsitektur jaringan saraf tiruan berkedalaman seperti CNN. Fungsi aktivasi bertanggung jawab untuk mengonversi nilai masukan menjadi keluaran yang sesuai. Nilai masukan dihitung dengan menjumlahkan bobot terhadap masukan dari *neuron* dan bias. Fungsi aktivasi membuat keputusan apakah neuron tersebut akan aktif atau tidak berdasarkan masukan tertentu yang kemudian menghasilkan keluaran yang sesuai.

Pada arsitektur CNN, lapisan-lapisan aktivasi *non-linear* digunakan setelah semua lapisan-lapisan yang memiliki bobot seperti lapisan-lapisan terdalam dan lapisan-lapisan konvolusi. Kehadiran lapisan aktivasi *non-linear* ini mengubah pemetaan dari masukan ke keluaran menjadi *non-linear*, dan memberikan kemampuan pada CNN untuk mempelajari pola-pola yang lebih kompleks. Selain itu, fungsi aktivasi harus memiliki kemampuan untuk melakukan diferensiasi, yang merupakan fitur penting karena memungkinkan penggunaan algoritma propagasi kesalahan (*error back-propagation*) untuk melatih jaringan.

Beberapa jenis fungsi aktivasi yang umum digunakan dalam CNN dan jaringan saraf lainnya antara lain:

1. *Sigmoid*; Fungsi ini menerima bilangan *real* sebagai masukan dan menghasilkan keluaran antara 0 hingga 1. Persamaan fungsi aktivasi *sigmoid* dapat dilihat pada Persamaan (2.7) sedangkan gambar fungsi aktivasi *sigmoid* dapat diilustrasikan pada Gambar 5.

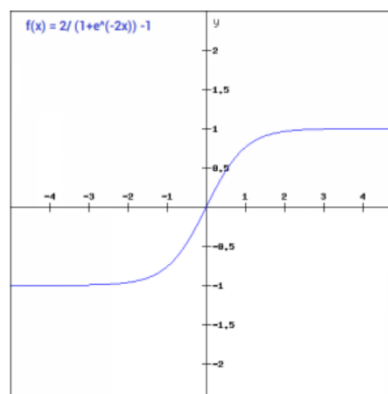
$$f(x)_{\text{sigm}} = \frac{1}{1 + e^{-x}} \quad (2.7)$$



Gambar 5. Fungsi aktivasi *sigmoid* (Sharma dkk., 2020).

2. *Hyperbolic tangent* (*tanh*); masukan fungsi ini adalah bilangan real, tetapi keluarannya terbatas pada rentang -1 hingga 1. Persamaan dari fungsi aktivasi *tanh* dapat dilihat pada Persamaan (2.8) dan gambar fungsi aktivasi *tanh* dapat diilustrasikan pada Gambar 6.

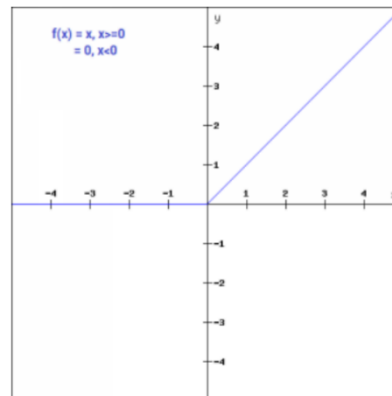
$$f(x)_{\text{tanh}} = \frac{e^x - e^{-x}}{e^x + e^{-x}} \quad (2.8)$$



Gambar 6. Fungsi aktivasi *tanh* (Sharma dkk., 2020).

3. *Rectified Linear Unit* (ReLU); Fungsi ini merupakan salah satu yang paling umum digunakan dalam konteks CNN. Fungsi ReLU mengubah seluruh nilai masukan menjadi bilangan positif. Keuntungan utama ReLU adalah beban komputasi yang lebih rendah dibandingkan dengan fungsi aktivasi lainnya. Berikut adalah fungsi ReLU dapat dilihat pada Persamaan (2.9) dan gambar dari fungsi aktivasi ReLU dapat diilustrasikan pada Gambar 7.

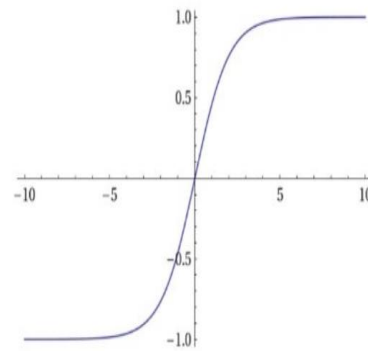
$$f(x)_{ReLU} = \max(0, x) \quad (2.9)$$



Gambar 7. Fungsi aktivasi ReLU (Sharma dkk., 2020).

4. *Softmax*; Fungsi ini umumnya digunakan untuk mengukur kinerja model CNN. Juga dikenal sebagai fungsi kerugian log. *Output*-nya adalah probabilitas  $p \in \{0,1\}$ . Selain itu, biasanya digunakan sebagai pengganti fungsi kerugian kesalahan kuadrat dalam masalah klasifikasi multi-kelas. Di lapisan output, fungsi aktivasi *softmax* digunakan untuk menghasilkan output dalam bentuk distribusi probabilitas. Fungsi aktivasi *softmax* dapat dilihat pada Persamaan (2.10) dan fungsi aktivasi *softmax* dapat diilustrasikan pada Gambar 8.

$$p_i = \frac{e^{a_i}}{\sum_{k=1}^N e_k^a} \quad (2.10)$$



Gambar 8. Fungsi aktivasi *softmax* (Kajalpawar, 2023).

### 2.8.3 Lapisan *Pooling*

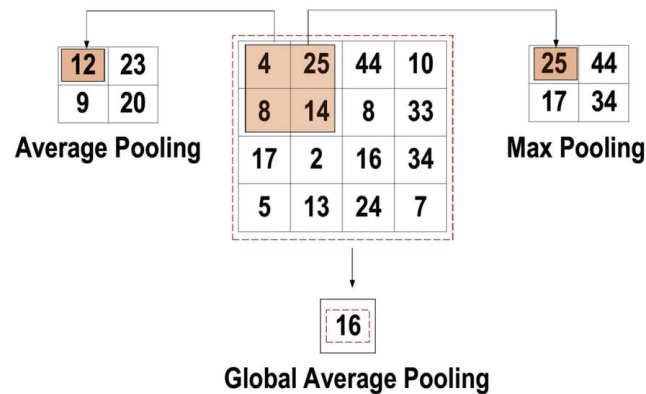
Lapisan *pooling* pada arsitektur CNN merupakan komponen yang digunakan untuk melakukan *sub-sampling* atau pengurangan ukuran terhadap peta fitur yang dihasilkan oleh lapisan konvolusi sebelumnya. Tujuannya adalah untuk menghasilkan peta fitur yang lebih kecil dari peta fitur asli, sambil tetap mempertahankan sebagian besar informasi penting.

Proses *pooling* melibatkan penggunaan operasi pada peta fitur dengan menggeser *kernel* pada seluruh peta fitur dan mengambil nilai tertentu dari setiap *kernel* tersebut. Dua parameter utama yang dapat dikontrol dalam lapisan *pooling* adalah:

1. Ukuran *Kernel*; *kernel* mengacu pada ukuran yang digunakan untuk mengambil nilai dari peta fitur. Biasanya, ukuran kernel bersifat kecil, seperti 2x2 atau 3x3, tetapi ukuran yang lebih besar juga dapat digunakan.
2. *Stride*; *stride* mengatur sejauh berapa *kernel* digeser pada setiap iterasi. *Stride* dapat mempengaruhi seberapa banyak informasi yang akan dipertahankan dan ukuran peta fitur yang dihasilkan.

Terdapat beberapa jenis operasi *pooling* yang umum digunakan, termasuk *Max Pooling*, *Min Pooling*, dan *Average Pooling*. Operasi *Max Pooling*, misalnya, akan mengambil nilai maksimum dari setiap *kernel*, sementara operasi *Average Pooling*

akan mengambil nilai rata-rata. Tiga tipe operasi pooling dapat diilustrasikan pada Gambar 9.



Gambar 9. Tiga tipe operasi *pooling* (Alzubaidi dkk., 2021).

