

**PERBANDINGAN NILAI AKURASI MODEL *MACHINE LEARNING*
ANTARA ARSITEKTUR MOBILENETV2 DAN XCEPTION PADA
SISTEM KLASIFIKASI SAMPAH**

(Skripsi)

**Oleh
RIAN KURNIAWAN
NPM 1815061018**



**FAKULTAS TEKNIK
UNIVERSITAS LAMPUNG
BANDAR LAMPUNG
2023**

ABSTRAK

PERBANDINGAN NILAI AKURASI MODEL *MACHINE LEARNING* ANTARA ARSITEKTUR MOBILENETV2 DAN XCEPTION PADA SISTEM KLASIFIKASI SAMPAH

Oleh
RIAN KURNIAWAN

Sampah yang dihasilkan setiap hari dapat menjadi masalah karena beberapa jenis sampah sulit untuk terurai sehingga dapat mencemari lingkungan. Sampah yang berpotensi dapat didaur ulang dan memiliki nilai jual adalah sampah jenis anorganik terutama sampah kardus, logam, kertas, kaca, plastik, karet, dan sampah lainnya seperti bungkus kemasan produk. Berbagai jenis sampah dapat diklasifikasi menggunakan model *machine learning*. Model *machine learning* yang digunakan untuk klasifikasi sistem sampah adalah model dengan metode *Convolutional Neural Network* (CNN). Data yang digunakan pada penelitian ini berjumlah 2647 data tipe gambar. Pemilihan arsitektur CNN mempertimbangkan akurasi dan biaya komputasi yang dibutuhkan. Penelitian ini bertujuan untuk mengetahui arsitektur, *optimizer*, dan *learning rate* terbaik pada sistem klasifikasi sampah. Model yang dirancang menggunakan arsitektur MobileNetV2 dengan *optimizer* SGD dan *learning rate* 0.1 memiliki akurasi sebesar 86.07% dan model yang dirancang menggunakan arsitektur Xception dengan *optimizer* Adam dan *learning rate* 0.001 memiliki akurasi sebesar 87.81%.

Kata Kunci: *Artificial Intelligence*, Klasifikasi, MobileNetV2, Xception

ABSTRACT

COMPARISON OF MACHINE LEARNING MODEL ACCURACY VALUE BETWEEN MOBILENETV2 AND XCEPTION ARCHITECTURE IN WASTE CLASSIFICATION SYSTEMS

**By
RIAN KURNIAWAN**

Garbage generated every day can be a problem because some types of waste are difficult to decompose so they can pollute the environment. Waste that can potentially be recycled and has a selling value is inorganic waste, especially cardboard, metal, paper, glass, plastic, rubber and other waste such as product packaging. Various types of waste can be classified using machine learning models. The machine learning model used for classification of waste systems is a model with the Convolutional Neural Network (CNN) method. The data used in this study amounted to 2647 image data types. The selection of the CNN architecture takes into account the required accuracy and computational costs. This study aims to determine the best architecture, optimizer, and learning rate in the waste classification system. The model designed using the MobileNetV2 architecture with the SGD optimizer and a learning rate of 0.1 has an accuracy of 86.07% and the model designed using the Xception architecture with the Adam optimizer and a learning rate of 0.001 has an accuracy of 87.81%.

Keywords: Artificial Intelligence, Classification, MobileNetV2, Xception

**PERBANDINGAN NILAI AKURASI MODEL *MACHINE LEARNING*
ANTARA ARSITEKTUR MOBILENETV2 DAN XCEPTION PADA
SISTEM KLASIFIKASI SAMPAH**

**Oleh
Rian Kurniawan**

Skripsi

Sebagai Salah Satu Syarat untuk Mendapat Gelar
SARJANA TEKNIK

Pada
Program Studi Teknik Informatika
Jursan Teknik Elektro
Fakultas Teknik Universitas Lampung



**FAKULTAS TEKNIK
UNIVERSITAS LAMPUNG
BANDAR LAMPUNG
2023**

Judul : **PERBANDINGAN NILAI AKURASI MODEL
MACHINE LEARNING ANTARA ARSITEKTUR
MOBILENETV2 DAN XCEPTION PADA SISTEM
KLASIFIKASI SAMPAH**
Nama Mahasiswa : **Rian Kurniawan**
Nomor Pokok Mahasiswa : **1815061018**
Program Studi : **Teknik Informatika**
Fakultas : **Teknik**

MENYETUJUI

1. Komisi Pembimbing



Yessi Mulyani, S.T., M.T.
NIP. 197312262000122001



Puput Budi Wintoro, S. Kom., M.T.I.
NIP. 198410312019031004

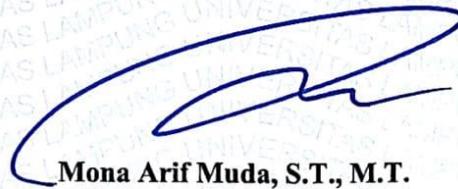
2. Mengetahui

Ketua Jurusan
Teknik Elektro



Herlinawati, S.T., M.T.
NIP. 197103141999032001

Ketua Program Studi
Teknik Informatika



Mona Arif Muda, S.T., M.T.
NIP. 197111122000031002

MENGESAHKAN

1. Tim Penguji

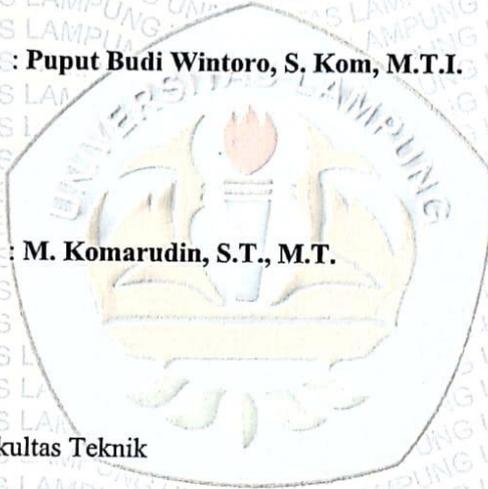
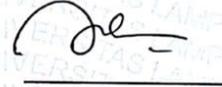
Ketua : **Yessi Mulyani, S.T., M.T.**



Sekretaris : **Puput Budi Wintoro, S. Kom, M.T.I.**



Penguji : **M. Komarudin, S.T., M.T.**



2. Dekan Fakultas Teknik



Dr. Eng. Helmy Fitriawan, S.T., M.Sc.

NIP. 197509282001121002

Tanggal Lulus Ujian Skripsi: 31 Januari 2023

SURAT PERNYATAAN

Saya yang bertandatangan dibawah ini, menyatakan bahwa skripsi saya yang berjudul “PERBANDINGAN NILAI AKURASI MODEL *MACHINE LEARNING* ANTARA ARSITEKTUR MOBILENETV2 DAN XCEPTION PADA SISTEM KLASIFIKASI SAMPAH” dengan ini menyatakan bahwa skripsi saya dibuat oleh saya sendiri. Semua hasil yang tertuang dalam skripsi ini telah mengikuti kaidah penulisan karya ilmiah Universitas Lampung. Apabila di kemudian hari terbukti bahwa skripsi ini merupakan salinan atau dibuat oleh orang lain, maka saya bersedia menerima sanksi sesuai dengan ketentuan hukum atau akademik yang berlaku.

Bandar Lampung, 21 Februari 2023

Pembuat Pernyataan,



Rian Kurniawan
NPM 1815061018

RIWAYAT HIDUP



Penulis dilahirkan di Bandar Lampung, 25 Maret 1998. Merupakan anak keempat dari lima bersaudara pasangan Bapak Hazairin Hukum dan Ibu Elvira Hamamy. Penulis telah menyelesaikan jenjang pendidikan mulai dari TK Gajah Mada Bandar Lampung pada tahun 2005, Pendidikan Sekolah Dasar (SD) Negeri 2 (Teladan) Rawa Laut Bandar Lampung pada tahun 2011, Sekolah Menengah Pertama (SMP) di SMP Negeri 4 Bandar Lampung pada tahun 2014, dan Sekolah Menengah Atas (SMA) di MA Negeri 1 Bandar Lampung tahun 2017. Penulis melanjutkan pendidikan di Fakultas Teknik Jurusan Teknik Elektro Program Studi Teknik Informatika Lampung, penulis juga aktif mengikuti kegiatan sebagai berikut:

1. Peserta Studi Independent Kampus Merdeka pada program Bangkit tahun 2021.
2. Pada bulan September 2021 penulis menyelesaikan Kuliah Kerja Nyata (KKN) di Kelurahan Kota Karang, Kecamatan Teluk Betung Timur, Kota Bandar Lampung.

PERSEMBAHAN

Segala puji syukur atas kehadiran Allah SWT berkat karunia, kesehatan, dan hidayah yang diberikan serta shalawat teriring salam kepada baginda Nabi Muhammad SAW yang kita nantikan syafa'at-nya di hari akhir kelak. Dengan segala kerendahan hati, saya persembahkan skripsi ini kepada:

Mamiku Tercinta (Elvira Hamamy)

sebagai sosok yang selalu dengan ikhlas sabar membimbingku dari kecil hingga saat ini serta doa yang selalu beliau panjatkan tak lain untuk kesuksesanku.

Papiku Terkasih (Hazairin Hukum)

sebagai sosok yang sangat ku kagumi, yang selalu memberi nasihat kepadaku, memberikanku semangat tiada henti tak lain untuk menggapai cita-citaku.

Kakak dan Adikku Tersayang (Wirathama, Anggun, Debby, Jerry)

sebagai sosok kakak dan adik yang selalu memberikan semangat sehingga penulis dapat berusaha menggapai keinginanku dan selalu memberi keceriaan dalam hari-hariku.

Serta, almamater yang saya sangat banggakan

UNIVERSITAS LAMPUNG

SANWACANA

Segala puji hanya bagi Allah SWT, Tuhan semesta alam yang Maha Pengasih lagi Maha Penyayang, atas limpahan rahmat, taufik serta hidayah-Nya sehingga penulis dapat menyelesaikan penulisan skripsi yang berjudul: PERBANDINGAN NILAI AKURASI MODEL *MACHINE LEARNING* ANTARA ARSITEKTUR MOBILENETV2 DAN XCEPTION PADA SISTEM KLASIFIKASI SAMPAH. Sebagai salah satu syarat untuk memperoleh gelar Sarjana Teknik pada Fakultas Teknik Universitas Lampung. Dalam penyelesaian skripsi ini, penulis mendapatkan banyak bantuan, bimbingan, saran dan dukungan dari segenap pihak, baik secara langsung maupun tidak langsung sehingga penyusunan skripsi ini berjalan dengan baik. Maka pada kesempatan ini penulis mengucapkan terima kasih kepada:

1. Bapak Dr. Eng. Helmy Fitriawan, S.T., M.Sc. selaku Dekan Fakultas Teknik Universitas Lampung.
2. Ibu Herlinawati, S.T.,M.T. selaku Ketua Jurusan Teknik Elektro Universitas Lampung.
3. Bapak Mona Arif Muda, S.T., M.T. selaku Ketua Program Studi Teknik Informatika Jurusan Teknik Elektro Universitas Lampung.
4. Ibu Yessi Mulyani, S.T., M.T. selaku Pembimbing Utama, yang telah bersedia meluangkan waktu untuk memberikan pengarahan dan bimbingan dalam pembuatan skripsi ini hingga selesai.
5. Bapak Puput Budi Wintoro, S. Kom, M.T.I. selaku Pembimbing Kedua, yang telah bersedia meluangkan waktu untuk memberikan pengarahan dan bimbingan dalam pembuatan skripsi ini hingga selesai.
6. Bapak M. Komarudin, S.T., M.T. selaku Pembimbing Akademik yang telah bersedia meluangkan waktu untuk memberikan pengarahan dan bimbingan selama menempuh pendidikan Program Studi Teknik Informatika Universitas Lampung.

7. Seluruh Dosen Program Studi Teknik Informatika yang telah membagikan ilmunya kepada penulis.
8. Seluruh teman teman Teknik Informatika Angkatan 2018 selaku teman kelas.

Semoga Allah SWT membalas segala bentuk kebaikan hati dan jasa yang telah kalian berikan kepada saya. Saya menyadari meskipun skripsi ini sudah disusun dengan sebaik mungkin, skripsi ini masih terdapat kekurangan dan masih jauh dari kata sempurna, namun saya sangat berharap melalui skripsi ini akan memberikan manfaat bagi siapapun yang membacanya dan bagi penulis dalam mengembangkan dan mengamalkan ilmu pengetahuan yang telah ditempuh selama ini.

