



**Klasifikasi Seragam Loreng Tentara Indonesia  
Angkatan Darat dan Komando Cadangan  
Strategis Angkatan Darat (Kostrad)  
Menggunakan Metode *Convolutional Neural  
Network (CNN)***

**Tugas Akhir**

**Oleh:  
Audrey Marito Hutahaean (4211801051)**

**Program Studi Teknik Mekatronika  
Jurusan Teknik Elektro  
Politeknik Negeri Batam  
2024**

## Pernyataan Keaslian Tugas Akhir

Saya yang bertandatangan dibawah ini menyatakan bahwa isi sebagian maupun keseluruhan Tugas Akhir saya yang berjudul: "Klasifikasi Seragam Loreng Tentara Indonesia Angkatan Darat dan Komando Cadangan Strategis Angkatan Darat (Kostrad) Menggunakan Metode *Convolutional Neural Network (CNN)*" adalah **hasil karya sendiri, diselesaikan tanpa menggunakan bahan-bahan yang tidak diizinkan, dan bukan merupakan karya pihak lain yang saya akui sebagai karya sendiri.** Semua referensi yang dikutip atau dirujuk telah ditulis secara lengkap pada daftar pustaka. Apabila ternyata pernyataan saya ini tidak benar, saya bersedia menerima sanksi sesuai peraturan yang berlaku.

Batam, 31 Juli 2024



---

Audrey Marito Hutahaean  
NIM: 4211801051

# Lembar Pengesahan

Tugas Akhir disusun untuk memenuhi salah satu syarat memperoleh gelar  
Sarjana Terapan Teknik (S.Tr.T)  
di  
Politeknik Negeri Batam

Oleh:  
Audrey Maito Hutahaean (4211801051)

Tanggal Sidang: 17 Juli, 2024

Disetujui oleh:

Dosen Penguji I

Dosen Pembimbing



Fadli Firdaus, S.Pd., M.Pd.  
NIK: 122271



Diono, S.Tr.T., M.Sc.  
NIK: 120243

Dosen Penguji II



Dr. Abdurahman Dwijotomo, S.ST., M.Sc.  
NIK: 122267

# **Klasifikasi Seragam Loreng Tentara Indonesia Angkatan Darat dan Komando Cadangan Strategis Angkatan Darat (Kostrad) Menggunakan Metode *Convolutional Neural Network (CNN)***

## **Abstrak**

Penelitian ini membahas penerapan metode *Convolutional Neural Network* dalam mengklasifikasikan seragam loreng tentara Indonesia, khususnya seragam loreng TNI AD dan seragam loreng Kostrad. Tujuan penelitian ini adalah merancang sebuah sistem yang mampu mengenali dan mengklasifikasi seragam loreng tentara Indonesia menggunakan CNN serta mengimplementasikan arsitektur CNN ke ponsel Android. Penelitian ini penting untuk meningkatkan keamanan dan mencegah tindak kecurangan, seperti pencurian informasi dan penyamaran identitas yang sering terjadi di masyarakat. Berdasarkan penelitian yang dilakukan, model arsitektur CNN yang dikembangkan menunjukkan hasil sebagai berikut: pengujian menggunakan *confusion matrix* menghasilkan akurasi 98%, pengujian menggunakan data latih menghasilkan akurasi 98.3%, pengujian menggunakan kamera 96.4%, dan pengujian dengan data di luar kelas menghasilkan akurasi 54.54%. Model ini dapat bekerja dengan baik pada ponsel Android.

Kata kunci: Seragam Loreng, Tentara Indonesia, Kostrad, *Convolutional Neural Network (CNN)*, Android

# ***The Indonesian National Armed Force (TNI AD) and Strategic Reserve Command (Kostrad) Camouflage Uniforms Classification Using Convolutional Neural Network (CNN)***

## ***Abstract***

*This research discusses the application of Convolutional Neural Network (CNN) methods in classifying Indonesian military camouflage uniforms, specifically TNI AD and Kostrad camouflage uniforms. The objective of this research is to design a system capable of recognizing and classifying Indonesian military camouflage uniforms using CNN and to implement the CNN architecture on Android phones. This research is important for enhancing security and preventing fraudulent activities, such as information theft and identity impersonation, which are common in society. Based on the conducted research, the developed CNN architecture model yields the following results: testing using a confusion matrix yields 98% accuracy, testing with training data yields 98.3% accuracy, testing with a camera yields 96.4% accuracy, and testing with out-of-class data yields 54.54% accuracy. This model performs well on Android phones.*

*Keywords: Camouflage Uniform, Indonesian Military, Kostrad, Convolutional Neural Network (CNN), Android*