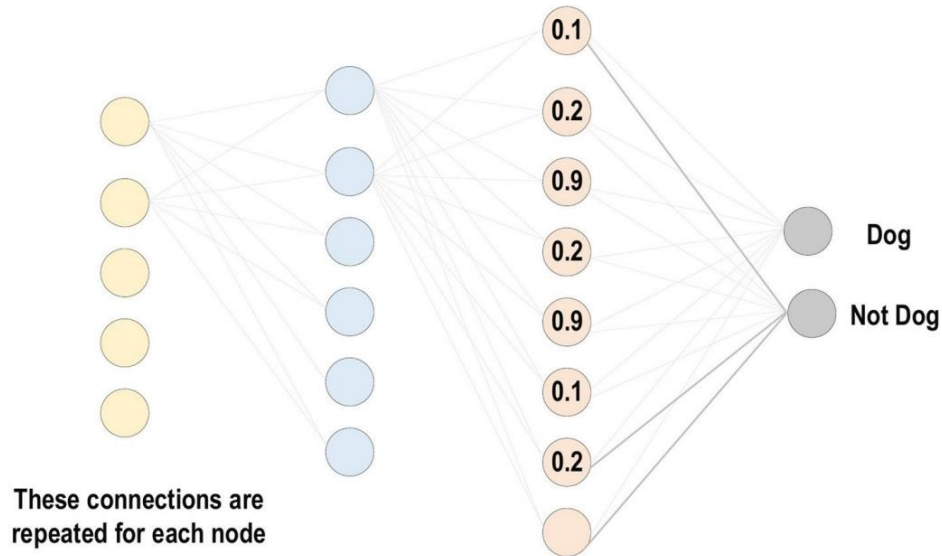
Lapisan *pooling* memiliki beberapa manfaat, seperti mengurangi ukuran peta fitur sehingga mengurangi jumlah parameter yang perlu dipelajari oleh jaringan (mengurangi *overfitting*), dan membantu menjaga invariansi terhadap translasi sehingga memungkinkan jaringan untuk lebih baik dalam mengenali fitur yang sama dalam berbagai posisi.

Namun, lapisan *pooling* juga memiliki beberapa kelemahan, seperti kehilangan informasi spasial yang lebih detail karena pengurangan ukuran, yang dapat mengurangi kemampuan jaringan untuk mengenali objek kecil. Selain itu, dalam beberapa kasus, penggunaan pooling dapat mengakibatkan penurunan kinerja jaringan, terutama ketika informasi spasial yang tepat diperlukan.

#### 2.8.4 Lapisan *Fully Connected*

Lapisan *fully connected* (FC) umumnya terletak di akhir setiap arsitektur CNN. Pada lapisan FC, setiap *neuron* terhubung dengan semua *neuron* dari lapisan sebelumnya, yang disebut sebagai pendekatan FC. Lapisan FC mengikuti metode dasar dari jaringan saraf *perceptron* berlapis ganda konvensional, karena merupakan jenis jaringan saraf maju. Masukan dari lapisan FC berasal dari lapisan

*pooling* atau konvolusi terakhir. Input pada lapisan FC berupa vektor, yang dibuat dari peta fitur setelah dilapisi. *Output* dari lapisan FC mewakili *output* akhir CNN. Lapisan FC diilustrasikan pada Gambar 10.



Gambar 10. Lapisan *Fully-connected* (Alzubaidi dkk., 2021).

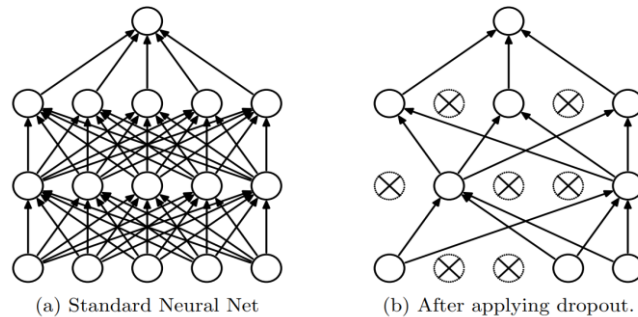
Pentingnya lapisan FC terletak pada kemampuannya untuk menggabungkan informasi dari seluruh peta fitur dan memahami hubungan antara berbagai fitur yang diekstraksi oleh lapisan sebelumnya. Setiap *neuron* dihubungkan dengan semua *neuron* di lapisan sebelumnya. Lapisan FC dapat mempelajari pola-pola yang kompleks dalam data dan menghasilkan representasi yang lebih tingkat atas fitur-fitur tersebut. Selain itu, lapisan FC juga berfungsi sebagai penghubung antara ekstraksi fitur dan *output* akhir model, yang menjadikannya bagian kunci dalam proses klasifikasi dan pengambilan keputusan.

### 2.8.5 Dropout Regularization

*Dropout Regularization* adalah teknik umum yang digunakan dalam pelatihan jaringan saraf untuk mengatasi masalah *overfitting*, yaitu kondisi di mana model memiliki kinerja yang baik pada data pelatihan, tetapi kinerjanya menurun saat diuji



pada data yang berbeda. Contoh hasil penggunaan *dropout* pada jaringan saraf standar dapat diilustrasikan pada Gambar 11.



Gambar 11. Contoh Penggunaan *Dropout Regularization* (Srivastava dkk., 2014).

Teknik ini bekerja dengan cara acak "mengabaikan" sebagian *neuron* selama setiap tahap pelatihan. Hal ini mengakibatkan proses pemilihan fitur tersebar merata di seluruh kelompok neuron, serta mendorong model untuk mempelajari fitur-fitur yang independen satu sama lain. Neuron yang di-*drop* tidak berpartisipasi dalam proses propagasi mundur (*back-propagation*) atau propagasi maju (*forward-propagation*). Namun, saat model digunakan untuk membuat prediksi pada tahap pengujian, seluruh jaringan saraf tetap aktif.

Metode *dropout regularization* melibatkan konsep bahwa jaringan saraf dapat dianggap sebagai sekumpulan *neuron* yang berurutan. Alih-alih menggunakan seluruh jaringan saraf saat prediksi, beberapa neuron dihapus secara acak dari jalur pelatihan. Efek dari setiap *dropout* terhadap akurasi keseluruhan dievaluasi, dan setelah beberapa iterasi, rangkaian *neuron* yang optimal dipilih untuk digunakan dalam model akhir. *Dropout regularization* efektif digunakan pada jaringan saraf yang dalam dan kompleks. Teknik ini membantu mencegah *overfitting*, yang merupakan tantangan umum dalam pelatihan jaringan saraf untuk tugas-tugas beragam, termasuk pengenalan gambar dan pemrosesan bahasa alami.

### 2.8.6 *Optimizer*

*Optimizer* merupakan algoritma atau metode yang digunakan untuk mengatur dan memperbarui parameter seperti bobot dan *learning rate* dalam model sehingga model tersebut dapat belajar dari data pelatihan dan menghasilkan hasil yang optimal (Asy Syifa dan Amelia Dewi, 2022). Tujuan utama dari *optimizer* adalah meminimalkan fungsi kerugian atau kesalahan prediksi model sehingga model dapat melakukan prediksi yang lebih akurat pada data baru. Pemilihan *optimizer* yang sesuai dan mengatur parameter-parameter yang diperlukan dapat membantu model jaringan saraf untuk mencapai konvergensi yang lebih cepat dan menghasilkan hasil yang lebih baik pada data yang belum pernah dilihat sebelumnya.

Pada dasarnya, *optimizer* bekerja dengan menghitung gradien dari fungsi kerugian terhadap parameter-parameter dalam model. Gradien ini menunjukkan arah di mana parameter-parameter harus diperbarui untuk mengurangi kesalahan. Kemudian, *optimizer* menggunakan gradien ini untuk menghitung perubahan yang harus diterapkan pada setiap parameter dalam model. Proses ini berulang secara iteratif selama pelatihan model.

Beberapa *optimizer* yang umum digunakan dalam pembelajaran mesin dan jaringan saraf termasuk algoritma *Gradient Descent*, *Stochastic Gradient Descent* (SGD), *Mini-batch Gradient Descent*, Momentum, dan *Adaptive Moment Estimation* (Adam). Setiap *optimizer* memiliki karakteristik dan keunggulan masing-masing dalam menghadapi tugas-tugas pembelajaran yang berbeda.

### 2.8.7 Fungsi Kerugian

Fungsi kerugian adalah salah satu komponen penting dalam ML dan jaringan saraf seperti CNN. Fungsi kerugian digunakan untuk mengukur seberapa baik atau buruk model jaringan saraf dalam membuat prediksi atau klasifikasi terhadap data yang diberikan. Fungsi kerugian mengukur perbedaan antara nilai yang diprediksi oleh model (yang disebut prediksi) dan nilai yang seharusnya (yang disebut label atau target). Perhitungan fungsi kerugian memiliki berbagai macam jenis tergantung pada jenis tugas yang dilakukan. Jenis tugas yang umumnya digunakan dalam pembelajaran mesin dan jaringan saraf meliputi klasifikasi, regresi, *unsupervised learning*, *object detection*, dan *face recognition* (Wang dkk., 2022).