Bandar Lampung, 21 Februari 2023
Penulis,



Rian Kurniawan
NPM 1815061018

DAFTAR ISI

	Halaman
DAFTAR ISI	iv
DAFTAR TABEL	vi
DAFTAR GAMBAR	vii
BAB I PENDAHULUAN	1
1.1 Latar Belakang	1
1.2 Rumusan Masalah	2
1.3 Batasan Masalah.....	3
1.4 Tujuan Penelitian.....	3
1.5 Manfaat Penelitian.....	3
1.6 Sistematika Penulisan.....	3
BAB II TINJAUAN PUSTAKA	5
2.1 Kecerdasan Buatan	5
2.2 <i>Machine Learning</i>	5
2.3 Jaringan Saraf Tiruan	6
2.3.1 <i>Perceptron</i>	8
2.3.2 Fungsi Aktivasi	8
2.3.3 Jenis Arsitektur Jaringan Saraf Tiruan.....	12
2.3.4. <i>Backpropagation</i>	14
2.4 <i>Convolutional Neural Network</i>	16
2.4.1 <i>Convolutional Layer</i>	17
2.4.2 <i>Max Pooling dan Average Pooling</i>	18
2.4.3 <i>Global Average Pooling</i>	19
2.4.4 <i>Fully Connected Layer</i>	20
2.4.5 <i>Optimizer</i>	20

2.4.6	<i>Loss Function</i>	21
2.5	Arsitektur MobileNetV2.....	22
2.6	Xception	24
2.7	Pembagian Data.....	25
2.8	Augmentasi Data	26
2.9	Google Colab.....	26
2.10	TensorFlow	27
2.11	Keras.....	27
2.11.1	<i>Library Keras</i>	27
2.12	Penelitian Terkait	30
BAB III METODOLOGI PENELITIAN		34
3.1	Waktu dan Tempat Penelitian	34
3.2	Jadwal Penelitian	34
3.3	Alat Dalam Penelitian	35
3.3.1	Alat Penelitian.....	35
3.3.2	Bahan Penelitian.....	36
3.4	Kerangka Penelitian	36
3.4.1	Pengumpulan Data	38
3.4.2	Pengolahan Data.....	38
3.4.3	Perancangan Model.....	39
3.4.4	Penyusunan Model	40
3.4.5	Pelatihan Model	41
3.4.6	Evaluasi Model.....	41
3.4.7	Perbandingan Model	42
BAB V KESIMPULAN DAN SARAN		55
5.1	Kesimpulan.....	55
5.2	Saran.....	55
DAFTAR PUSTAKA		56

DAFTAR TABEL

	Halaman
Tabel 1. Penelitian Terkait	30
Tabel 2. Jadwal Penelitian.....	34
Tabel 3. Alat Penelitian.....	35

DAFTAR GAMBAR

	Halaman
Gambar 1. Model Jaringan Saraf Tiruan.....	7
Gambar 2. Perceptron.....	8
Gambar 3. Sigmoid	9
Gambar 4. Tanh.....	10
Gambar 5. ReLU	10
Gambar 6. Softmax	11
Gambar 7. Lapisan Tunggal.....	12
Gambar 8. Jaringan Lapisan Jamak	13
Gambar 9. Jaringan Lapisan Kompetitif	13
Gambar 10. Arsitektur CNN	17
Gambar 11. Operasi Konvolusi.....	18
Gambar 12. Max Pooling dan Average Pooling	19
Gambar 13. Global Average Pooling	19
Gambar 14. Connected Layer	20
Gambar 15. Arsitektur MobileNetV2	22
Gambar 16. Linear Bottleneck dan Shortcut Connection di antara Bottlenecks...	23
Gambar 17. Arsitektur Xception[30]	24
Gambar 18. Kerangka Penelitian	37
Gambar 19. Pengumpulan Data	38

BAB I PENDAHULUAN

1.1 Latar Belakang

Satu-satunya Universitas di Indonesia yang memiliki Sistem Pengelolaan Sampah Terintegrasi adalah Universitas Lampung. Faktanya, di kampus-kampus lain masih belum ada unit khusus yang menangani pengelolaan sampah. Penegakan peraturan dan pengawasan belum dilaksanakan secara ketat, serta belum ada sistem pengelolaan sampah terpadu. Universitas Lampung menghasilkan sampah sekitar 10 metrik ton per hari yang dimana sampah ini sebenarnya dapat berpotensi menghasilkan 10 milyar kalori per hari[1].

Sampah yang dihasilkan setiap hari dapat menjadi masalah karena beberapa jenis sampah sulit untuk terurai sehingga dapat mencemari lingkungan.[2] Sampah terdiri dari dua jenis yaitu sampah organik dan sampah anorganik. Sampah organik adalah sampah yang berasal dari sisa-sisa organisme hidup, sedangkan sampah anorganik berasal dari organisme yang tidak hidup. Sampah yang berpotensi dapat didaur ulang dan memiliki nilai jual adalah sampah jenis anorganik terutama sampah kardus, logam, kertas, kaca, plastik, karet, dan sampah lainnya seperti bungkus kemasan produk[3].

Berbagai jenis sampah dapat diklasifikasi menggunakan model *machine learning*. Model *machine learning* yang digunakan untuk klasifikasi sistem sampah adalah model dengan metode *Convolutional Neural Network*. Model dengan metode ini akan mengenali citra sampah dengan cara mengekstrak fitur citra dan mengenali polanya sesuai dengan label yang ada pada data latih[4].

Dalam beberapa dekade terakhir, Deep Learning telah menjadi tools powerfull. Hal ini dibuktikan dengan kemampuannya menangani data dalam jumlah besar serta mampu mengenali pola dari data yang dikelola. Salah satu algoritma yang populer dalam menangani jumlah data besar yaitu Convolutional Neural Network[5].

Arsitektur *Convolutional Neural Network* yang digunakan untuk merancang model *machine learning* memegang peran penting. Semakin tepat pemilihan arsitektur *Convolutional Neural Network* maka akan semakin baik akurasi model yang dibuat untuk memprediksi citra sampah. Selain akurasi yang baik, setiap arsitektur memiliki ukuran (*size*), parameter, dan *cost* (CPU/GPU) yang berbeda-beda. Arsitektur Xception merupakan arsitektur yang memiliki akurasi tinggi, ukuran (*size*) yang kecil, dan jumlah parameter yang lebih sedikit dari beberapa arsitektur lainnya, sehingga model yang dilatih menggunakan arsitektur ini akan efektif dan efisien digunakan untuk memprediksi citra. Arsitektur MobileNetV2 merupakan arsitektur yang memiliki akurasi tinggi dengan ukuran (*size*) yang paling kecil, dan parameter yang paling sedikit dibandingkan arsitektur lainnya, sehingga model yang dilatih menggunakan arsitektur ini akan efisien dalam segi waktu pelatihan dan dapat memprediksi citra dengan cukup baik[6].

1.2 Rumusan Masalah

Adapun rumusan masalah dari skripsi ini adalah :

Bagaimana perbandingan nilai akurasi model *machine learning* yang dibuat menggunakan arsitektur MobileNetV2 dan Xception dengan pengaturan *learning rate* 0.1, 0.01, dan 0.001 serta *optimizer Adaptive Moment Estimation* (Adam), *Stochastic Gradient Descent* (SGD), dan *Root Mean Square Propagation* (RMSprop).

1.3 Batasan Masalah

Batasan masalah pada penelitian ini adalah :

1. Dataset yang digunakan untuk membuat model *machine learning* diperoleh dari Kaggle, dataset kemudian digabungkan dengan data yang diperoleh secara mandiri.
2. Dataset yang digunakan terdiri dari enam kelas yaitu *cardboard*, *metal*, *paper*, *glass*, *plastic*, dan *trash*.
3. Jenis sampah yang dapat diklasifikasi yaitu *cardboard*, *metal*, *paper*, *glass*, *plastic*, dan *trash*.
4. Arsitektur CNN yang digunakan untuk membuat model machine learning yaitu arsitektur MobileNetV2 dan Xception dengan *optimizer* Adam, SGD, dan RMSProp serta *learning rate* sebesar 0.001, 0.01, dan 0.1.

1.4 Tujuan Penelitian

Tujuan dari penelitian ini yaitu untuk mengetahui arsitektur CNN, *optimizer*, dan *learning rate* yang memiliki nilai akurasi optimum melalui studi perbandingan antara arsitektur MobileNetV2 dan Xception pada sistem klasifikasi sampah.

1.5 Manfaat Penelitian

Dengan dilakukannya penelitian ini, diharapkan dapat memberi manfaat yaitu untuk memahami arsitektur *Convolutional Neural Network*, *optimizer*, dan *learning rate* yang terbaik pada sistem klasifikasi sampah.

1.6 Sistematika Penulisan

Sistematika penulisan skripsi ini antara lain :

BAB I PENDAHULUAN

Bab ini merupakan pendahuluan berisi tentang latar belakang, tujuan, rumusan masalah, batasan masalah, manfaat penelitian dan sistematika penulisan.

BAB II TINJAUAN PUSTAKA

Bab ini merupakan tinjauan Pustaka berisi definisi maupun istilah yang diambil dari berbagai sumber (buku, jurnal, dan sebagainya). BAB II berisi tentang pengertian Jaringan Saraf Tiruan, *Convolutional Neural Network*, dan Arsitektur *Convolutional Neural Network* yang dipakai seperti arsitektur MobileNetV2 dan Xception.

BAB III METODOLOGI PENELITIAN

Bab ini merupakan metodologi penelitian berisi tempat dan waktu penelitian, jadwal penelitian, alat dan bahan penelitian.

BAB IV HASIL DAN PEMBAHASAN

Bab ini merupakan hasil dan pembahasan berisi tentang hasil penelitian yang dilakukan serta pengolahan data dari hasil penelitian.

BAB V KESIMPULAN

Bab ini merupakan kesimpulan berisi kesimpulan dan saran berdasarkan penelitian.

BAB II **TINJAUAN PUSTAKA**

2.1 Kecerdasan Buatan

Kecerdasan Buatan atau *Artificial Intelligence* (AI) adalah simulasi kecerdasan manusia yang dimodelkan ke dalam mesin dengan cara memprogramnya sehingga mesin tersebut dapat berpikir layaknya manusia. Dengan kata lain Kecerdasan Buatan merupakan sistem komputer yang dapat melakukan pekerjaan-pekerjaan yang biasa dikerjakan menggunakan kecerdasan manusia.

Sama seperti manusia, Kecerdasan Buatan memerlukan data untuk dijadikan pengetahuan. Data ini digunakan agar kecerdasannya menjadi lebih baik dari sebelumnya. Kecerdasan buatan belajar dengan cara *self-correction* atau mengkoreksi diri dari kesalahan yang dibuatnya hingga mampu memenuhi targetnya[7].

2.2 *Machine Learning*

Pada mulanya, *machine learning* merupakan istilah yang digunakan untuk menyebut sebuah cabang ilmu komputer yang mempelajari metode perancangan algoritma yang mampu belajar atau beradaptasi terhadap pola data tanpa diprogram secara eksplisit. *Machine learning* memiliki beberapa metode komputasi yang dapat meningkatkan kinerja dengan memanfaatkan pengetahuan yang berasal dari pengalaman ketika belajar[8].

Terdapat beberapa metode algoritma pada *machine learning* di antaranya sebagai berikut:

1. *Supervised machine learning algorithms*

Metode algoritma ini memberikan label pada data dan menerapkan informasi sebelum melakukan pembelajaran. Algoritma ini membandingkan pengalaman belajar di masa lalu dengan pengalaman belajar di masa sekarang.

2. *Unsupervised machine learning algorithms*

Metode algoritma ini tidak memberikan label pada data dan tidak menerapkan informasi sebelum melakukan pembelajaran. Algoritma ini akan menemukan struktur tersembunyi pada data yang tidak diberikan label.