# Kata Pengantar

Puji syukur penulis ucapkan atas kehadiran Tuhan yang maha esa atas kasih dan karunia-Nya sehingga penulis mampu menyelesaikan Laporan Tugas Akhir dengan judul “Klasifikasi Seragam Loreng Tentara Indonesia Angkatan Darat dan Komando Cadangan Strategis Angkatan Darat (Kostrad) Menggunakan Metode *Convolutional Neural Network (CNN)*”. Tugas akhir ini dibuat untuk memenuhi syarat kelulusan di jurusan Teknik Elektro, program studi D-IV Teknik Mekatronika, Politeknik Negeri Batam.

Pada kesempatan ini penulis menyampaikan terima kasih atas segala bantuan dan dukungan yang telah diberikan selama proses pembuatan tugas akhir ini kepada:

1. Tuhan Yang Maha Esa atas semua kasih dan karunia-Nya.
2. Kedua orang tua, kakak adik (Kak Irma, Kak Dwyky, Kak Adnes, Ebi, Meta, Palastio), nenek, dan Brendan Denning, atas bantuan, dukungan, dan doa yang tak henti dalam perjalanan penulis menyusun Tugas Akhir.
3. Bapak Uuf Brajawidagda., Ph. D selaku Direktur Politeknik Negeri Batam.
4. Bapak Dr. Budi Sugandi, S.T., M.Eng. selaku Ketua Jurusan Teknik Elektro Politeknik Negeri Batam.
5. Bapak Indra Hardian Mulyadi, S.T., M. Eng. selaku Ketua Program Studi Teknik Mekatronika Politeknik Negeri Batam dan wali dosen penulis.
6. Bapak Diono, S.Tr.T., M.Sc. selaku dosen pembimbing penulis yang telah menyediakan waktu dan pikiran untuk mengarahkan penulis dalam menyelesaikan Tugas Akhir ini.
7. Bapak Fadli Firdaus, S.Pd., M.Pd. dan Bapak Dr. Abdurahma Dwijotomo, S.ST., M.Sc. selaku dosen penguji penulis.
8. Ibu Nadhrah Wivanius, S.Si., M.Si. selaku dosen pengampu mata kuliah Seminar Proposal.
9. Bapak Muhammad Naufal Airlangga Diputra, S.Pd., M.P.H. selaku dosen pengampu mata kuliah Tugas Akhir.
10. Seluruh dosen, laboran, dan administrasi di jurusan Teknik Elektro Politeknik Negeri Batam.
11. Seluruh teman – teman yang telah membantu dan memotivasi penulis.
12. *Last but not least, I want to thank me for believing in me, for doing all this hard work, and for never quitting.*

Penyusunan Tugas Akhir ini jauh dari kata sempurna dan masih banyak kekurangan, oleh karena itu segala kritik dan saran untuk perbaikan dari semua pihak sangat penulis harapkan. Sekian pengantar dari penulis, besar harapan semoga Tugas Akhir ini dapat memberikan manfaat bagi para pembaca.

Batam, 31 Juli 2024



Audrey Marito Hutahaean

# Daftar Isi

Pernyataan Keaslian Tugas Akhir .....	i
Lembar Pengesahan .....	ii
Abstrak .....	iii
<i>Abstract</i> .....	iv
Kata Pengantar .....	v
Daftar Isi .....	vii
Daftar Gambar .....	ix
Daftar Tabel .....	x
Bab 1. Pendahuluan .....	1
1.1. Latar Belakang .....	1
1.2. Rumusan Masalah .....	2
1.3. Tujuan .....	2
1.4. Manfaat .....	3
1.5. Batasan .....	3
Bab 2. Tinjauan Pustaka .....	4
2.1. Pengolahan Citra ( <i>Image Processing</i> ) .....	4
2.2. Arsitektur <i>Neural Network</i> .....	5
2.3. <i>Machine Learning</i> .....	6
2.4. <i>Deep Learning</i> .....	8
2.5. <i>Convolutional Neural Network (CNN)</i> .....	9
2.5.1. <i>Convolutional Layer</i> .....	11
2.5.2. <i>Pooling Layer</i> .....	16
2.5.3. <i>Activation Function / ReLu Layer</i> .....	18
2.5.4. <i>Fully Connected Layer</i> .....	18
2.6. Arsitektur NasNetMobile .....	18
2.7. Tentara Negara Indonesia (TNI) .....	18
2.8. Penelitian yang Relevan .....	21