Pada tugas klasifikasi, seperti pengenalan gambar, fungsi kerugian seperti *cross-entropy* sering digunakan. Fungsi kerugian tersebut mengukur seberapa baik model memetakan *input* ke probabilitas kelas yang berbeda. Hal tersebut berguna dalam tugas-tugas seperti pengenalan gambar di mana model harus memilih salah satu dari beberapa kelas yang mungkin. Fungsi *cross-entropy* juga dikenal sebagai fungsi kerugian *log*. Pada lapisan *output*, fungsi ini menggunakan aktivasi *softmax* untuk menghasilkan distribusi probabilitas. Representasi matematis dari fungsi kerugian *cross-entropy* disajikan pada Persamaan (2.11) sebagai berikut:

$$H(p, y) = - \sum_i y_i \log(p_i) \text{ dimana } i \in [1, N] \quad (2.11)$$

Di sisi lain, untuk tugas regresi di mana model harus memprediksi nilai numerik, fungsi kerugian seperti *euclidean* atau *mean square Error* lebih cocok. Fungsi-fungsi ini mengukur seberapa baik model memprediksi nilai yang sesuai dengan data aktual. Representasi matematis dari fungsi kerugian *euclidean* ini disajikan pada Persamaan (2.12) sebagai berikut:

$$H(p, y) = \frac{1}{2N} \sum_{i=1}^N (p_i - y_i)^2 \quad (2.12)$$

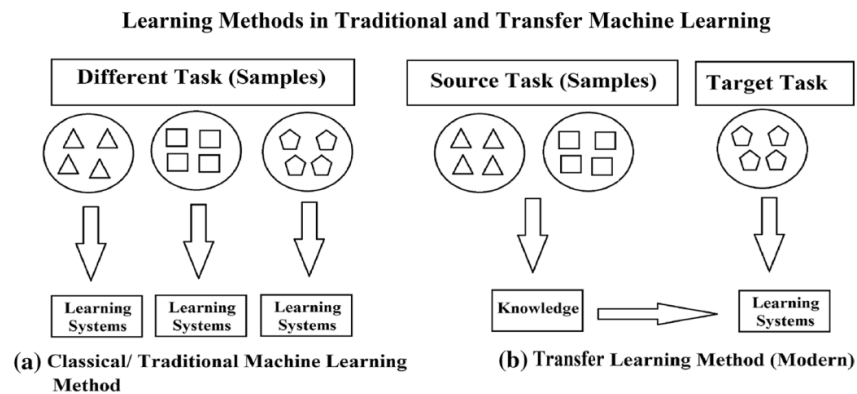
Jenis tugas lainnya, seperti *unsupervised learning*, *object detection*, dan *face recognition*, mungkin memerlukan fungsi kerugian khusus yang dirancang sesuai

dengan karakteristik tugas tersebut. Pada semua kasus, pemilihan fungsi kerugian yang tepat adalah langkah kunci untuk mengoptimalkan kinerja model dan mencapai hasil yang diinginkan.

## ***2.9 Transfer Learning***

*Transfer Learning* merupakan suatu pengembangan dari pendekatan dalam bidang ML yang fokus pada pemanfaatan model-model yang sudah ada untuk mengatasi tantangan-tantangan. *Transfer learning* memungkinkan penggunaan pengetahuan yang telah diperoleh dari tugas-tugas sebelumnya untuk meningkatkan kinerja dalam menyelesaikan tugas yang sedang dihadapi, tanpa perlu membangun model dari awal. Analogi sederhana dari *transfer learning* adalah ketika seseorang belajar cara mengendarai sepeda motor. Jika seseorang memiliki pengalaman sebelumnya dalam mengendarai sepeda, dia akan lebih cepat belajar daripada orang yang sama sekali tidak memiliki pengalaman. Konsep *transfer learning* mirip dengan ide bahwa manusia dapat mengaitkan keterampilan yang telah diperoleh sebelumnya dalam menyelesaikan tantangan-tantangan saat ini dengan lebih cepat dan akurat.

*Transfer learning*, berfokus pada pembelajaran beberapa tugas secara bersamaan, meskipun tugas-tugas tersebut berbeda. *Transfer learning* juga diperlakukan dalam kaitannya dengan pembelajaran multi-tugas. Perbedaan antara pembelajaran *traditional machine learning* dan *transfer learning* modern dapat dilihat pada Gambar 12. *Traditional machine learning* mencoba untuk memulai dari nol, sementara *transfer learning* berusaha untuk mentransfer informasi dari tugas-tugas utama ke tugas baru dengan kumpulan data pelatihan berkualitas tinggi.



Gambar 12. Perbedaan *traditional machine learning* dengan *transfer learning* (Hosna dkk., 2022).

Selain itu, *transfer learning* menjadi penting ketika ketersediaan kumpulan data terbatas, yang bisa disebabkan oleh masalah-masalah data seperti jarangya data, biaya untuk mengumpulkan dan mengevaluasi data, atau bahkan ketidakterediaan data. Namun, strategi *transfer learning* semakin menarik karena sumber data besar menjadi lebih mudah diakses. Penggunaan kumpulan data yang terkait tetapi tidak sama memungkinkan pendekatan pembelajaran ini menjadi metode yang efektif. *Transfer learning* telah berhasil diterapkan dalam berbagai aplikasi pembelajaran mesin, seperti dalam simulasi dunia nyata, klasifikasi sentimen, permainan, klasifikasi gambar, dan *zero-shot translation*.

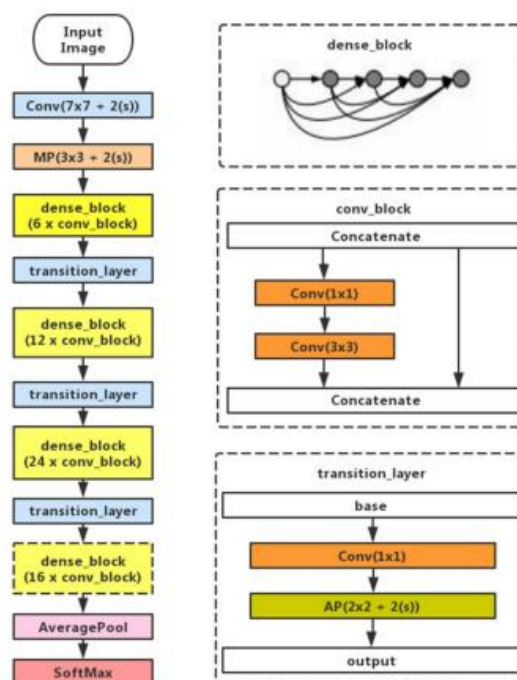
Model *pre-trained* dapat digunakan untuk *transfer learning* dengan dua cara.

1. Cara pertama adalah *Fixed Feature Extractor*. Pada metode ini, digunakan model *pre-trained* yang telah dilatih tanpa melatih ulang model tersebut. Bobot model *pre-trained* dibiarkan tetap tidak berubah.
2. Cara kedua adalah *Fine Tuning*. Pada metode ini, digunakan model *pre-trained* yang telah dilatih dan kemudian melatih ulang model tersebut untuk kebutuhan yang baru. Bobot model *pre-trained* ini akan diubah sesuai dengan kebutuhan. *Inception*, *Xception*, *VGG*, dan *Resnet* merupakan beberapa contoh model *pre-trained* yang dapat digunakan untuk *transfer learning*.

Penelitian ini menggunakan lima model *pre-trained* antara lain:

a. *Densely Connected Convolutional Networks 121 (DenseNet121)*

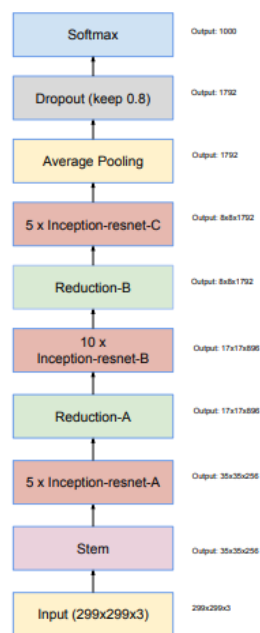
Model *DenseNet* adalah arsitektur CNN yang diperkenalkan pada tahun 2016 oleh Gao Huang, Zhuang Liu, Laurens van der Maaten, dan Kilian Q. Weinberger (Huang dkk., 2016). Model ini menghubungkan setiap lapisan dengan lapisan lainnya secara *feedforward*. Model *pre-trained* ini memiliki dua fitur utama yang membuatnya berbeda dari arsitektur CNN lainnya yaitu memiliki struktur blok yang padat, di mana setiap lapisan terhubung ke setiap lapisan lainnya secara *feedforward*. Serta menggunakan lapisan *bottleneck* yang membantu mengurangi jumlah parameter tanpa mengurangi jumlah fitur yang dipelajari oleh jaringan. *DenseNet121*, merupakan varian *DenseNet* dengan 121 lapisan. Model tersebut mendefinisikan tiga kelas: *Bottleneck*, *Transition*, dan *DenseNet*. Kelas *Bottleneck* mewakili lapisan bottleneck dari arsitektur *DenseNet*. Kelas *Transition* mewakili lapisan transisi, yang digunakan untuk mengurangi dimensi spasial peta fitur. Kelas *DenseNet* mewakili seluruh arsitektur *DenseNet* (Huang dkk., 2016). Arsitektur *DenseNet121* dapat diilustrasikan pada Gambar 13.



Gambar 13. Arsitektur *DenseNet121* (Ji dkk., 2019).

b. *Inception Residual Network v2 (InceptionResNetv2)*

Model *InceptionResNet* merupakan model jaringan saraf hasil kombinasi dua pendekatan terbaru yaitu koneksi residual dan revisi terbaru dari arsitektur Inception (Szegedy dkk., 2016). Model ini menggabungkan kedua pendekatan dengan memanfaatkan keuntungan dari koneksi residual untuk meningkatkan akurasi, namun tetap mempertahankan efisiensi komputasi yang menjadi ciri khas arsitektur *Inception*. Model *InceptionResNetV2* adalah jaringan saraf konvolusi yang telah dilatih dengan lebih dari satu juta gambar dari database *ImageNet*. Model ini memiliki 164 lapisan dan dapat mengklasifikasikan gambar ke dalam 1000 kategori objek. Arsitektur model *InceptionResNetV2* dapat diilustrasikan pada Gambar 14.