3. *Semi-supervised machine learning algorithms*

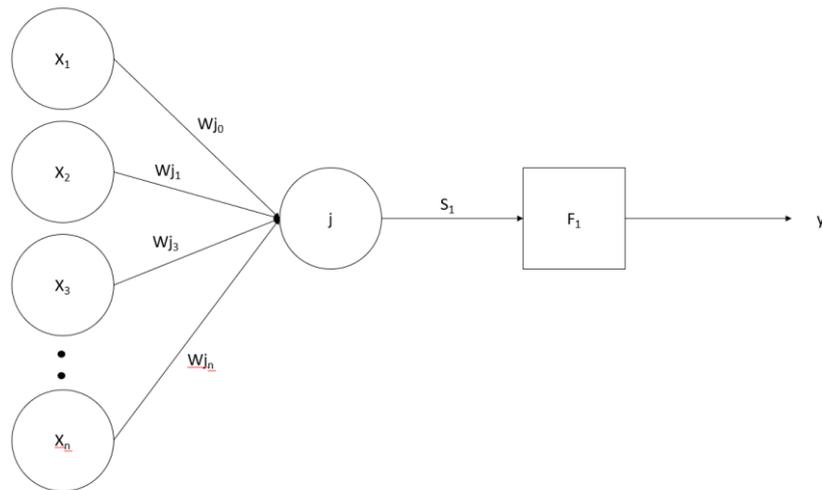
Metode algoritma ini memberikan label pada sebagian data dan membiarkan sebagian data tanpa label. Algoritma ini dapat meningkatkan efisiensi *output* yang dihasilkan.

4. *Reinforcement machine learning algorithms*

Metode algoritma ini akan memberikan poin (*reward*) ketika model menjadi semakin baik dan akan memberikan poin (*error*) ketika model menjadi semakin buruk[9].

2.3 Jaringan Saraf Tiruan

Otak manusia mampu mengenali pola dari informasi yang didapatkan oleh panca indera dengan akurasi yang sangat tinggi. Kemampuan pengenalan pola ini didapatkan melalui proses belajar yang kompleks. Jaringan saraf tiruan adalah sebuah metode yang digunakan untuk meniru cara belajar manusia berdasarkan arsitektur saraf otak dan kemudian menerapkannya ke dalam perangkat lunak komputer[10].



Gambar 1. Model Jaringan Saraf Tiruan

Model sel saraf umumnya terdiri atas:

1. Masukan (x) yang berfungsi sebagai penerima sinyal
2. Bobot koneksi (w_{ji}) yang berfungsi sebagai penyimpan informasi
3. Bias (w_0) yang berfungsi sebagai pengatur daerah nilai ambang
4. Elemen pemroses (Σ) dan fungsi aktivasi (f) yang berfungsi sebagai pemroses informasi
5. Keluaran (y_j) sebagai keluaran yang akan menyampaikan hasil pemrosesan informasi ke sel berikutnya[11].

Persamaan umum matematika dari Gambar 1. yaitu sebagai berikut:

$$S_j(x) = \sum_{i=1}^n w_{ji} x_i + w_0 x_0$$

$$y_j(x) = f(s_j(x))$$

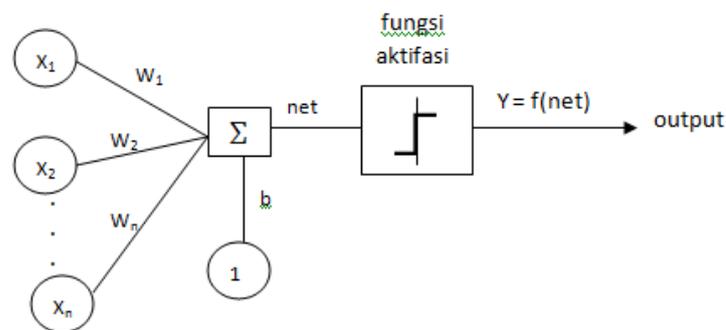
Jika $s_j > 0$, maka $y_j = 1$; Jika $s_j < 0$, maka $y_j = 0$; y_j adalah harga keluaran elemen pemroses j .

w_i = bobot masukan yang diberikan untuk menentukan tingkat aktivitas neuron.

x_i = sejumlah n masukan yang dimasukkan ke jaringan saraf [11].

2.3.1 Perceptron

Perceptron merupakan metode pelatihan jaringan saraf terbimbing (*supervised learning*). *Perceptron* merupakan jaringan sederhana yang digunakan untuk mengklasifikasikan suatu pola tertentu yang dikenal dengan pemisahan secara linear. Pada dasarnya perceptron dengan satu lapisan pada jaringan saraf memiliki bobot yang dapat diatur. Jaringan saraf memiliki sejumlah *neuron* dan masukan.



Gambar 2. *Perceptron*

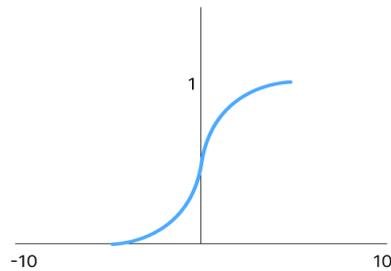
Berdasarkan Gambar 2 diatas dapat dilihat bahwa *perceptron* bekerja dengan menjumlahkan hasil kali masukan X_1 hingga X_n dengan bobotnya yaitu W_1 hingga W_n . Kemudian seluruh hasil dari operasi sebelumnya yang dilambangkan oleh Σ dimasukkan ke fungsi aktivasi untuk menghasilkan keluaran. Fungsi aktivasi *Rosenblatt* menggunakan *hard limiter* yang dapat membatasi keluaran pada nilai 1 atau 0[12].

2.3.2 Fungsi Aktivasi

Fungsi aktivasi merupakan mekanisme yang digunakan untuk menjumlahkan sinyal masukan dan menentukan apakah nilai masukan tersebut sudah mencapai suatu ambang batas yang dapat memicu munculnya nilai keluaran. Adapun beberapa fungsi aktivasi yang sering dipakai yaitu sebagai berikut:

1. Sigmoid

Fungsi aktivasi *Sigmoid* merupakan fungsi aktivasi non-linear bernilai positif seperti dapat dilihat pada Gambar 3 berikut.



Gambar 3. Sigmoid

Pada fungsi aktivasi ini, *input* dapat berupa sembarang angka dengan *output* berupa angka antara 0 dan 1. Fungsi aktivasi *sigmoid* digunakan untuk memprediksi keluaran berbasis probabilitas (antara 0 dan 1) dan digunakan pada klasifikasi citra [13].

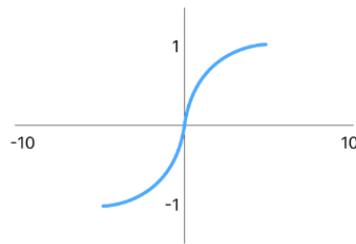
$$f(x) = \frac{1}{1 + e^{-x}}$$

x = neuron input

e = konstanta *euler* (2,71828)

2. Tanh

Fungsi aktivasi Tanh merupakan fungsi aktivasi non-linear bernilai positif dan negatif seperti dapat dilihat pada Gambar 4 berikut.

Gambar 4. *Tanh*

Pada fungsi aktivasi ini, *input* dapat berupa sembarang angka dengan *output* berupa angka antara -1 dan 1. digunakan untuk memprediksi keluaran berbasis probabilitas (antara -1 dan 1) yang berfokus pada nilai nol (*zero-centered*).

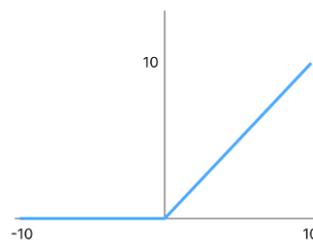
$$f(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}}$$

x = *neuron input*

e = konstanta *euler* (2,71828)

3. *ReLU (Rectified Linear Unit)*

Fungsi aktivasi *Rectified Linear Unit* (ReLU) merupakan fungsi aktivasi linear bernilai positif seperti dapat dilihat pada Gambar 5 berikut.

Gambar 5. *ReLU*

Pada fungsi aktivasi ini, *input* dapat berupa sembarang angka dengan *output* berupa angka dari *input* kecuali angka negatif diubah menjadi nol. Perubahan dari negatif ke nol ini terjadi karena saat operasi konvolusi, fitur citra yang diekstraksi memiliki

nilai piksel negatif sehingga perlu diaktivasi agar komputer dapat mengenali piksel yang dihasilkan dari operasi konvolusi.

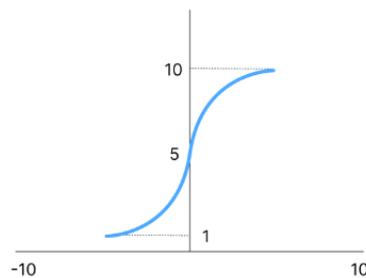
$$f(x) = \max(0, x)$$

$x = \text{neuron input}$

max = fungsi untuk merubah nilai negatif menjadi 0 dan mempertahankan nilai positif dengan nilai aslinya (x)[4]

4. *Softmax*

Fungsi aktivasi *softmax* digunakan pada lapisan *output* yang digunakan sebagai prediksi. Adapun grafik fungsi aktivasi ini dapat dilihat pada gambar berikut.



Gambar 6. *Softmax*

Adapun persamaan fungsi aktivasi ini adalah sebagai berikut:

$$p_i = \frac{e^{x_i + \log(C)}}{\sum_{k=1}^N e^{x_k + \log(C)}}$$

p_i = probabilitas pada *neuron* ke- i

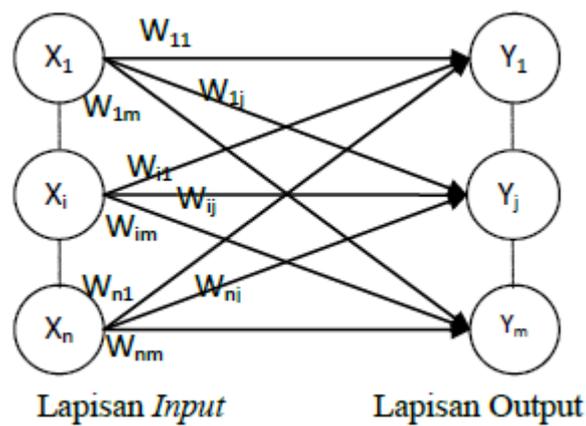
x_i = nilai *strength* pada *neuron* ke- i

C = nilai konstan yang dapat ditentukan [14]

2.3.3 Jenis Arsitektur Jaringan Saraf Tiruan

Jaringan saraf tiruan merupakan algoritma yang dapat memetakan data yang masuk ke lapisan masukan menuju target pada lapisan keluaran dengan melewati *neuron-neuron* pada lapisan tersembunyi. Terdapat beberapa jenis arsitektur jaringan saraf tiruan yaitu lapisan tunggal (*single layer*), lapisan jamak (*multi layer*), dan lapisan kompetitif[15].

1. Jaringan Lapisan Tunggal

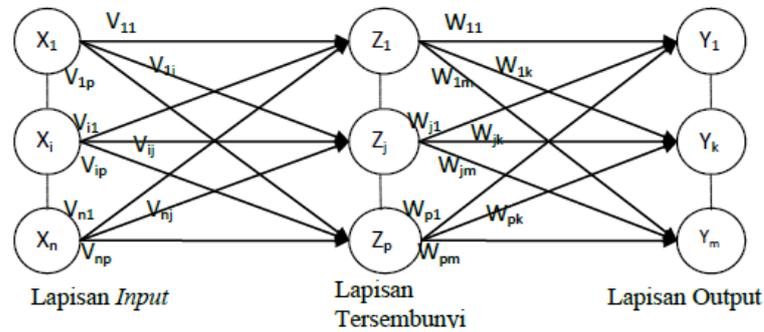


Gambar 7. Lapisan Tunggal

(sumber: <https://www.kajianpustaka.com/2016/11/jaringan-saraf-tiruan-jst.html>)

Jaringan lapisan tunggal merupakan jaringan saraf tiruan yang hanya memiliki satu lapisan. Informasi yang masuk melalui lapisan *input* akan diproyeksikan langsung menuju lapisan *output* tetapi tidak berlaku sebaliknya atau dapat dikatakan dengan istilah *feed forward*[16].