Bab 3. Metodologi Penelitian .....	35
3.1. Tahapan Penelitian .....	35
3.1.1. Perancangan Perangkat Keras .....	37
3.1.2. Perancangan Perangkat Lunak .....	38
3.2. Alat dan Bahan (Opsional) .....	50
3.3. Pengujian .....	50
Bab 4. Hasil dan Pembahasan .....	51
4.1. Alur Kerja Sistem .....	51
4.2. Hasil Proses <i>Training Model</i> .....	55
4.3. Hasil Pengujian Model Menggunakan <i>Confusion Matrix</i> .....	57
4.4. Hasil <i>Deployment</i> .....	59
4.5. Hasil Pengujian Klasifikasi Menggunakan Data Pelatihan .....	62
4.6. Hasil Pengujian Klasifikasi Menggunakan Kamera .....	63
4.7. Hasil Pengujian Klasifikasi Terhadap Perubahan Jarak .....	65
4.8. Hasil Pengujian Klasifikasi Terhadap Perubahan Waktu .....	69
4.9. Hasil Pengujian Klasifikasi Menggunakan Jenis Seragam Loreng Lain .....	72
Bab 5. Kesimpulan dan Saran .....	79
5.1. Kesimpulan .....	79
5.2. Saran .....	80
Daftar Pustaka .....	81

## Daftar Gambar

Gambar 1 Representasi <i>Neural Network</i> [8] .....	5
Gambar 2 <i>Identity Activation Function</i> .....	6
Gambar 3 Ilustrasi Regresi [9] .....	8
Gambar 4 Struktur <i>Neural Network</i> Pada <i>Deep Learning</i> [11] .....	9
Gambar 5 Tahapan <i>Convolutional Neural Network (CNN)</i> [14] .....	10
Gambar 6 Gambaran Umum <i>Convolutional Neural Network (CNN)</i> [18] .....	11
Gambar 7 Arsitektur Algoritma <i>Convolutional Neural Network (CNN)</i> [9] .....	11
Gambar 8 Ilustrasi Operasi Konvolusi [19] .....	12
Gambar 9 Convolution layer dengan satu buah filter berukuran 5x5 [18] .....	12
Gambar 10 Convolution layer dengan satu buah filter berukuran 5x5 [18] .....	13
Gambar 11 Proses Konvolusi <i>Neural Network</i> [9] .....	14
Gambar 12 Teknik <i>zero padding, stride = 1</i> [18] .....	14
Gambar 13 Proses konvolusi antara citra dengan <i>filter, stride =1</i> [18] .....	14
Gambar 14 Contoh ilustrasi konvolusi 1x1 dengan 64 <i>filter</i> [18] .....	16
Gambar 15 Proses <i>pooling</i> untuk mereduksi dimensi data [18] .....	16
Gambar 16 Teknik <i>max pooling</i> [18] .....	17
Gambar 17 Teknik <i>average pooling</i> [18] .....	17
Gambar 18 Seragam PDL TNI AL, TNI AD, dan TNI AU [23] .....	19
Gambar 19 Seragam Loreng TNI AD [24] dan TNI Kostrad [25] .....	20
Gambar 20 Tahapan Penelitian .....	35
Gambar 21 Perancangan Perangkat Keras .....	37
Gambar 22 Diagram Alir .....	38
Gambar 23 Tahapan Pengumpulan dan Pra Proses Data .....	39
Gambar 24 Tahapan Augmentasi Data .....	40
Gambar 25 Contoh Augmentasi Data [16] .....	40
Gambar 26 Tahapan Evaluasi .....	46
Gambar 27 Tahapan Penyebaran .....	47
Gambar 28 Diagram Alir Kerja Sistem Klasifikasi .....	48
Gambar 29 <i>Use Case</i> untuk <i>training data</i> .....	48
Gambar 30 Diagram Aktivitas Klasifikasi .....	49
Gambar 31 Pembagian data <i>train</i> , data <i>test</i> , dan data <i>validation</i> .....	53
Gambar 32 <i>Model Summary</i> .....	54
Gambar 33 Grafik Hasil Proses <i>Training</i> .....	56
Gambar 34 <i>Confusion Matrix</i> .....	57
Gambar 35 Cara Menambahkan Model TFLite .....	60
Gambar 36 Tampilan Utama Aplikasi .....	61
Gambar 37 Contoh Hasil Klasifikasi .....	61
Gambar 38 Hasil Klasifikasi yang Gagal (Data Pelatihan) .....	63
Gambar 39 Hasil Klasifikasi yang Gagal (Kamera) .....	64