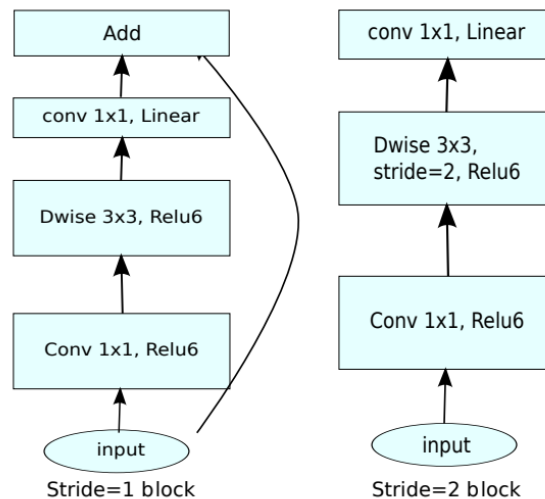


Gambar 14. Arsitektur *InceptionResNetV2* (Szegedy dkk., 2016).

c. *Mobile Network v2 (MobileNetV2)*

Model *MobileNetV2* memiliki arsitektur yang dirancang untuk efisiensi pada perangkat dengan sumber daya terbatas. Model ini menggunakan *depthwise separable convolution*. Model ini menggunakan arsitektur yang sangat sederhana sehingga memungkinkan pembuatan berbagai model yang sangat efisien untuk aplikasi *mobile*. Unit dasar yang digunakan dalam model

memiliki beberapa keunggulan yang membuatnya cocok untuk digunakan pada perangkat mobile. Keunggulan tersebut berupa penggunaan memorinya sangat efisien saat memproses informasi, serta menggunakan operasi-operasi standar yang sudah tersedia di semua *framework neural network* sehingga penggunaannya menjadi mudah (Sandler dkk., 2018). Arsitektur *MobileNetV2* dapat direpresentasikan pada Gambar 15.

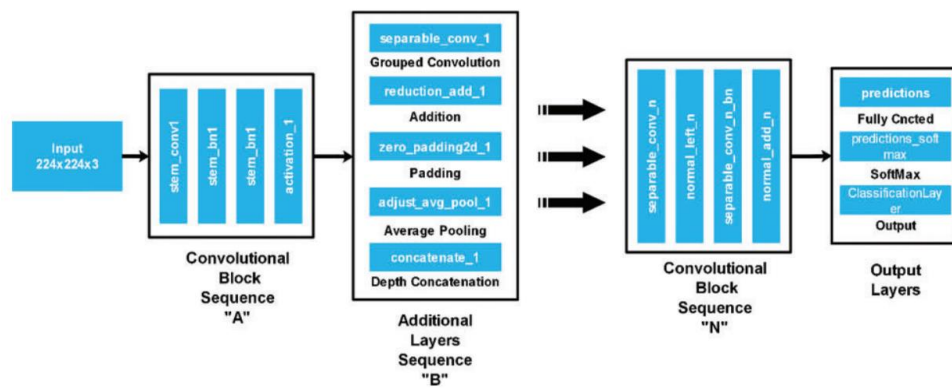


Gambar 15. Arsitektur *MobileNetV2* (Sandler dkk., 2018).

d. *Neural Architecture Search Network Mobile (NasNetMobile)*

Model *NasNetMobile* merupakan model jaringan saraf yang dikembangkan menggunakan teknik *neural architecture search* (NAS). Teknik NAS adalah teknik terbaru dalam pembelajaran mesin yang berfokus pada pencarian arsitektur jaringan saraf yang optimal. Secara umum, NAS terdiri dari tiga komponen *search space*, *search strategy*, *performance estimation*. *Search space* yang mencari kemungkinan struktur jaringan saraf, seperti lapisan *convolutional*, lapisan *fully-connected*, dan *pooling*. *search strategy* menggunakan pencarian acak (*random search*) dan pembelajaran *reinforcement learning* untuk memilih kandidat arsitektur jaringan saraf yang potensial. *Performance estimation* bertujuan untuk memperkirakan performa kandidat arsitektur jaringan saraf dengan melihat penggunaan sumber daya komputasi dan waktu yang dibutuhkan (Shah dkk., 2021). Arsitektur *NasNetMobile* dapat diilustrasikan pada Gambar 16.

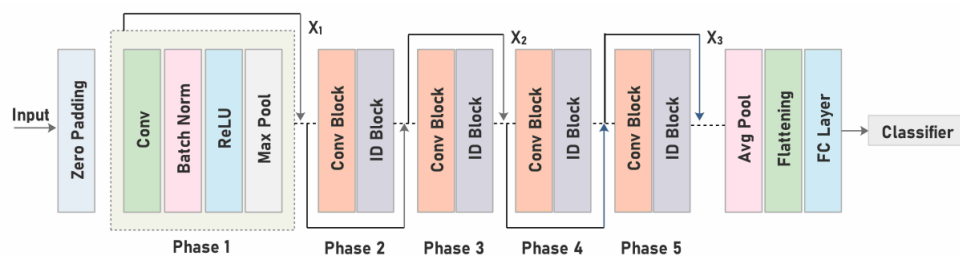




Gambar 16. Arsitektur *NasNetMobile* (Shah dkk., 2021).

e. *Residual Network 50 v2 (ResNet50V2)*

Model *ResNet* memiliki cara kerja yang berbeda dari jaringan saraf biasa. Model *ResNet* tidak mempelajari fungsi pemetaan dari *input* ke *output* secara langsung, tetapi mempelajari fungsi residual, yaitu perbedaan antara *input* dan *output* yang diinginkan. Model *ResNet* membuat lapisan-lapisan tersebut fokus mempelajari perbedaan antara input dan output yang diinginkan. Pendekatan ini membantu *ResNet* mengatasi masalah degradasi performa yang sering terjadi pada jaringan saraf yang sangat dalam (He et al., 2015). Model *ResNet50V2* merupakan versi yang lebih baik dari *ResNet50* dan telah mencapai hasil yang luar biasa dalam berbagai tugas, termasuk klasifikasi gambar, deteksi objek, dan segmentasi gambar. Arsitektur *ResNet50V2* disajikan pada Gambar 17.



Gambar 17. Arsitektur *ResNet50V2* (Patel dan Khan, 2023).

## 2.10 Evaluasi Model

Pada evaluasi model, digunakan sebuah metode pengukuran performa yang sering dikenal sebagai *confusion matrix*. Metode ini sangat berguna dalam masalah klasifikasi di mana output dapat memiliki dua kelas atau lebih. *Confusion matrix* untuk klasifikasi *multiclass* memiliki dimensi  $N \times N$ , dengan  $N$  merupakan jumlah label kelas yang dimiliki. Visualisasi dari *confusion matrix* disajikan pada Gambar 18.

		Predicted Class			
		$C_1$	$C_2$	...	$C_N$
Actual Class	$C_1$	$C_{1,1}$	FP	...	$C_{1,N}$
	$C_2$	FN	TP	...	FN
	...	...	...	...	...
	$C_N$	$C_{N,1}$	FP	...	$C_{N,N}$

Gambar 18. *Confusion Matrix* klasifikasi *multiclass* (Markoulidakis dkk., 2021).

*Confusion matrix* terdiri dari empat atribut yang mencerminkan kombinasi dari nilai yang diprediksi (*predicted*) dan model dengan nilai yang sebenarnya (*actual*). Keempat atribut tersebut adalah:

1. *True Positive* : Jumlah data yang bernilai positif baik pada kategori yang diprediksi maupun kategori yang sebenarnya.
2. *False Positive* : Jumlah data yang bernilai positif pada kategori yang diprediksi tetapi bernilai negatif pada kategori yang sebenarnya.
3. *True Negative* : Jumlah data yang bernilai negatif baik pada kategori yang diprediksi maupun kategori yang sebenarnya.
4. *False Negative*: Jumlah data yang bernilai negatif pada kategori yang diprediksi tetapi bernilai positif pada kategori yang sebenarnya

Keempat atribut tersebut akan menjadi dasar perhitungan beberapa metrik evaluasi yang umum digunakan untuk masalah klasifikasi, seperti akurasi, presisi, sensitifitas, dan spesifisitas (Ramírez-Arias dkk., 2022). Pada klasifikasi multi-kelas yang memiliki kelas lebih dari 2, diperlukan suatu metrik evaluasi yang

memberikan rata-rata dari nilai akurasi, presisi, sensitifitas, dan spesifisitas dari tiap-tiap kelas (Hossin dan Sulaiman, 2015)