2. Jaringan Lapisan Jamak

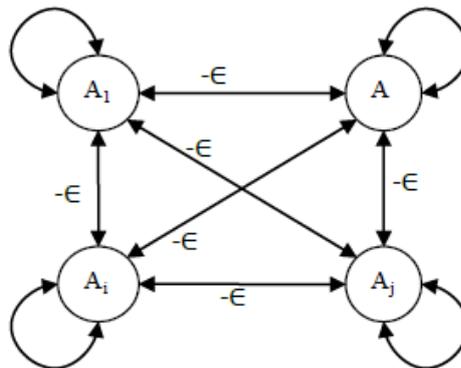


Gambar 8. Jaringan Lapisan Jamak

(sumber: <https://www.kajianpustaka.com/2016/11/jaringan-saraf-tiruan-jst.html>)

Jaringan lapisan jamak merupakan jaringan dengan adanya satu atau lebih lapisan tersembunyi (*hidden layer*). Jaringan ini dapat memecahkan masalah yang lebih banyak dibandingkan dengan lapisan tunggal tetapi membutuhkan pelatihan yang lebih rumit[16].

3. Jaringan Lapisan Kompetitif



Gambar 9. Jaringan Lapisan Kompetitif

(sumber: <https://www.kajianpustaka.com/2016/11/jaringan-saraf-tiruan-jst.html>)

Pada jaringan ini *neuron* melakukan interkoneksi yang tidak dilakukan pada jenis arsitektur lainnya. *Neuron-neuron* pada arsitektur ini akan bersaing untuk

mendapatkan hak menjadi aktif atau yang diistilahkan dengan sebutan *winner takes all*[16].

2.3.4. *Backpropagation*

Backpropagation adalah metode pelatihan jaringan saraf tiruan yang diawasi (*supervised learning*). Metode ini mengevaluasi kesalahan setiap *neuron* setelah satu set data dilatih. Tujuan metode ini yaitu untuk memperbaiki bobot jaringan dengan cara memperbaiki nilai kesalahan (*error*). Nilai ini dihitung menggunakan metode penurunan gradien (*gradient descent*) yaitu mengurangi nilai bobot suatu *neuron* lama dengan turunan *error* dari suatu *neuron output*-nya. Adapun langkah-langkah metode ini yaitu sebagai berikut:

1. Ambil satu set pelatihan yang terdiri dari *neuron input*, bobot antar *neuron*, dan nilai *target*.
2. Definisikan nilai *neuron input*, bobot antar *neuron*, nilai target, dan *learning rate*.
3. Jumlahkan seluruh *neuron* dan bias yang ada pada jaringan.
4. Aktivasi *neuron* yang sudah dijumlahkan menggunakan fungsi aktivasi, sehingga nilai *neuron output* ditemukan.
5. Hitung *error* menggunakan nilai yang ada dengan cara mengurangkan nilai target dengan nilai *neuron output*.
6. Membagi hasil pengurangan sebelumnya dengan jumlah *neuron* yang ada pada *output*, maka nilai *error* didapatkan.
7. Lakukan *backpropagation* dengan cara memperbarui bobot baru yang diperoleh dari hasil pengurangan bobot lama dengan turunan *error* pada *output*.
8. Proses dilanjutkan hingga *error* berkurang secara signifikan dan tidak mengalami penurunan.

Misalkan, ditentukan nilai *input* yaitu $x_1 = 1$, $x_2 = 2$, $x_3 = 3$, nilai target yaitu $t_1 = 0$, $t_1 = 1$, nilai bobot yaitu $w_{11} = 0.4$, $w_{12} = 0.7$, $w_{21} = 0.2$, $w_{22} = 0.5$, $w_{31} = 0.6$, $w_{32} = 0.3$, bias yaitu $b_1 = 1$, $b_2 = 1$, dan *learning rate* = 0.1. Selanjutnya, mencari nilai z_1 dan z_2 yang merupakan nilai penjumlahan seluruh bobot dan bias yang menuju lapisan *output* y_1 dan y_2 , sehingga didapat nilai $z_1 = 3.6$ dan $z_2 = 4.2$. Setelah

mendapatkan nilai z_1 dan z_2 selanjutnya melakukan aktivasi menggunakan fungsi *sigmoid* untuk mendapatkan nilai *output* y_1 dan y_2 , maka nilai *output* yang didapatkan yaitu $y_1 = 0.9737$ dan $y_2 = 0.9852$. Kemudian menghitung nilai *error* dari *output* terhadap nilai target menggunakan *mean square error* yang dimana persamaannya yaitu sebagai berikut:

$$Error (E) = \frac{1}{n} \sum_{i=1}^n (t_i - y_i)^2$$

Setelah memasukan nilai *output* y_1, y_2 dan nilai target t_1, t_2 pada persamaan tersebut, maka diperoleh nilai *error* sebesar 0.47416494087. Kemudian melakukan *backpropagation* untuk memperbarui nilai bobot:

$$w_{\text{new}} = w_{\text{old}} - \alpha \frac{\partial E}{\partial w}$$

Memperbarui nilai bobot baru dapat dilakukan dengan cara mencari turunan *error* terhadap suatu bobot yang dihitung misalkan w_{11} . Persamaan tersebut dapat dituliskan sebagai berikut:

$$\frac{\partial E}{\partial w_{11}} = \left(\frac{\partial E}{\partial y_1} \right) \cdot \left(\frac{\partial y_1}{\partial w_{11}} \right)$$

Dari persamaan tersebut maka kita memerlukan nilai turunan *error* terhadap turunan y_1 sehingga persamaan yang didapat yaitu:

$$\frac{\partial E}{\partial y_1} = 2 \times \frac{1}{2} ((t_1 - y_1)^{2-1} (-1))$$

Dari turunan tersebut maka diperoleh nilai $-(t_1 - y_1)$. Kemudian melanjutkan *backpropagation* untuk mencari nilai z_1 dan z_2 dengan cara mencari turunan y_1 terhadap bobot w_{11} sehingga persamaannya dapat ditulis sebagai berikut:

$$\frac{\partial y_1}{\partial w_{11}} = \left(\frac{\partial y_1}{\partial z_1} \right) \cdot \left(\frac{\partial z_1}{\partial w_{11}} \right)$$

Dari persamaan tersebut, maka diperoleh nilai turunan y_1 terhadap z_1 yaitu $y_1(1-y_1)$ serta turunan z_1 terhadap bobot w_{11} yaitu x_1 . Selanjutnya, menggabungkan seluruh nilai yang diperoleh ke persamaan berikut:

$$\frac{\partial E}{\partial w_{11}} = \left(\frac{\partial E}{\partial y_1} \right) \cdot \left(\frac{\partial y_1}{\partial w_{11}} \right)$$

Setelah memasukan seluruh nilai kedalam persamaan tersebut maka diperoleh nilai turunan *error* terhadap bobot w_{11} yaitu 0.02493481144. Setelah didapat nilai tersebut selanjutnya memasukkannya ke persamaan berikut:

$$w_{\text{new}} = w_{\text{old}} - \alpha \frac{\partial E}{\partial w}$$

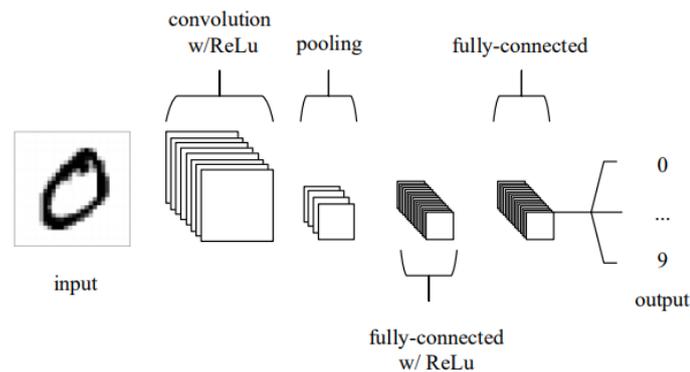
Setelah memasukan nilai ke persamaan tersebut maka diperoleh nilai bobot w_{11} baru yaitu 0.3975. Kemudian dengan cara yang sama maka nilai bobot baru yang didapat yaitu $w_{12} = 0.698$, $w_{21} = 0.195$, $w_{22} = 0.496$, $w_{31} = 0.523$, $w_{32} = 0.294$ dan nilai bias baru yaitu $b_1 = 0.9975$, $b_2 = 0.9998$. Kemudian menghitung kembali nilai *error* menggunakan bobot dan bias baru sehingga diperoleh nilai *error* baru yaitu 0.465977. Dapat dilihat bahwa nilai *error* mengalami penurunan setelah melakukan *backpropagation*[17].

2.4 Convolutional Neural Network

Convolutional Neural Network merupakan salah satu algoritma *deep learning* yang digunakan untuk melakukan klasifikasi citra. Algoritma ini merupakan jenis *Multi Layer Perceptron* yang terinspirasi dari jaringan saraf biologis manusia. *Layer* pada *Convolutional Neural Network* terdiri dari tinggi (*height*), lebar (*width*), dan kedalaman (*depth*). Lebar dan tinggi merupakan ukuran *layer* sedangkan kedalaman mengacu pada banyaknya *layer*[18].

CNN beroperasi menggunakan konvolusi dan setidaknya menggunakan satu lapisan. Jaringan saraf tiruan pada umumnya dapat mengubah nilai *input* dengan

memasukkan ke dalam rangkaian lapisan tersembunyi. Setiap *layer* dapat terdiri dari sekumpulan *neuron*, dimana setiap lapisan akan berhubungan penuh dengan lapisan sebelumnya. Terakhir, lapisan-lapisan sebelumnya akan terhubung sepenuhnya untuk menghasilkan prediksi[19].



Gambar 10. Arsitektur CNN

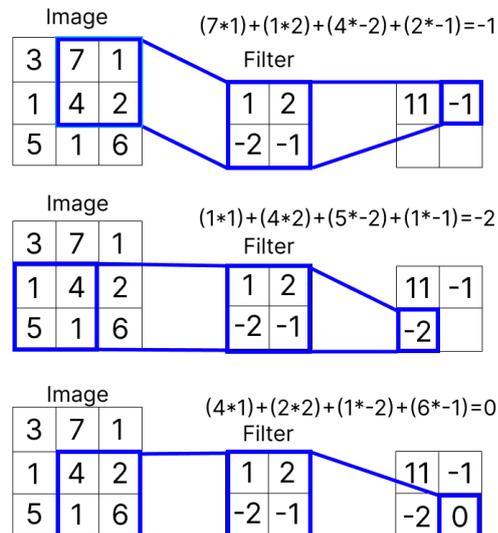
(sumber: https://www.researchgate.net/figure/An-simple-CNN-architecture-comprised-of-just-five-layers_fig4_285164623)

Berdasarkan Gambar 10. Arsitektur CNN di atas, terdapat tiga jenis lapisan pada *convolutional neural network* di antaranya yaitu *convolutional*, *pooling*, dan *fully-connected* yang mana dari masing-masing jenis lapisan tersebut memiliki fungsi yang berbeda sesuai dengan prosesnya. Pada lapisan *convolutional*, terdapat proses mengekstraksi fitur citra dengan melakukan operasi konvolusi menggunakan filter. Kemudian pada lapisan *pooling* dilakukan pengecilan dimensi dari fitur yang sudah diekstraksi sebelumnya. Pada lapisan *fully-connected*, citra yang telah diperkecil pada lapisan *pooling* diubah menjadi satu dimensi dan diklasifikasikan sesuai dengan label yang ada pada data latih[4].

2.4.1 Convolutional Layer

Convolutional Layer atau lapisan konvolusi merupakan lapisan yang di dalamnya terdapat operasi ekstraksi esensi fitur. Operasi ekstraksi esensi fitur adalah *dot product* antara bobot yang ada pada filter dengan piksel citra yang akan difilter dan

kemudian akan dilakukan penjumlahan setelahnya seperti pada Gambar 11. Operasi Konvolusi[4].

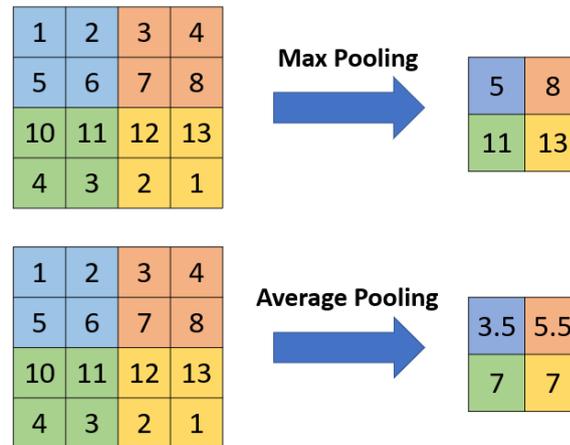


Gambar 11. Operasi Konvolusi

2.4.2 Max Pooling dan Average Pooling

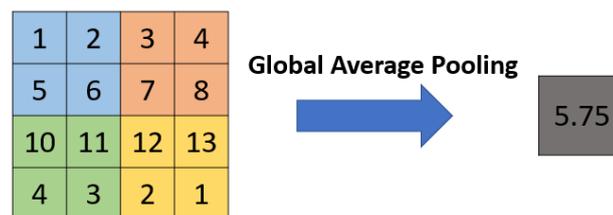
Max pooling adalah lapisan pada jaringan saraf tiruan yang berfungsi untuk mencari nilai maksimum pada kernel matriks *pooling*, sedangkan *average pooling* mencari nilai rata-rata pada kernel matriks *pooling*. Gambar 12 menjelaskan perbedaan antara *Max pooling* dan *Average Pooling*. Pada gambar tersebut terdapat citra berukuran 4x4 piksel. Nilai setiap masing-masing piksel pada gambar berbeda. Ukuran filter yang digunakan yaitu 2 piksel. *Stride* digunakan bernilai 2 yang berarti pergeseran filter dilakukan sebanyak 2 piksel. Tahap pertama yaitu pada bagian filter berwarna biru dilakukan operasi *max pooling*, sehingga nilai yang diambil yaitu 6 merupakan nilai maksimum pada filter tersebut dan menempatkannya di bagian sebelah kanan seperti yang terlihat pada gambar. Kemudian filter bergeser ke kanan sebanyak 2 piksel ke bagian berwarna hijau dan mengambil nilai maksimum yaitu 8, kemudian diletakan di bagian sebelah kanan seperti pada gambar. Kemudian filter akan bergeser ke kanan dan ke bawah hingga seluruh piksel pada citra terlewati dan memfilternya. *Average pooling* akan

menggunakan cara yang sama dalam memfilter citra, tetapi hanya nilai rata-rata filternya yang akan diambil.[20].



Gambar 12. *Max Pooling* dan *Average Pooling*

2.4.3 *Global Average Pooling*

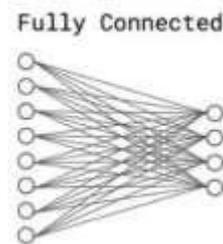


Gambar 13. *Global Average Pooling*

Global Average Pooling adalah operasi pooling yang digunakan untuk menggantikan operasi klasik CNN yang menggunakan lapisan *fully-connected*. Ide operasi ini yaitu untuk membuat satu fitur map untuk melakukan klasifikasi terhadap kategori yang ada pada *multi layer perceptron*. Operasi ini akan mengambil nilai rata-rata dari setiap fitur dan memasukannya ke lapisan *softmax* atau prediksi[21].

2.4.4 *Fully Connected Layer*

Fully Connected Layer merupakan model tradisional jaringan saraf tiruan yang berfungsi untuk mengklasifikasikan citra sesuai dengan kategorinya yang ada pada *output*.



Gambar 14. *Connected Layer*

2.4.5 *Optimizer*

Optimizer adalah algoritma atau metode yang digunakan untuk mengubah atribut yang ada pada jaringan saraf tiruan. Atribut-atribut yang dapat diubah menggunakan algoritma ini yaitu nilai *weight* dan *learning rate* dengan tujuan untuk mengurangi nilai *loss* selama proses pelatihan[22].

Adapun beberapa *optimizer* yaitu sebagai berikut:

1. *Adaptive Moment Estimation (Adam)*

Adam merupakan algoritma yang dikembangkan dari algoritma *Stochastic Gradient Descent (SGD)* klasik yang dimana nilai bobot jaringan telah diperbaru. Penggunaan algoritma ini memerlukan pengaturan *learning rate* terlebih dahulu sebelum melakukan proses pelatihan. *Learning rate* adalah parameter pelatihan yang digunakan untuk menghitung nilai koneksi bobot ketika proses pelathinan dilakukan. Besar kecilnya nilai *learning rate* yang digunakan akan berpengaruh terhadap keefektifan dan kefisiensian proses pelatihan[23].

2. *Root Mean Square Propagation (RMSProp)*

Root Mean Square Propagation (RMSProp) adalah algoritma yang menggunakan besaran nilai rata-rata dari *weight* untuk menentukan besaran *learning rate*. Pada algoritma ini nilai rata-rata dari *weight* ditentukan dari nilai pertama dalam gradien[24].

3. *Stochastic Gradient Descent (SGD)*

Stochastic Gradient Descent adalah metode optimasi berbasis statistik yang efisien digunakan untuk mencari nilai koefisien untuk meminimalkan *loss (error) function* pada data yang diuji sehingga akurasi yang diperoleh akan maksimal dan *error* dapat diminimalisir[25].

2.4.6 *Loss Function*

Loss Function adalah persamaan matematika yang digunakan untuk menghitung nilai *loss*. Nilai *loss* yang dihitung digunakan pada proses *back propagation* untuk mengevaluasi parameter seperti nilai bobot (*weight*) dan bias supaya jaringan saraf tiruan menjadi lebih baik. *Loss function* dibagi menjadi dua yaitu *regression* dan *classification*. *Regression* memprediksi nilai diskrit, sedangkan *classification* memprediksi nilai kontinyu.

Salah satu *loss function* dari *classification* yaitu *categorical cross entropy* atau disebut juga *softmax loss*. Fungsi ini digunakan pada *multiclass classification*. Adapun persamaan matematika fungsi *cross entropy* ini adalah sebagai berikut:

$$CE = -\log \left(\frac{e^{s_p}}{\sum_j^C e^{s_j}} \right)$$

C = indeks akhir lapisan

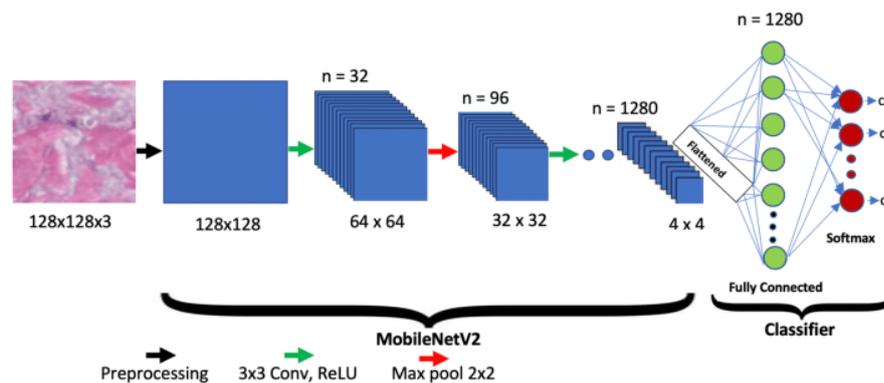
j = indeks awal lapisan

s_p = vektor *output*

s_j = vektor dari indeks j [26]

2.5 Arsitektur MobileNetV2

MobileNetV2 merupakan salah satu arsitektur *Convolutional Neural Network* yang dapat mengatasi masalah sumberdaya komputasi pada perangkat seluler (*mobile*) yang digunakan untuk mengatasi kebutuhan sumber daya komputasi yang besar. MobileNetV2 merupakan arsitektur CNN penyempurna versi sebelumnya yaitu MobileNetV1. Adapun yang membedakan antara arsitektur MobileNetV2 dan arsitektur lainnya yaitu ada pada lapisan konvolusi yang digunakan. Ketebalan lapisan *filter* pada MobileNetV2 sesuai dengan ketebalan dari *input image* yang digunakan[27].



Gambar 15. Arsitektur MobileNetV2

(sumber: https://www.researchgate.net/figure/The-proposed-MobileNetV2-network-architecture_fig1_350152088)

Adapun beberapa lapisan *convolutional neural network* yang digunakan pada arsitektur ini yaitu sebagai berikut:

1. *Depthwise separable convolution*

Depthwise separable convolution merupakan blok bangunan utama untuk arsitektur jaringan saraf tiruan yang digunakan untuk menggantikan operasi konvolusional

penuh yang membagi dua lapisan terpisah. Lapisan pertama yaitu lapisan *depthwise convolution* dan lapisan kedua yaitu lapisan *pointwise convolution*[28].

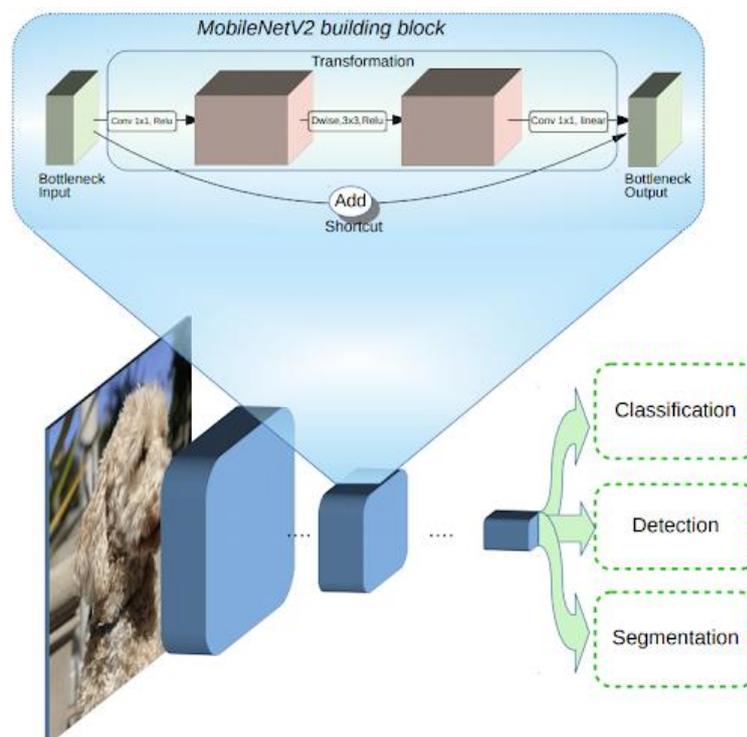
2. *Depthwise Convolution*

Depthwise convolution merupakan konvolusi yang melakukan pemfilteran ringan menggunakan filter konvolusi tunggal untuk setiap saluran input[28].

3. *Pointwise convolution*

Pointwise convolution merupakan kernel berukuran 1×1 yang digunakan untuk menggabungkan seluruh hasil konvolusi untuk memproduksi kelas efisien dari *depthwise convolution*. *Pointwise convolution* bertugas membangun fitur baru melalui komputasi kombinasi *linier* dari saluran *input*[29].

4. *Linear Bottleneck* dan *Shortcut Connection* di antara *Bottlenecks*

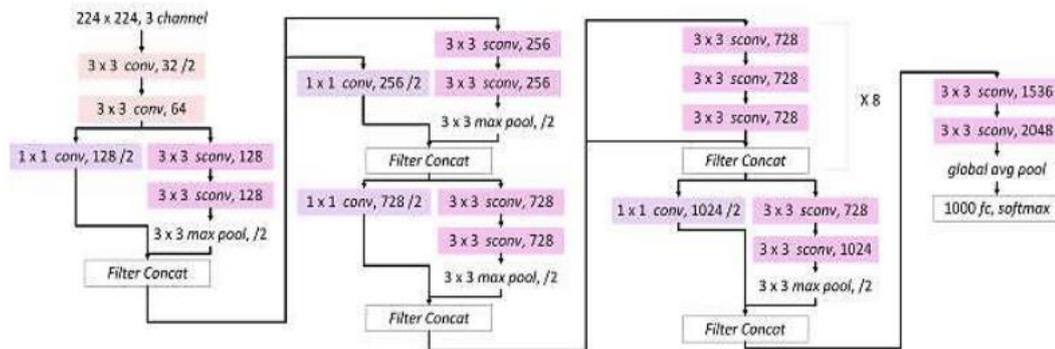


Gambar 16. *Linear Bottleneck* dan *Shortcut Connection* di antara *Bottlenecks*
(sumber: <https://ai.googleblog.com/2018/04/mobilenetv2-next-generation-of-on.html>)

Apabila pada MobileNetV1 hanya ditemukan *depthwise* dan *pointwise convolution*, maka pada MobileNetV2 ini terdapat fitur tambahan yaitu *linear bottleneck* dan *shortcut connection* di antara *bottlenecks*. Pada bagian *bottleneck* terdapat *input* dan *output* antara model sedangkan lapisan yang berada di bagian dalam mengenkapsulasi kemampuan model untuk mengubah *input* dari konsep yang lebih rendah (piksel) menjadi lebih tinggi (gambar). *Shortcut* antar *bottlenecks* memungkinkan proses pelatihan menjadi lebih cepat dengan akurasi lebih baik[27].