### 2.10.1 Rata-rata Akurasi

Akurasi merupakan rasio prediksi benar (positif dan negatif) dengan keseluruhan data. Metrik ini paling umum digunakan karena mudah dihitung dan digunakan dalam melakukan klasifikasi. Pada klasifikasi multiclass diperlukan nilai yang memberikan rata-rata dari akurasi tiap kelas. Nilai rata-rata akurasi dapat dihitung dengan Persamaan (2.13) sebagai berikut:

$$acc_{avg} = \sum_{i=1}^l \frac{TP_i}{TP_i + FP_i + FN_i + TN_i} \quad (2.13)$$

Dimana

$TP_i$  : *True positive* untuk  $C_i$

$TN_i$  : *True negative* untuk  $C_i$

$FP_i$  : *False positive* untuk  $C_i$

$FN_i$  : *False Negative* untuk  $C_i$

$l$  : banyak kelas / kategori yang dimiliki

### 2.10.2 Rata-rata Presisi

Presisi merupakan rasio antara *true positive* (TP) dengan keseluruhan data yang diprediksi positif. Sehingga, presisi berusaha memperkecil terjadinya *false positive* (FP). Pada klasifikasi multi-kelas diperlukan nilai rata-rata dari presisi tiap kelas. Nilai rata-rata presisi dapat dihitung dengan Persamaan (2.14) dan Persamaan (2.15) sebagai berikut:

$$p_{C_i} = \frac{TP_i}{TP_i + FP_i} \quad (2.14)$$

$$p_{avg} = \frac{\sum_{i=1}^l p_{C_i}}{l} \quad (2.15)$$

Dimana

$P_{C_i}$  : Presisi untuk  $C_i$

$TP_i$  : *True positive* untuk  $C_i$

$FP_i$  : *False positive* untuk  $C_i$

$l$  : banyak kelas / kategori yang dimiliki

### 2.10.3 Rata-rata Sensitifitas

Sensitifitas merupakan rasio antara *true positive* (TP) dengan keseluruhan data yang kenyataannya bernilai positif. Sehingga, sensitifitas berusaha memperkecil terjadinya *false negative* (FN). Pada klasifikasi multi-kelas diperlukan nilai yang memberikan rata rata sensitifitas dari tiap kelas. Nilai rata-rata sensitifitas dapat dihitung pada Persamaan (2.16) dan Persamaan (2.17) sebagai berikut:

$$r_{C_i} = \frac{TP_i}{TP_i + FN_i} \quad (2.16)$$

$$r_{avg} = \frac{\sum_{i=1}^l r_{C_i}}{l} \quad (2.17)$$

Dimana

$r_{C_i}$  : Sensitifitas untuk  $C_i$

$TP_i$  : *True positive* untuk  $C_i$

$FN_i$  : *False Negative* untuk  $C_i$

$l$  : banyak kelas / kategori yang dimiliki

#### 2.10.4 Rata-rata Spesifisitas

Spesifisitas merupakan rasio antara *true negative* (TN) dengan keseluruhan data yang kenyataannya bernilai negatif. Sehingga, spesifisitas berusaha memperkecil terjadinya *false positive* (FP). Pada klasifikasi multi-kelas diperlukan nilai yang memberikan rata-rata spesifisitas dari tiap kelas. Nilai rata-rata spesifisitas dapat dihitung pada Persamaan (2.18) dan Persamaan (2.19) sebagai berikut:

$$s_{C_i} = \frac{TN_i}{TN_i + FP_i} \quad (2.18)$$

$$s_{avg} = \frac{\sum_{i=1}^l s_{C_i}}{l} \quad (2.19)$$

Dimana

$s_{C_i}$  : Spesifisitas untuk  $C_i$

$TN_i$  : *True positive* untuk  $C_i$

$FP_i$  : *False Negative* untuk  $C_i$

$l$  : banyak kelas / kategori yang dimiliki

### III. METODOLOGI PENELITIAN

#### 3.1 Tempat dan Waktu Penelitian

Berikut merupakan tempat dan waktu yang dilakukan dalam mengerjakan penelitian:

a. Tempat Penelitian

Penelitian ini dilakukan pada Jurusan Matematika, Fakultas Matematika dan Ilmu Pengetahuan Alam, Universitas Lampung. Lokasi bertempat di Jalan Prof. Dr. Soemantri Brojonegoro No. 1, Gedung Meneng, Bandar Lampung.

b. Waktu Penelitian

Penelitian dilakukan pada semester ganjil tepatnya Agustus 2023 hingga penyelesaian bulan Februari 2024 tahun ajaran 2023/2024. Pengerjaan terbagi menjadi tiga tahap, tahap pertama melakukan pengumpulan dan pemahaman terhadap studi literatur serta menentukan tema penelitian. Setelahnya menentuka tema penelitian, akan dilakukan pengumpulan data terkait. Tahap satu memerlukan waktu pengerjaan kurang lebih 15 minggu. Tahap ke-2 merupakan tahap pengerjaan program yang dimulai dari *pre-processing data*, augmentasi data, dan pemodelan klasifikasi menggunakan model *transfer learning* dengan CNN hingga tahap pengujian program yang memerlukan waktu kurang lebih 5 minggu. Tahap terakhir merupakan tahap penyusunan hasil pengujian dan analisis program dalam draf laporan dan penyampaian hasil penelitian melalui seminar hasil dan sidang komprehensif kurang lebih 7 minggu.

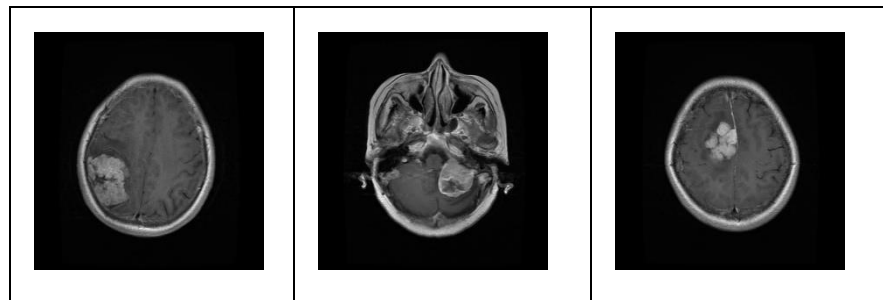
### 3.2 Data dan Alat

Data dan alat yang digunakan sebagai penunjang penelitian ini ialah:

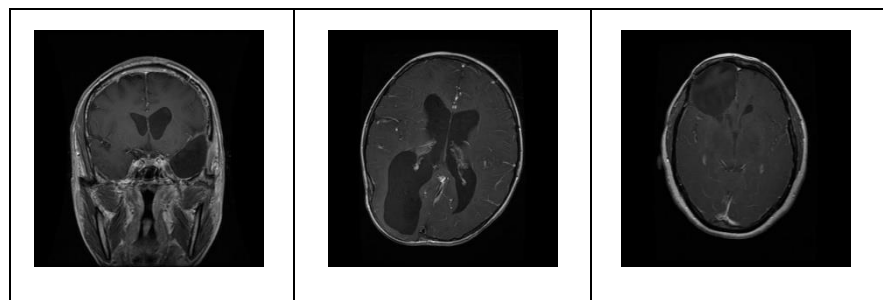
a. Data

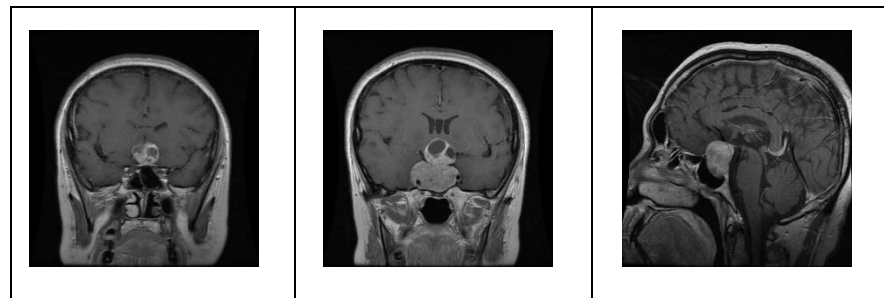
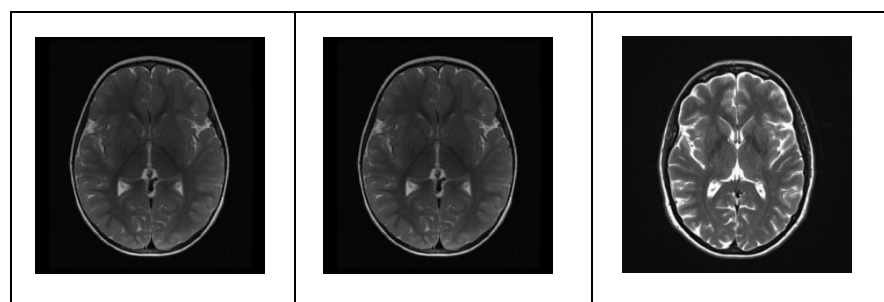
Data yang digunakan dalam penelitian ini merupakan data sekunder yang diperoleh secara *online* pada *website kaggle* yang dapat diakses pada <https://doi.org/10.34740/KAGGLE/DSV/2645886> yang merupakan data gambar mengenai penyakit tumor pada otak. Kumpulan data ini terdiri dari 7020 gambar MRI otak manusia yang terklasifikasi ke dalam 4 kategori atau *class* yaitu *meningioma* yang terdiri atas 1644 gambar, *glioma* yang terdiri atas 1620 gambar, *pituitary* yang terdiri atas 1756 gambar, dan *no-tumor* yang terdiri atas 2000 gambar. Kumpulan data merupakan hasil gabungan dari tiga sumber yang berbeda, yaitu figshare, SARTAJ, dan Br35H.