2.6 Xception

Xtreme of Inception (Xception) adalah arsitektur *Convolutional Neural Network* yang menggunakan metode *depthwise separable convolution*. Arsitektur ini merupakan hasil dari pengembangan arsitektur Inception yang memiliki 36 lapisan konvolusi yang membentuk basis jaringan ekstraksi fitur.



Gambar 17. Arsitektur Xception[30]

Arsitektur Xception memiliki lapisan-lapisan konvolusional dan jaringan residual yang memisahkan lapisan konvolusional. Pada blok konvolusional pertama dilakukan pemisahan setiap saluran secara spasial dengan menggunakan konvolusi 1x1, selanjutnya *output* tersebut ditangkap dengan melakukan *filter concat* yang kemudian dilakukan konvolusi selanjutnya[30].

Adapun beberapa lapisan convolutional neural network yang digunakan pada arsitektur ini yaitu sebagai berikut:

1. *Depthwise separable convolution*

Depthwise separable convolution merupakan blok bangunan utama untuk arsitektur jaringan saraf tiruan yang digunakan untuk menggantikan operasi konvolusional penuh yang membagi dua lapisan terpisah. Lapisan pertama yaitu lapisan *depthwise convolution* dan lapisan kedua yaitu lapisan *pointwise convolution*[28].

2. *Depthwise Convolution*

Depthwise convolution merupakan konvolusi yang melakukan pemfilteran ringan menggunakan filter konvolusi tunggal untuk setiap saluran input[28].

3. *Pointwise convolution*

Pointwise convolution merupakan kernel berukuran 1×1 yang digunakan untuk menggabungkan seluruh hasil konvolusi untuk memproduksi kelas efisien dari *depthwise convolution*. *Pointwise convolution* bertugas membangun fitur baru melalui komputasi kombinasi linier dari saluran *input*[29].

2.7 **Pembagian Data**

Pembagian data adalah membagi data kedalam beberapa bagian yaitu data latih, data validasi, dan data uji. Data latih adalah data yang digunakan pada tahapan pelatihan model. Data validasi adalah data yang digunakan untuk memvalidasi hasil model yang sudah dilatih. Data uji adalah data yang digunakan untuk menguji model setelah tahapan pengujian dan validasi. *Library* split-folders digunakan untuk membagi dataset ke dalam beberapa direktori yaitu direktori “train”, “validation”, dan “test.” [31]

2.8 Augmentasi Data

Augmentasi data adalah cara yang digunakan untuk menambah keragaman data yang tersedia tanpa mengumpulkan data baru untuk model yang dilatih. Tujuan augmentasi data yaitu untuk menghindari resiko *overfitting* atau nilai akurasi pengujian jauh lebih kecil dari nilai akurasi pelatihan[32].

2.9 Google Colab

Google Colab adalah layanan *open source* berbasis cloud yang dikembangkan oleh Google Research. Layanan ini merupakan alat yang dapat menyimpan, menulis, dan membagikan program yang dibuat di dalamnya ke Google Drive. Google Colab mampu menjalankan kode Python tanpa perlu melakukan proses instalasi. Ada beberapa manfaat pada Google Colab di antaranya yaitu sebagai berikut:

1. Pre-Installed Library

Google Colab dilengkapi dengan *library-library* yang dapat digunakan untuk mengolah data secara lengkap misalnya *library* Keras, TensorFlow, dan PyTorch.

2. Kolaborasi

Google Colab memudahkan penggunaanya untuk bekerjasama dengan pengguna lainnya, sehingga pekerjaan yang di lakukan di dalamnya dapat diakses dan dimanejemen secara berkelompok.

3. Fitur GPU Gratis

Google Colab menawarkan sumber daya komputasi gratis yang berjalan di atas *cloud*. Layanan ini mampu menjalankan program selama hingga dua belas jam.

4. Saling Terhubung

Google Colab dapat dihubungkan dengan Jupyter Notebook yang berada di komputer pribadi (*local runtime*), Google Drive, dan Github[33].

2.10 TensorFlow

TensorFlow adalah perangkat lunak yang memiliki pustaka *open-source* yang dapat digunakan untuk membuat model *machine learning*. TensorFlow merupakan hasil pengembangan oleh Google Brain dari perusahaan Google. TensorFlow sangat fleksibel karena dapat menggunakan banyak jenis *runtime* seperti *Graphical Processing Units* (GPU), *Control Processing Units* (CPU), dan *Tensor Processing Units* (TPU). Fokus utama TensorFlow adalah pada *deeplearning* dan *machine learning*[34].

2.11 Keras

Keras merupakan *Application Programming Interface* (API) *neural-network* tingkat tinggi yang disusun dari bahasa pemrograman Python. Keras dapat dijalankan pada *software* TensorFlow. Keras dapat berjalan dengan *runtime* *Graphical Processing Units* (GPU) dan *Control Processing Units* (CPU)[35].

2.11.1 Library Keras

Keras menyediakan beberapa *library* yang dapat digunakan untuk membuat model *machine learning*.

1. Kelas ImageDataGenerator

ImageDataGenerator merupakan kelas pada API Keras yang digunakan sebagai augmentasi data. Kelas ImageDataGenerator akan membangkitkan data sesuai dengan setiap argumen yang digunakan. Terdapat beberapa argumen pada kelas

ImageDataGenerator yang berguna dalam proses augmentasi data. Nilai untuk setiap argumen dapat diberikan secara bebas sesuai dengan kebutuhan. Adapun argumen pada kelas ini yaitu sebagai berikut:

1. Argumen `rescale` mengubah nilai array menjadi nilai yang lain.
 2. Argumen `rotation_range` untuk merotasi gambar.
 3. Argumen `width_shift_range` untuk menggeser gambar ke kanan atau ke kiri dari posisi awal
 4. Argumen `height_shift_range` untuk menggeser gambar ke atas atau ke bawah dari posisi awal.
 5. Argumen `shear_range` untuk memiringkan sekaligus merenggankan gambar dari posisi awal.
 6. Argumen `zoom_range` untuk memperbesar dan memperkecil gambar sebesar.
 7. Argumen `horizontal_flip` untuk membalik gambar secara horizontal.
 8. Argumen `fill_mode` diberikan untuk memberikan nilai piksel yang kosong.
2. Fungsi `flow_from_directory`

Adapun argumen pada fungsi `flow_from_directory` yaitu sebagai berikut:

1. Argumen pertama yaitu lokasi direktori yang akan digunakan untuk menyimpan data yang akan diaugmentasi yang mana biasanya argument ini dinamai dengan variabel direktori tersebut seperti `train_dir`, `test_dir`, dan `val_dir`.
2. Argumen `batch_size` yang digunakan untuk membagi dataset kedalam beberapa batch. Nilai `batch_size` yang populer dan direkomendasikan untuk berbagai arsitektur CNN yaitu sebesar 32.
3. Argumen `class_mode` yang digunakan untuk menggunakan mode apakah kelas yang digunakan, misalkan kelas `categorical` yang digunakan untuk mengkategorikan beberapa kelas.
4. Argumen `target_size` yang digunakan untuk mengubah ukuran piksel untuk agar sesuai dengan lapisan input model. Nilai `target_size` yang populer dan umumnya untuk berbagai arsitektur CNN yaitu sebesar 224 x 224.

3. Kelas ModelCheckpoint dan EarlyStopping

Kelas ModelCheckpoint digunakan untuk menyimpan bobot dari model yang memiliki nilai akurasi terbaik ke dalam *file* berekstensi *.hdf5*. Kelas EarlyStopping digunakan untuk memberhentikan proses pelatihan ketika nilai *accuracy* tidak lagi meningkat dalam beberapa *epoch* yang sesuai dengan besarnya nilai pada variabel *patience* diberikan.

4. Kelas MobileNetV2

Kelas ini berisikan arsitektur MobileNetV2. Terdapat argumen di dalamnya yaitu *input_shape* untuk menentukan ukuran *input* dan *channel* yang akan digunakan, *include_top* untuk menyertakan lapisan atas arsitektur, dan *weight* merupakan nilai bobot yang digunakan. Argumen pada kelas ini yaitu sebagai berikut:

1. Argumen *input_shape* merupakan *input* ukuran piksel gambar yang digunakan dengan kanalnya yaitu hitam-putih atau RGB (*Red, Green, Blue*).
2. Argumen *include_top* untuk menyertakan lapisan terakhir sebelum lapisan top arsitektur MobileNetV2.
3. Argumen *weights* merupakan bobot yang akan digunakan arsitektur.
4. Terdapat variabel *trainable* pada fungsi *layers* yang dapat membekukan nilai bobot arsitektur ketika proses pelatihan dilakukan.

5. Kelas Xception

Kelas ini berisikan arsitektur Xception. Terdapat argumen di dalamnya yaitu *input_shape* untuk menentukan ukuran *input* dan *channel* yang akan digunakan, *include_top* untuk menyertakan lapisan atas arsitektur, dan *weight* merupakan nilai bobot yang digunakan. Argumen pada kelas ini yaitu sebagai berikut:

1. Argumen *input_shape* merupakan *input* ukuran piksel gambar yang digunakan dengan kanalnya yaitu hitam-putih atau RGB (*Red, Green, Blue*).
2. Argumen *include_top* untuk menyertakan lapisan terakhir sebelum lapisan top arsitektur MobileNetV2.
3. Argumen *weights* merupakan bobot yang akan digunakan arsitektur.

4. Terdapat variabel trainable pada fungsi *layers* yang dapat membekukan nilai bobot arsitektur ketika proses pelatihan dilakukan.

6. Kelas Model

Kelas ini merupakan objek yang akan menyatukan *layer* yang digunakan pada model. Argumen input merupakan lapisan *input* yang digunakan model dan argumen output merupakan lapisan *output* atau prediksi yang ada pada model.

7. Metode *Compile*

Metode ini digunakan untuk menyusun model dengan *optimizer*, *loss function*, dan *metrics* yang diinginkan.

8. Metode *Fit*

Metode ini digunakan untuk melatih model sesuai dengan *epoch* yang ditentukan.

9. Metode *Evaluate*

Metode ini digunakan untuk mengembalikan nilai sesuai dengan *metrics* yang digunakan dari model yang sudah dilatih[6].

2.12 Penelitian Terkait

Tabel 1. Penelitian Terkait

No.	Tahun	Judul	Nama Peneliti	Metode	Hasil
1	2021	PERANCANGAN MODEL PENGENALAN CITRA DIGITAL TOMOGRAFI	Maulana Akbar Dwijaya , Umar Ali Ahmad , dan	CNN	Model deteksi paru yang dirancang menggunakan

		PARU-PARU UNTUK DETEKSI COVID-19 MENGUNAKAN JARINGAN SARAF TIRUAN	Rudi Purwo Wijayanto		an metode CNN memiliki akurasi sebesar 93%
2	2022	DETEKSI MASKER WAJAH MENGUNAKAN METODE CONVOLUTIONAL NEURAL NETWORK (CNN) MENINGKATKAN NILAI AKURASI MELALUI ARSITEKTUR LAYER KONVOLUSI	Mahendar Dwi Payana , Desita Ria Yusian TB, Zuhar Musliyana, dan M. Bayu Wibawa	CNN	Model deteksi masker yang dirancang menggunakan metode CNN memiliki akurasi sebesar 92.287%
3	2021	Comparative Study of VGG16 and MobileNetV2 for Masked Face Recognition	Faisal Dharma Adhinata, Nia Annisa Ferani Tanjung, Widi Widayat, Gracia Rizka	MobileNet V2 danVGG16	Model deteksi masker yang dirancang menggunakan arsitektur MobileNet V2

			Pasfica, dan Fadlan Raka Satura		memiliki akurasi sebesar 95.42% dan model VGG16 sebesar 92.57%
4	2022	Rancang Bangun Sistem Klasifikasi Sampah Anorganik Kantor menggunakan Deep Learning Arsitektur Xception berbasis NVIDIA Jetson Nano	Rahmatul Bijak Nur Kholis dan Fitri Utamingrum	Xception	Model klasifikasi sampahan g dirancang menggunakan arsitektur Xception memiliki akurasi pada sistem sebesar 91,67%
5	2021	Optimasi Akurasi Metode Convolutional Neural Network untuk Identifikasi Jenis Sampah	Rima Dias Ramadhani, Afandi Nur Aziz Thohari, Condro Kartiko, Apri Junaidi, Tri Ginanjar Laksana, dan	CNN	Model identifikasi jenis sampah yang dirancang menggunakan metode CNN memiliki akurasi

			Novanda Alim Setya Nugraha		sebesar 91,2%
--	--	--	----------------------------------	--	------------------

Dari penelitian-penelitian tersebut, hal yang dapat disimpulkan yaitu *Convolutional Neural Network* (CNN), Xception, dan MobileNetV2 merupakan metode dan arsitektur terbaik untuk melakukan klasifikasi citra. Pada penelitian ini, metode dan arsitektur tersebut akan diimplementasikan pada sistem klasifikasi sampah untuk membuat model dengan nilai akurasi yang optimal.