Tabel 2. Data gambar penyakit tumor jenis *meningioma*



Tabel 3. Data gambar penyakit tumor jenis *glioma*



Tabel 4. Data gambar penyakit tumor jenis *pituitary*Tabel 5. Data gambar penyakit tumor jenis *no-tumor*

## b. Alat

Peralatan yang digunakan dalam menunjang penelitian ini antara lain:

a) Perangkat Keras (*Hardware*)

Perangkat yang digunakan dalam penelitian ini adalah laptop dengan merek Acer Aspire A515-R5XD dengan tipe 64-bit *operating system*, *x64-based processor*. Spesifikasi *hardware* perangkat tersebut adalah sebagai berikut:

*Processor* : AMD Ryzen 5 5500u with *Radeon graphics*

*Memori* : SSD 475 GB dengan tambahan 957 GB

*RAM* : 16 GB

b) Perangkat Lunak (*Software*)

Pada penelitian ini, perangkat lunak yang digunakan untuk melakukan klasifikasi citra gambar dengan menggunakan *transfer learning* dengan CNN adalah sebagai berikut:

## 1) Sistem operasi

*Windows 11 Home Single Language 64-bit*, yaitu versi terbaru dari sistem operasi *windows* yang dikembangkan oleh



*Microsoft. Windows 11* memiliki fitur-fitur baru seperti antarmuka pengguna yang lebih modern, integrasi dengan *microsoft teams*, dan dukungan untuk aplikasi *android*.

2) *Virtual Studio Code 1.84.0*

*VS Code* yaitu *integrated development environment (IDE) open source* yang mendukung berbagai bahasa pemrograman dan *platform*. *VS Code* memiliki fitur-fitur seperti *debugging*, *code completion*, *syntax highlighting*, dan *extension*.

3) *Library TensorFlow 2.14.0*

*TensorFlow* yaitu *library open source* untuk *machine learning* yang dikembangkan oleh *google*. *TensorFlow* menyediakan berbagai algoritma dan model untuk *deep learning*, termasuk *CNN* dan *transfer learning*.

4) *Library NumPy 1.26.1*

*NumPy* yaitu *library open source* untuk komputasi ilmiah pada bahasa pemrograman *Python*. *NumPy* menyediakan objek array multidimensi dan fungsi-fungsi matematika yang dapat dioperasikan pada *array* tersebut.

5) *Library Matplotlib 3.7.2*

*Matplotlib* yaitu *library open source* untuk visualisasi data pada bahasa pemrograman *Python*. *Matplotlib* dapat menghasilkan berbagai jenis plot seperti *line plot*, *bar plot*, *scatter plot*, histogram, dan *pie chart*.

6) *Library Seaborn 0.12.2*

*Seaborn* yaitu *library open source* untuk visualisasi data statistik pada bahasa pemrograman *python*. *Seaborn* berbasis pada *matplotlib* dan menyediakan *interface* yang lebih mudah dan elegan untuk membuat plot statistik seperti *box plot*, *violin plot*, *heat map*, dan *pair plot*.

7) *Library OpenCV-Python 4.8.0.76*

*OpenCV-Python* yaitu *library open source* untuk *computer vision* dan *machine learning* pada bahasa pemrograman *Python*.

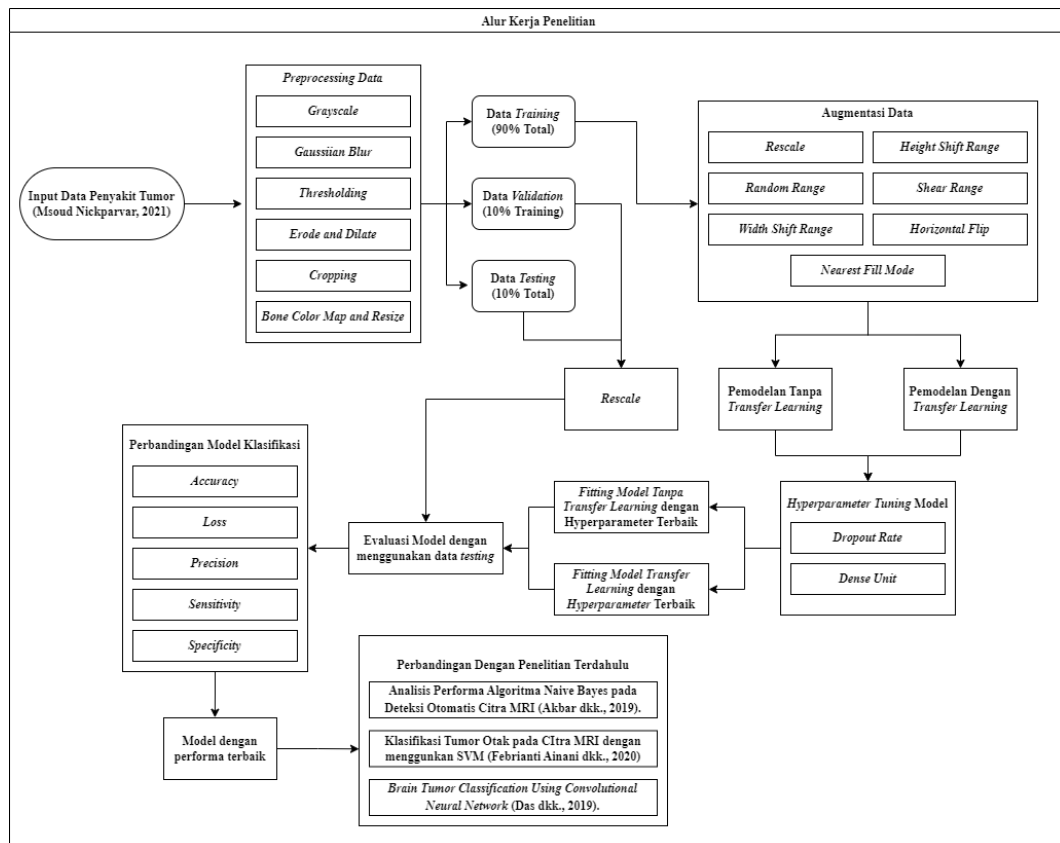
*OpenCV-Python* menyediakan fungsi-fungsi untuk mengolah citra gambar, seperti membaca, menulis, menampilkan, memotong, memutar, mengubah warna, mengaburkan, menajamkan, mendeteksi tepi, dan mengenali wajah.

8) *Library Imutils 0.5.4*

*Imutils* yaitu *library open source* yang menyediakan fungsi-fungsi bantuan untuk *computer vision* pada bahasa pemrograman *Python*. *Imutils* menyediakan fungsi-fungsi seperti mengubah ukuran citra gambar, memutar citra gambar dengan titik pusat tertentu, menerapkan transformasi perspektif pada citra gambar, dan mencari kontur pada citra biner.

### 3.3 Alur Kerja Penelitian

Alur kerja penelitian terdiri atas beberapa tahapan. Berikut adalah tahapan yang dilakukan dapat diilustrasikan pada Gambar 19.



Gambar 19. Alur Pengerjaan Penelitian.

Berdasarkan alur pengerjaan pada Gambar 19, berikut penjelasan setiap tahapan.

#### 1. Data Penyakit Tumor

Data yang digunakan dalam penelitian ini merupakan data sekunder yang diperoleh secara online pada *website kaggle* yang dapat diakses pada <https://doi.org/10.34740/KAGGLE/DSV/2645886> yang merupakan data gambar mengenai penyakit tumor pada otak. *Kumpulan data* ini terdiri dari 7020 gambar MRI otak manusia yang terklasifikasi ke dalam 4 kategori atau class yaitu *meningioma*, *glioma*, *pituitary*, dan *no-tumor*. *Kumpulan data* ini merupakan hasil gabungan dari tiga sumber kumpulan data berbeda, yaitu *figshare*, *SARTAJ*, dan *Br35H*

## 2. *Pre-processing Data*

*Pre-processing* pada data gambar yang telah di-input melibatkan konversi gambar menjadi citra skala abu-abu, penggunaan filter *gaussian* untuk mengurangi *noise*, *thresholding*, operasi morfologi, pengambilan kontur terbesar, identifikasi titik ekstrem, *cropping*, *bone colormap*, serta *resize* pada gambar untuk mempersiapkannya sebelum digunakan dalam proses pelatihan model jaringan saraf.

## 3. *Splitting Data*

Data yang telah melalui proses *pre-processing* akan dibagi menjadi tiga bagian utama yaitu pelatihan, pengujian, dan validasi. Kumpulan data terdiri dari 7.020 gambar. *Splitting* data dilakukan dengan rasio 90% untuk pelatihan (5.688 gambar), dan 10% dari kumpulan data total untuk pengujian (701 gambar). Pengambilan data validasi dilakukan dengan rasio 10% dari data pelatihan sebanyak 631. Rasion tersebut digunakan untuk melatih model, sementara validasi dan pengujian dilakukan pada subset data yang terpisah. Rasio 90:10 ini dipilih untuk memastikan bahwa model dapat belajar secara memadai, divalidasi, dan diuji pada data yang tidak pernah dilihat selama proses pelatihan.

## 4. *Augmentasi Data*

Melakukan proses augmentasi data menggunakan *ImageDataGenerator* untuk meningkatkan variasi data pelatihan. Augmentasi melibatkan serangkaian transformasi seperti rotasi, pergeseran lebar dan tinggi, *shear*, dan *horizontal flip*, yang bertujuan untuk memperkaya *kumpulan data* dan meningkatkan kemampuan model dalam memahami variasi gambar. Data pelatihan diolah dengan berbagai augmentasi, sementara data validasi dan data pengujian hanya mengalami *rescaling*. Langkah augmentasi penting dalam meningkatkan performa model dalam klasifikasi gambar.

## 5. *Pemodelan dan hyperparameter tuning*

Setelah dilakukan proses augmentasi data, langkah selanjutnya adalah membangun model-model yang akan dirancang pada penelitian yaitu model CNN berbasis *transfer learning* untuk mengklasifikasikan citra MRI otak dan model CNN tanpa basis *transfer learning* sebagai

perbandingan. Model yang akan dipakai menggunakan pendekatan *transfer learning* dengan mengimplementasikan beberapa *pre-trained model* yang melibatkan *DenseNet121*, *InceptionResNet V2*, *MobileNet V2*, *NasNetMobile*, dan *ResNet50 V2*. Setelah merancang model, dilakukan *hyperparameter tuning*. Proses tuning ini fokus pada dua *hyperparameter* utama, yaitu *dense unit* dan *dropout rate*, dengan menerapkan metode *grid search*. Setiap kombinasi nilai diuji untuk menentukan kombinasi yang optimal dan meningkatkan kinerja model pada tugas klasifikasi citra MRI otak.

6. *Model fitting* dengan menggunakan *hyperparameter* terbaik

Setelah mendapatkan *hyperparameter* terbaik dari *hyperparameter tuning*, langkah selanjutnya adalah melakukan *model fitting*, yaitu proses untuk melatih model dengan menggunakan data pelatihan dan menguji performanya dengan menggunakan data validasi lalu dievaluasi dengan menggunakan data pengujian. *Model fitting* bertujuan untuk menyesuaikan parameter dari model agar dapat meminimalkan fungsi kerugian dan memaksimalkan metrik kinerja. *Model fitting* dilakukan dengan menggunakan 15 *epoch*, yaitu jumlah iterasi yang dilakukan untuk melatih seluruh data pelatihan.

7. Evaluasi model dan perbandingan model klasifikasi

Evaluasi model dan perbandingan model klasifikasi dilakukan dengan metrik evaluasi antara lain rata-rata akurasi, rata-rata presisi, rata-rata sensitivitas, dan rata-rata spesifisitas untuk setiap kelas. Metrik-metrik ini dihitung berdasarkan *confusion matrix* yang diperoleh. Metrik evaluasi yang digunakan bertujuan untuk mengetahui seberapa baik model dapat mengklasifikasikan citra MRI otak.

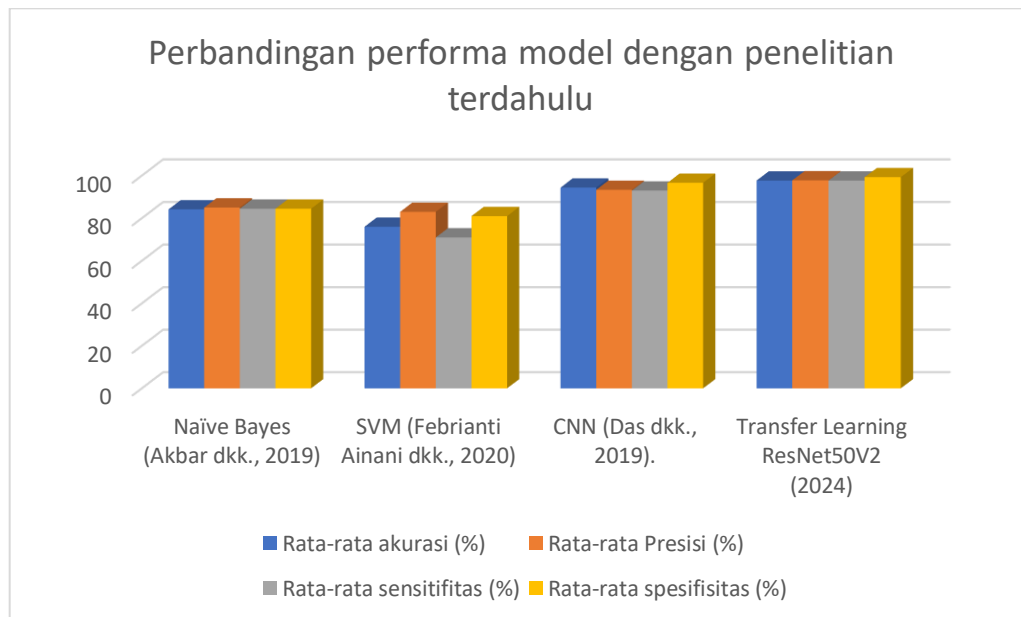
8. Perbandingan dengan penelitian sebelumnya

Pada tahap ini dilakukan perbandingan hasil pengujian dengan riset berikut:

- 1) *Analisis Performa Algoritma Naïve Bayes Pada Deteksi Otomatis Citra MRI (Akbar dkk., 2019)*.

- 2) *Klasifikasi Tumor Otak pada Citra Magnetic Resonance Image dengan Menggunakan Metode Support Vector Machine (Febrianti Ainani dkk., 2020).*
- 3) *Brain Tumor Classification Using Convolutional Neural Network (Das dkk., 2019).*

Visualisasi dari perbandingan performa model dengan penelitian terdahulu dapat dilihat pada Gambar 42.



Gambar 42. Perbandingan performa model dengan penelitian terdahulu.

Performa *ResNet50V2* ini menandakan bahwa penerapan *transfer learning* dengan model yang telah dilatih sebelumnya dapat memberikan kontribusi signifikan dalam meningkatkan kinerja klasifikasi tumor otak. Akurasi yang tinggi ini mengindikasikan kemampuan model dalam mengenali dan mengklasifikasikan variasi penyakit tumor otak pada citra MRI. Oleh karena itu, berdasarkan hasil penelitian ini, model *ResNet50V2* dapat dianggap sebagai model yang cukup baik untuk tugas klasifikasi tumor otak dengan citra MRI.

## V. PENUTUP

### 5.1 Kesimpulan

Pada penelitian ini, telah dilakukan pembangunan dan evaluasi model *transfer learning* dengan CNN untuk klasifikasi citra tumor otak. Adapun informasi yang dapat disimpulkan adalah sebagai berikut:

1. Pembangunan model klasifikasi:  
Model klasifikasi untuk klasifikasi citra tumor otak dibangun melalui beberapa langkah. Berbagai langkah seperti *pre-processing*, augmentasi, *splitting* kumpulan data, dan *hyperparameter tuning* dilakukan untuk memastikan membangun model. Model yang telah dibangun dievaluasi performa dalam mengklasifikasikan tumor otak dengan melibatkan metrik evaluasi yaitu rata-rata akurasi, rata-rata presisi, rata-rata sensitivitas, dan rata-rata spesifisitas berdasarkan *confusion matrix* untuk setiap kelas pada data pengujian. Proses ini bertujuan untuk menghasilkan model yang dapat secara optimal mengklasifikasikan citra MRI otak.
2. Perbandingan dengan model klasifikasi lainnya:  
Dibandingkan dengan penelitian terdahulu, khususnya *naïve bayes*, SVM, dan CNN, model *ResNet50V2* menunjukkan peningkatan yang signifikan dalam hal akurasi dan performa keseluruhan.
3. Performa model klasifikasi pada klasifikasi citra tumor otak:  
Secara umum, semua model yang mengadopsi *transfer learning* mencapai performa yang sangat baik dengan akurasi di atas 95%. Model *ResNet50V2* merupakan model dengan performa terbaik diantara semua model yang telah dirancang dengan rata-rata akurasi sebesar 97,7%. Model



*ResNet50V2* juga memiliki nilai rata-rata presisi, rata-rata sensitivitas, dan rata-rata spesifisitas yang tinggi, masing-masing sebesar 97,80%, 97,70%, dan 99,30%.

## 5.2 Saran

Berdasarkan hasil penelitian, beberapa saran dapat diberikan untuk peningkatan kualitas dan keberlanjutan penelitian ini:

1. Memperbanyak *model pre-trained* yang digunakan:  
Eksplorasi lebih lanjut dengan mempertimbangkan penggunaan lebih banyak *model pre-trained* untuk mendapatkan pemahaman yang lebih luas terhadap performa model.
2. Memperbaiki pola grafik akurasi dan *loss* yang belum optimal:  
Diperlukan perbaikan pada pola grafik akurasi dan *loss* yang tidak berjalan secara mulus. Hal ini dapat melibatkan peninjauan kembali *hyperparameter* model, atau penyesuaian langkah-langkah *pre-processing* untuk memastikan hasil yang lebih optimal.