BAB III METODOLOGI PENELITIAN

3.1 Waktu dan Tempat Penelitian

Penelitian ini dilaksanakan di Laboratorium Komputer Jurusan Teknik Elektro Universitas Lampung, dengan waktu penelitian dilakukan pada 21 Agustus 2022 – 31 Januari 2023 dengan penjadwalan sebagai berikut.

3.2 Jadwal Penelitian

Jadwal yang direncanakan pada penelitian ini adalah sebagai berikut:

Tabel 2. Jadwal Penelitian

No	Aktivitas	Bulan ke-						Milestone
		1	2	3	4	5	6	
1	Perumusan Masalah							<i>Research Question</i>
2	Studi Literatur							Review dari jurnal terkait
3	Pengumpulan data							Mendapatkan dataset yang bersumber dari Kaggle dan menambahkan dataset asli dari penulis
4	Pengolahan data							Mengolah data yang didapat pada Google Colab
5	Perancangan model							Merancang model <i>machine learning</i> menggunakan

						Arsitektur CNN MobileNetV2 dan Xception pada Google Colab
6	Pelatihan model					Melatih model di Google Colab
7	Pembandingan model					Melakukan perbandingan nilai akurasi antara model yang menggunakan arsitektur MobileNetV2 dan Xception

3.3 Alat Dalam Penelitian

Alat dan bahan yang digunakan dalam penelitian ini adalah sebagai berikut:

3.3.1 Alat Penelitian

Adapun alat-alat yang digunakan dalam penelitian, berupa sebuah laptop dengan spesifikasi yang ditunjukkan pada tabel 2 berikut:

Tabel 3. Alat Penelitian.

No	Nama Alat	Spesifikasi	Deskripsi
1	Laptop	Intel Core i3, RAM 6GB, dengan sistem operasi Windows 10	Perangkat keras yang digunakan untuk membuat model.
2	Kamera <i>Smartphone</i>	12 MP	Perangkat keras yang digunakan untuk mengambil sampel dataset.
3	Google Colab		Perangkat lunak yang digunakan

			untuk membuat dan membandingkan model <i>machine learning</i>
4	Microsoft Word 2019, Microsoft Excel 2019		Perangkat lunak yang digunakan untuk melakukan perhitungan dan penyusunan skripsi

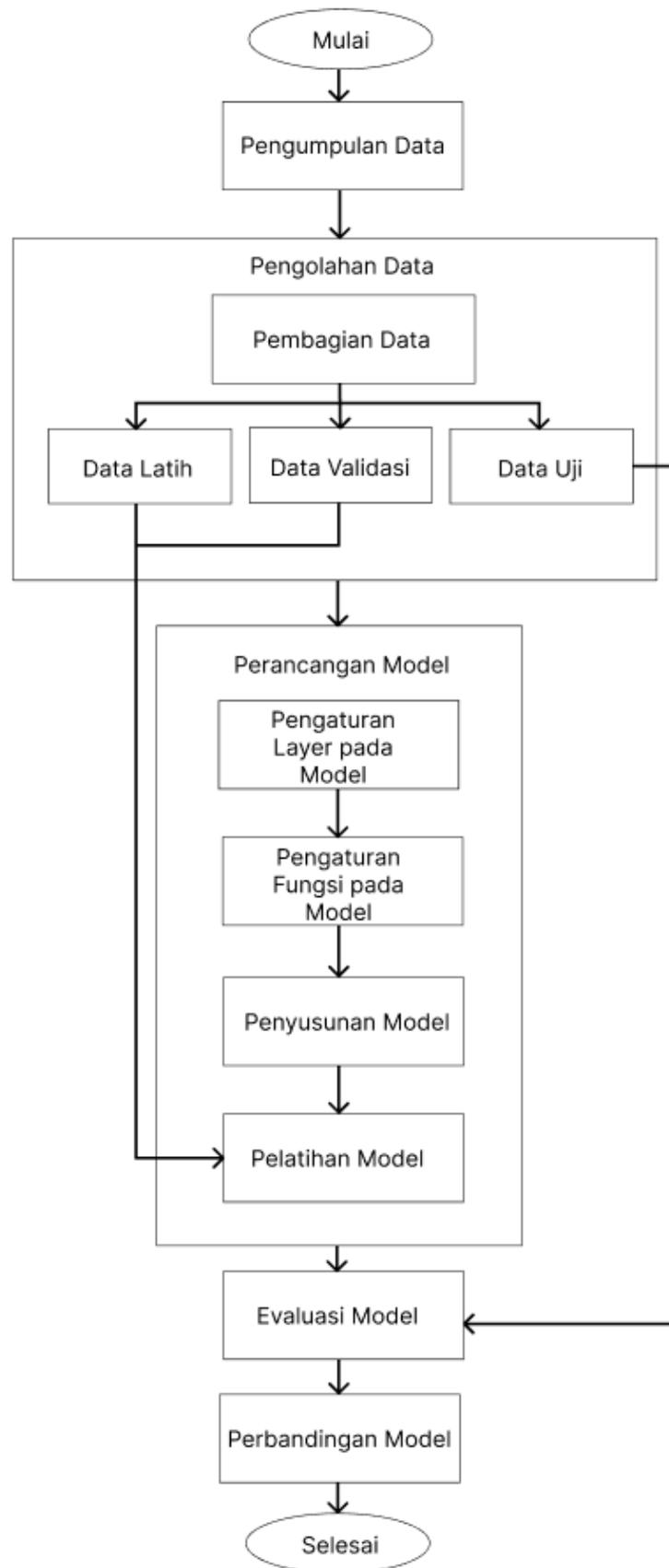
3.3.2 Bahan Penelitian

Adapun bahan penelitian pada penelitian ini adalah sebagai berikut:

Bahan penelitian berupa dataset Kaggle, dataset foto yang diperoleh secara mandiri, dataset yang bersumber dari internet, buku, skripsi, jurnal, dan sumber ilmiah lainnya yang didapat dari berbagai situs web.

3.4 Kerangka Penelitian

Berdasarkan landasan teori dan rumusan masalah maka dapat diidentifikasi bahwa kerangka berpikir yang digunakan pada penelitian ini yaitu sebagai berikut:



Gambar 18. Kerangka Penelitian

3.4.1 Pengumpulan Data

Pada tahapan ini, dicari dataset dari berbagai sumber seperti Kaggle, internet, hingga memproduksi dataset secara mandiri dengan cara memfoto beberapa sampel secara langsung menggunakan *smartphone*. Memproduksi dataset secara mandiri ini bertujuan untuk meningkatkan akurasi model yang akan dilatih. Data yang dikumpulkan harus memiliki ukuran piksel lebih dari 224x224 piksel dan harus berwarna. Ini karena lapisan *input* arsitektur yang akan digunakan memiliki ukuran 224x224 piksel serta berkanal RGB (*Red, Green, Blue*). Adapun contoh gambar yang dikumpulkan dapat dilihat pada Gambar 19. Pengumpulan Data berikut.



Gambar 19. Pengumpulan Data

Gambar di atas merupakan dataset yang akan digunakan pada tahapan pelatihan, validasi, atau pengujian. Gambar tersebut mewakili masing-masing kelas yaitu *cardboard*, *glass*, *metal*, *paper*, *plastic*, dan *trash*.

3.4.2 Pengolahan Data

Pada tahapan ini dilakukan pembagian data dari data yang sudah dikumpulkan menjadi tiga bagian yaitu data latih (*training*), data validasi (*validation*), dan data uji (*testing*). Pembagian data tersebut dilakukan agar data dapat dilatih pada model yang akan dirancang dan kemudian dapat dievaluasi.

Data latih digunakan untuk melatih model *machine learning*. Data uji digunakan untuk menguji model *machine learning* yang di mana pada tahap pelatihan model akan diperoleh nilai akurasi modelnya untuk setiap *epoch*. Data validasi digunakan untuk melakukan validasi terhadap model yang dimana pada tahap ini akan diperoleh nilai validasi akurasi dari model *machine learning* yang sudah dilatih. Nilai akurasi ini yang menentukan baik atau buruknya suatu model yang dibuat.

3.4.3 Perancangan Model

Pada tahapan ini, model dirancang sesuai dengan konfigurasi yang ada pada setiap arsitektur CNN, misalkan lapisan teratas arsitektur MobileNetV2 memiliki 1280 dimensi yang kemudian diberikan *Global Average Pooling* sebagai pengganti *Flatten*, maka penulis akan membuat jumlah *neuron* pada lapisan *Fully-Connected* sebanyak 1280 *neuron*. Sedangkan untuk arsitektur Xception, penulis akan menggunakan 2048 *neuron* dan kemudian menyambungkannya ke lapisan *output* untuk melakukan klasifikasi. Penulis akan menggunakan lapisan *input* dengan ukuran 224 x 224 piksel untuk kedua arsitektur.

1. Pengaturan Fungsi pada Model

Pada tahapan ini penulis menggunakan fungsi-fungsi yang ada pada *framework* TensorFlow untuk membangun model MobileNetV2 dan Xception.

2. Pengaturan *Layer* pada Model

Pada tahapan ini, lapisan *input*, *fully-connected*, dan lapisan *output* diatur sesuai dengan masing-masing arsitektur. Pada arsitektur MobileNetV2 *neuron* pada lapisan *fully-connected* yang akan digunakan yaitu 1280 *neuron*, sedangkan pada arsitektur Xception sebanyak 2048 *neuron*. Adapun pengaturan layer pada masing-masing model yaitu sebagai berikut:

1. Arsitektur MobileNetV2

Pada tahapan ini, digunakan *input shape* dengan nilai 224 x 224 piksel dengan kanal RGB karena citra yang digunakan memiliki warna. Lapisan *fully-connected* yang digunakan sesuai dengan arsitektur ini yaitu sebanyak 1280 *neuron*. Jumlah *neuron* keluaran (prediksi) akan menyesuaikan jumlah kelas pada dataset yang akan digunakan.

2. Arsitektur Xception

Pada tahapan ini, digunakan *input shape* dengan nilai 224 x 224 piksel dengan kanal RGB karena citra yang digunakan memiliki warna. Lapisan *fully-connected* yang digunakan sesuai dengan arsitektur ini yaitu sebanyak 2048 *neuron*. Jumlah *neuron* keluaran (prediksi) akan menyesuaikan jumlah kelas pada dataset yang akan digunakan.

3.4.4 Penyusunan Model

Pada tahapan ini, *optimizers*, *learning rate*, dan *metrics* yang digunakan didefinisikan pada model untuk setiap arsitektur yang akan dilatih.

1. Arsitektur MobileNetV2

Model yang akan dibuat akan disusun menggunakan *optimizer* dan *learning rate* yang berbeda. *Optimizer* yang akan digunakan yaitu Adam, RMSprop, dan SGD. Sedangkan *learning rate* yang akan digunakan yaitu sebesar 0.001, 0.01, dan 0.1.

2. Arsitektur Xception

Sama seperti arsitektur MobileNetV2 sebelumnya, model yang akan dibuat akan disusun menggunakan *optimizer* dan *learning rate* yang sama. *Optimizer* yang akan

digunakan yaitu Adam, RMSprop, dan SGD. Sedangkan *learning rate* yang akan digunakan yaitu sebesar 0.001, 0.01, dan 0.1.