Harapannya penelitian ini dapat terus berkembang dan memberikan kontribusi yang lebih besar dalam penanganan klasifikasi citra tumor otak atau citra lainnya dengan pendekatan *transfer learning*.

## DAFTAR PUSTAKA

- Akbar, F., Rais, A. N., Sobari, I. A., Zuama, R. A., dan Rudiarto, B. 2019. Analisis Performa Algoritma *Naïve Bayes* Pada Deteksi Otomatis Citra MRI. *Jurnal Ilmu Pengetahuan dan Teknologi Komputer*. **5**(1): 37-42.
- Alzubaidi, L., Zhang, J., Humaidi, A. J., Al-Dujaili, A., Duan, Y., Al-Shamma, O., Santamaría, J., Fadhel, M. A., Al-Amidie, M., dan Farhan, L. 2021. Review of Deep Learning: Concepts, CNN Architectures, Challenges, Applications, Future Directions. *Journal of Big Data*. **8**(1): 1-74.
- Asy Syifa, S., dan Amelia Dewi, I. 2022. MIND (Multimedia Artificial Intelligent Networking Database Arsitektur Resnet-152 dengan Perbandingan Optimizer Adam dan RMSProp untuk Mendeteksi Penyakit Paru-Paru. *Journal MIND Journal*. **7**(2): 139–150.
- Baiq Nurul Azmi, Arief Hermawan, dan Donny Avianto. 2023. Analisis Pengaruh Komposisi Data Training dan Data Testing pada Penggunaan PCA dan Algoritma Decision Tree untuk Klasifikasi Penderita Penyakit Liver. *JTIM : Jurnal Teknologi Informasi Dan Multimedia*. **4**(4): 281–290.
- Das, S., Aranya, O. F. M. R., dan Labiba, N. N. 2019. Brain Tumor Classification Using Convolutional Neural Network. *1st International Conference on Advances in Science, Engineering and Robotics Technology 2019 (ICASERT 2019)*. pp. 1-5.
- Elgeldawi, E., Sayed, A., Galal, A. R., dan Zaki, A. M. 2021. Hyperparameter Tuning for Machine Learning Algorithms Used for Arabic Sentiment Analysis. *Informatics*. **8**(79): <https://doi.org/10.3390/informatics8040079>
- Febrianti Ainani, Sardjono Tri, dan Babgei Atar. 2020. Klasifikasi Tumor Otak pada Citra Magnetic Resonance Image dengan Menggunakan Metode Support Vector Machine. *Jurnal Teknik ITS*. **9**(1): 118-120
- Ghozali, M., dan Sumarti, H. 2021. Jurnal Review: Pengobatan Klinis Tumor Otak pada Orang Dewasa. *Jurnal Phi Jurnal Pendidikan Fisika dan Fisika Terapan*. **2**(1): 1-14.

- He, K., Zhang, X., Ren, S., & Sun, J. 2016. Deep Residual Learning for Image Recognition. *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. pp. 770-778.
- Heranurweni, S., Destyningtias, B., dan Nugroho, A. K. 2018. Klasifikasi Pola Image pada Pasien Tumor Otak Berbasis Jaringan Syaraf Tiruan (Studi Kasus Penanganan Kuratif Pasien Tumor Otak). *Elektrika*. **10**(2): 37-40.
- Hosna, A., Merry, E., Gyalmo, J., Alom, Z., Aung, Z., dan Azim, M. A. 2022. Transfer Learning: A Friendly Introduction. *Journal of Big Data*. **9**(102). <https://doi.org/10.1186/s40537-022-00652-w>
- Hossin, M., dan Sulaiman, M. (2015). A Review on Evaluation Metrics for Data Classification Evaluations. *International Journal of Data Mining & Knowledge Management Process*. **5**(2): 1–11.
- Huang, G., Liu, Z., van der Maaten, L., & Weinberger, K. Q. 2017. Densely Connected Convolutional Networks. *In Proceedings of the IEEE conference on computer vision and pattern recognition*. pp. 4700-4708.
- Ji, Q., Huang, J., He, W., & Sun, Y. 2019. Optimized deep convolutional neural networks for identification of macular diseases from optical coherence tomography images. *Algorithms*. **12**(51). <https://doi.org/10.3390/a12030051>
- Joshi, A. V. 2020. *Machine Learning and Artificial Intelligence*. Redmond, WA, USA
- Pawar, K. 2023. *Introduction to Softmax for Neural Network*. <https://insideaiml.com/blog/SoftMaxActivation-Function-1034>. Diakses pada 14 Oktober 2023
- Kusumanto, R. D., dan Tompunu, A. N. 2011. Pengolahan Citra Digital untuk Mendeteksi Obyek Menggunakan Pengolahan Warna Model Normalisasi RGB. *Semantik*. **1**(1). ISBN 979-26-0255-0
- Markoulidakis, I., Rallis, I., Georgoulas, I., Kopsiaftis, G., Doulamis, A., dan Doulamis, N. 2021. Multiclass Confusion Matrix Reduction Method and Its Application on Net Promoter Score Classification Problem. *Technologies*. **9**(4): 412-419
- Patel, S., & Khan, N. R. 2023. A Weighted-Average-Ensembling Based Hybrid CNN Model for Improved Covid-19 Detection. *Tuijin Jishu/Journal of Propulsion Technology*. **44**(4): 6016-6033
- Qodri, K. N., Soesanti, I., dan Nugroho, H. A. 2021. Image Analysis for MRI-Based Brain Tumor Classification Using Deep Learning. *IJITEE (International Journal of Information Technology and Electrical Engineering)*. **5**(1): 21-28.

- Ramadhani, R. A., Pangestu, B. W., dan Purbaningtyas, R. 2022. Klasifikasi Tumor Otak Menggunakan Convolutional Neural Network dengan Arsitektur EfficientNet-B3. *JUST IT: Jurnal Sistem Informasi, Teknologi Informasi dan Komputer*. **12**(3): 55-59.
- Ramírez-Arias, F. J., García-Guerrero, E. E., Tlelo-Cuautle, E., Colores-Vargas, J. M., García-Canseco, E., López-Bonilla, O. R., Galindo-Aldana, G. M., dan Inzunza-González, E. 2022. Evaluation of Machine Learning Algorithms for Classification of EEG Signals. *Technologies*. **10**(4): 10-11
- Sandler, M., Howard, A., Zhu, M., Zhmoginov, A., dan Chen, L.-C. 2018. MobileNetV2: Inverted Residuals and Linear Bottlenecks. *In Proceedings of the IEEE conference on computer vision and pattern recognition*. pp. 4510-4520.
- Shah, F. A., Khan, M. A., Sharif, M., Tariq, U., Khan, A., Kadry, S., dan Thinnukool, O. 2021. A cascaded design of best features selection for fruit diseases recognition. *Computers, Materials and Continua*. **70**(1): 1491–1507.
- Sharma, S., Sharma, S., dan Athaiya, A. 2020. Activation Functions in Neural Networks. *International Journal of Engineering Applied Sciences and Technology*. **4**(12): 310-316.
- Sheard, R., Pelosi, E., dan Cancer Council Australia. 2020. Understanding Brain Tumours: A Guide for People with Brain or Spinal Cord Tumours, Their Families and Friends. <https://nla.gov.au/nla.cat-vn8544285>. Diakses pada 19 Oktober 2023.
- Sihombing, P. R., dan Yuliati, I. F. 2021. Penerapan Metode Machine Learning dalam Klasifikasi Risiko Kejadian Berat Badan Lahir Rendah di Indonesia. *MATRIK: Jurnal Manajemen, Teknik Informatika Dan Rekayasa Komputer*. **20**(2): 417-426.
- Srivastava, N., Hinton, G., Krizhevsky, A., dan Salakhutdinov, R. 2014. Dropout: A Simple Way to Prevent Neural Networks from Overfitting. *Journal of Machine Learning Research*. **15**(56): 1929–1958.
- Susanto, A. 2019. Penerapan Operasi Morfologi Matematika Citra Digital untuk Ekstraksi Area Plat Nomor Kendaraan Bermotor. *Pseudocode*. **6**(1): 49-57.
- Szegedy, C., Ioffe, S., Vanhoucke, V., & Alemi, A. 2016. Inception-v4, Inception-ResNet and the Impact of Residual Connections on Learning. *In Proceedings of the AAAI conference on artificial intelligence*. **31**(1): 1-11.
- Tantika, R. S., dan Kudus, A. 2022. Penggunaan Metode Support Vector Machine Klasifikasi Multiclass pada Data Pasien Penyakit Tiroid. *Bandung Conference Series: Statistics*. **2**(2): 159–166.

- Wang, Q., Ma, Y., Zhao, K., dan Tian, Y. 2022. A Comprehensive Survey of Loss Functions in Machine Learning. *Annals of Data Science*. **9**(2): 187–212.
- Wibawa, A. P., Guntur, M., Purnama, A., Fathony Akbar, M., dan Dwiyanto, F. A. 2018. Metode-Metode Klasifikasi. *Prosiding Seminar Ilmu Komputer dan Teknologi Informasi*. **3**(1): 134-138.
- Yueniwati, Y. 2017. *Pencitraan pada Tumor Otak: Modalitas dan Interpretasinya*. UB Press, Malang.