3.4.5 Pelatihan Model

Pada tahapan ini model dilatih sesuai dengan arsitektur masing-masing yaitu MobileNetV2 dan Xception.

1. Arsitektur MobileNetV2

Pada tahapan ini, model dilatih menggunakan fungsi dan argumen yang disediakan *framework* TensorFlow. Penulis akan menggunakan dataset yang sebelumnya dibagi untuk proses pelatihan dan validasi model serta menggunakan fungsi *callbacks* yang disediakan *framework* TensorFlow dengan harapan dapat mengefisienkan proses pelatihan.

2. Arsitektur Xception

Pada tahapan model dilatih menggunakan fungsi dan argumen yang disediakan *framework* TensorFlow. Penulis akan menggunakan dataset yang sebelumnya dibagi untuk proses pelatihan dan validasi model serta menggunakan fungsi *callbacks* yang disediakan *framework* TensorFlow dengan harapan dapat mengefisienkan proses pelatihan.

3.4.6 Evaluasi Model

Pada tahapan ini dilakukan evaluasi model menggunakan data uji dan metode *evaluate* yang disediakan *framework* TensorFlow dengan tujuan mendapatkan nilai akurasi model.

3.4.7 Perbandingan Model

Pada tahapan ini, dilakukan perbandingan nilai akurasi yang dihasilkan setiap model dengan *optimizer* dan *learning rate* yang berbeda-beda untuk mencari model mana yang memiliki akurasi terbaik.

BAB V

KESIMPULAN DAN SARAN

5.1 Kesimpulan

Model *machine learning* pada sistem klasifikasi sampah yang dirancang menggunakan arsitektur Xception dengan *optimizer Adam* dan *Learning Rate* sebesar 0.001 memiliki akurasi yang lebih baik dibandingkan dengan model yang dirancang menggunakan arsitektur MobileNetV2 yang menggunakan *optimizer SGD* dan *Learning Rate* sebesar 0.1 tetapi arsitektur Xception membutuhkan waktu > 2 detik untuk setiap *epoch* dibandingkan arsitektur MobileNetV2. Model Xception dengan *optimizer* dan *learning rate* tersebut memiliki akurasi sebesar 87.81% sedangkan model MobileNetV2 dengan *optimizer* dan *learning rate* tersebut memiliki akurasi sebesar 86.07%.

5.2 Saran

1. Penelitian berikutnya dapat dilakukan dengan meningkatkan jumlah dataset agar akurasi dari kedua model dapat menjadi lebih baik.
2. Penelitian berikutnya dapat menggunakan arsitektur CNN, *optimizer*, dan *learning rate* yang berbeda agar akurasi model yang didapat menjadi lebih baik.

DAFTAR PUSTAKA

- [1] “Sustainability Report 2021 – Kampus Hijau Universitas Lampung.” <https://greenmetric.unila.ac.id/media/laporan-hijau-berkelanjutan/2021/sustainability-report-2021/> (accessed Jan. 25, 2023).
- [2] “Mengenal Berbagai Jenis Sampah yang Sulit Terurai dan Waktunya!” <https://www.gramedia.com/literasi/sampah-yang-sulit-terurai-dan-waktunya/> (accessed Jan. 25, 2023).
- [3] D. Kusumaningsari, J. Fisika, F. Matematika, D. Ilmu, and P. Alam, “PEMANFAATAN DAN PENGOLAHAN SAMPAH ORGANIK DAN NON-ORGANIK UTILIZATION AND PROCESSING OF ORGANIC AND NON-ORGANIC WASTE.”
- [4] M. D. Payana, D. R. Y. TB, Z. Musliyana, and M. B. Wibawa, “DETEKSI MASKER WAJAH MENGGUNAKAN METODE CONVOLUTIONAL NEURAL NETWORK (CNN) MENINGKATKAN NILAI AKURASI MELALUI ARSITEKTUR LAYER KONVOLUSI,” *JOURNAL OF INFORMATICS AND COMPUTER SCIENCE*, vol. 8, no. 1, pp. 30–35, 2022, doi: 10.33143/JICS.VOL8.ISS1.2123.
- [5] “Pengertian dan Cara Kerja Algoritma Convolutional Neural Network (CNN) - Trivusi.” <https://www.trivusi.web.id/2022/04/algoritma-cnn.html> (accessed Jan. 25, 2023).
- [6] “Keras API reference.” <https://keras.io/api/> (accessed Jan. 25, 2023).
- [7] “Apa Itu Kecerdasan Buatan? Berikut Pengertian dan Contohnya - Dicoding Blog.” <https://www.dicoding.com/blog/kecerdasan-buatan-adalah/> (accessed Jan. 25, 2023).
- [8] Yaya Heryadi and Teguh Wahyono, *Machine Learning Konsep dan Implementasi*. Yogyakarta: Penerbit Gava Media, 2020.

- [9] A. S. R. Sinaga, “Machine Learning Prediksi Karakter Pengguna Hastag (#) Bahasa Generasi Milenial Di Sosial Media,” *IJAI (Indonesian Journal of Applied Informatics)*, vol. 4, no. 2, pp. 165–171, Aug. 2020, doi: 10.20961/IJAI.V4I2.41764.
- [10] Dios Kurniawan, *Penenalan Machine Learning dengan Python*. Jakarta: PT Elex Media Komputindo, 2020.
- [11] Lanny W. Pandjaitan, *Dasar-dasar Komputasi Cerdas*. Yogyakarta: ANDI.
- [12] M. Yanto, “PENERAPAN JARINGAN SYARAF TIRUAN DENGAN ALGORITMA PERCEPTRON PADA POLA PENENTUAN NILAI STATUS KELULUSAN SIDANG SKRIPSI,” *Jurnal Teknoif Teknik Informatika Institut Teknologi Padang*, vol. 5, no. 2, pp. 79–87, Oct. 2017, doi: 10.21063/JTIF.2017.V5.2.79-87.
- [13] M. A. Dwijaya, U. A. Ahmad, and R. P. Wijayanto, “Perancangan Model Pengenalan Citra Digital Tomografi Paru-Paru Untuk Deteksi Covid-19 Menggunakan Jaringan Saraf Tiruan,” *eProceedings of Engineering*, vol. 8, no. 6, Dec. 2021, doi: 10.34818/EOE.V8I6.17039.
- [14] “Klasifikasi Emosi Berdasarkan Ciri Wajah Menggunakan Convolutional Neural Network | Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer.” <https://j-ptiik.ub.ac.id/index.php/j-ptiik/article/view/6732> (accessed Jan. 25, 2023).
- [15] “Jaringan Syaraf Tiruan | Pemrograman Matlab.” <https://pemrogramanmatlab.com/data-mining-menggunakan-matlab/jaringan-syaraf-tiruan-menggunakan-matlab/> (accessed Jan. 25, 2023).
- [16] “Jaringan Saraf Tiruan (JST).” <https://www.kajianpustaka.com/2016/11/jaringan-saraf-tiruan-jst.html> (accessed Jan. 26, 2023).
- [17] “Contoh Perhitungan Algoritma Backpropagation - Structilmy.” <https://structilmy.com/blog/2019/07/31/contoh-perhitungan-algoritma-backpropagation/> (accessed Jan. 26, 2023).
- [18] A. Rahim, K. Kusriani, and E. T. Luthfi, “Convolutional Neural Network untuk Kalasifikasi Penggunaan Masker,” *Inspiration: Jurnal Teknologi*

- Informasi dan Komunikasi*, vol. 10, no. 2, pp. 109–115, Dec. 2020, doi: 10.35585/INSPIR.V10I2.2569.
- [19] IdMetafora.com, “Memahami Apa Itu Convolutional Neural Network! Yuk Simak Penjelasannya di Sini, Jasa Pembuatan Website - Metafora Indonesia Tehnology,” Apr. 2014.
- [20] S. RIZAL, N. IBRAHIM, N. K. C. PRATIWI, S. SAIDAH, and R. Y. N. FU’ADAH, “Deep Learning untuk Klasifikasi Diabetic Retinopathy menggunakan Model EfficientNet,” *ELKOMIKA: Jurnal Teknik Energi Elektrik, Teknik Telekomunikasi, & Teknik Elektronika*, vol. 8, no. 3, p. 693, Aug. 2020, doi: 10.26760/ELKOMIKA.V8I3.693.
- [21] “Global Average Pooling Explained | Papers With Code.” <https://paperswithcode.com/method/global-average-pooling> (accessed Jan. 25, 2023).
- [22] “Various Optimization Algorithms For Training Neural Network | by Sanket Doshi | Towards Data Science.” <https://towardsdatascience.com/optimizers-for-training-neural-network-59450d71caf6> (accessed Jan. 25, 2023).
- [23] N. Rochmawati, H. B. Hidayati, Y. Yamasari, H. P. A. Tjahyaningtjas, W. Yustanti, and A. Prihanto, “Analisa Learning Rate dan Batch Size pada Klasifikasi Covid Menggunakan Deep Learning dengan Optimizer Adam,” *Journal of Information Engineering and Educational Technology*, vol. 5, no. 2, pp. 44–48, Dec. 2021, doi: 10.26740/JIEET.V5N2.P44-48.
- [24] “Apa itu Adam Optimizer? – JawabanApapun.com.” <https://jawabanapapun.com/apa-itu-adam-optimizer/> (accessed Jan. 25, 2023).
- [25] R. R. Hadistio, R. Rinaldi Hadistio, H. Mawengkang, and M. Zarlis, “Perbandingan Algoritma Stochastic Gradient Descent dan Naïve Bayes Pada Klasifikasi Diabetic Retinopathy,” *JURNAL MEDIA INFORMATIKA BUDIDARMA*, vol. 6, no. 1, pp. 271–277, Jan. 2022, doi: 10.30865/MIB.V6I1.3426.
- [26] Muhammad Afif Amanullah Fawwaz, Kurniawan Nur Ramadhani, and Febryanti Sthevanie, “Klasifikasi Ras pada Kucing menggunakan Algoritma Convolutional Neural Network(CNN),” 2020.

- [27] “MobileNet: Deteksi Objek pada Platform Mobile | by Rizqi Okta Ekoputris | Nodeflux | Medium.” <https://medium.com/nodeflux/mobilenet-deteksi-objek-pada-platform-mobile-bbbf3806e4b3> (accessed Jan. 25, 2023).
- [28] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L.-C. Chen, “MobileNetV2: Inverted Residuals and Linear Bottlenecks.” pp. 4510–4520, 2018.
- [29] A. P. Muharram, “EVALUASI PERFORMA METODE DEEP LEARNING UNTUK KLASIFIKASI CITRA LESI KULIT THE HAM10000”, doi: 10.5614/sniko.2018.10.
- [30] Sandhopi, L. Z. P.C.S.W, and Y. Kristian, “Identifikasi Motif Jepara pada Ukiran dengan Memanfaatkan Convolutional Neural Network,” *Jurnal Nasional Teknik Elektro dan Teknologi Informasi*, vol. 9, no. 4, pp. 403–413, Dec. 2020, doi: 10.22146/JNTETI.V9I4.541.
- [31] “split-folders · PyPI.” <https://pypi.org/project/split-folders/> (accessed Jan. 25, 2023).
- [32] J. Sanjaya and M. Ayub, “Augmentasi Data Pengenalan Citra Mobil Menggunakan Pendekatan Random Crop, Rotate, dan Mixup,” *Jurnal Teknik Informatika dan Sistem Informasi*, vol. 6, no. 2, Aug. 2020, doi: 10.28932/JUTISI.V6I2.2688.
- [33] “Mengenal Google Colab, Aplikasi Pengolahan Data Open Source - Nuril Hidayah - anakteknik.co.id.” <https://www.anakteknik.co.id/111620868375095332510/articles/mengenal-google-colab-aplikasi-pengolahan-data-open-source> (accessed Jan. 25, 2023).
- [34] “Apa itu tensorflow? 3 Hal Penting Untuk Dipahami.” <https://auftechnique.com/apa-itu-tensorflow/> (accessed Jan. 25, 2023).
- [35] “Keras Vs Pytorch Vs TensorFlow - DosenIT.com.” <https://dosenit.com/ai/keras-vs-pytorch-vs-tensorflow> (accessed Jan. 25, 2023).