## Daftar Tabel

Tabel 1 <i>State of The Art</i> .....	21
Tabel 2 Speksifikasi Perangkat Keras.....	38
Tabel 3 <i>Available Models on Keras Applications</i> [39].....	42
Tabel 4 Estimasi Biaya .....	50
Tabel 5 Tabel Jumlah Citra .....	51
Tabel 6 Contoh Citra <i>Dataset</i> .....	52
Tabel 7 Tabel <i>Hyperparameter</i> .....	54
Tabel 8 Tabel <i>Training</i> .....	55
Tabel 9 Tabel Pengujian Klasifikasi Menggunakan Data Pelatihan.....	62
Tabel 10 Tabel Pengujian Klasifikasi Menggunakan Kamera .....	63
Tabel 11 Tabel Pengujian Klasifikasi Terhadap Perubahan Jarak .....	65
Tabel 12 Tabel Pengujian Klasifikasi Terhadap Perubahan Waktu .....	69
Tabel 13 Tabel Hasil Klasifikasi Menggunakan Jenis Seragam Loreng Lain .....	72

# Bab 1. Pendahuluan

## 1.1. Latar Belakang

Tentara Nasional Indonesia (TNI) merupakan pilar utama dalam sistem pertahanan dan keamanan negara, dengan tanggung jawab menjaga keutuhan wilayah, keamanan negara, dan melindungi masyarakat Indonesia. Salah satu aspek yang membedakan tentara dari setiap negara adalah seragam loreng yang mereka kenakan. Seragam loreng tentara dilengkapi dengan corak kamuflase untuk menyatu dengan lingkungan sekitar, memudahkan personel dalam tindakan operasional dan pertempuran. Seragam loreng Tentara Nasional Indonesia (TNI) mencerminkan keberagaman geografis Indonesia. Tentara Nasional Indonesia (TNI) terbagi menjadi tiga Angkatan, yaitu Tentara Nasional Indonesia Angkatan Darat (TNI AD), Tentara Nasional Indonesia Angkatan Udara (TNI AU), dan Tentara Nasional Indonesia Angkatan Laut (TNI AL). Setiap angkatan juga memiliki seragam loreng yang berbeda. Selain itu, pasukan khusus yang ada di angkatan juga memiliki seragam loreng yang berbeda, contohnya seragam loreng yang dikenakan TNI Kostrad.

Oleh karena perbedaan ini, identifikasi dan klasifikasi seragam ini dapat dilakukan menggunakan teknologi pengolahan citra digital. Hal ini dapat bermanfaat juga untuk meningkatkan keamanan mengingat banyaknya tindak kecurangan, seperti pencurian informasi dan penyamaran identitas yang sering terjadi di masyarakat. Contohnya, kasus penipuan oleh orang yang menyamar sebagai anggota TNI yang sering terjadi, meresahkan, dan merugikan masyarakat [1]. Untuk itu, perlu dibuat sistem yang dapat mencegah kejadian serupa, seperti sistem yang dijelaskan dalam buku ini.

Perkembangan teknologi pengolahan citra pada era modern ini memberikan kontribusi signifikan dalam mempermudah kegiatan manusia. Pengolahan citra merupakan sebuah proses yang memanfaatkan perangkat komputer untuk mengolah citra menjadi citra yang memiliki kualitas yang lebih baik. Pengolahan citra ini dapat dimanfaatkan untuk pengklasifikasian objek dan pendeteksian objek secara *real-time* menggunakan kamera.

Penelitian ini memanfaatkan teknologi pengolahan citra digital dengan menggunakan metode *deep learning* algoritma *Convolutional Neural Network* (CNN) untuk mengklasifikasi seragam loreng Tentara Indonesia, seperti seragam loreng malvinas Tentara Nasional Indonesia Angkatan Darat (TNI AD) dan seragam loreng Komando Cadangan Strategis Angkatan Darat (Kostrad). Seragam loreng TNI AD dan seragam loreng Kostrad memiliki warna mirip yaitu hijau, yang dimana seragam loreng TNI AD berwarna hijau tua dan seragam loreng Kostrad berwarna hijau lumut muda. Penelitian ini menjadikan dua seragam loreng ini menjadi objek penelitian untuk melihat bagaimana performa model CNN dalam mengklasifikasi

seragam loreng yang memiliki warna mirip. Oleh karena keterbatasan dana, jenis seragam loreng yang dijadikan untuk objek penelitian hanya dua.

*Convolutional Neural Network (CNN)* merupakan algoritma pembelajaran mendalam yang memiliki kemampuan baik dalam pengenalan dan klasifikasi gambar. Setelah model CNN dilatih dan diuji, akan dilakukan *deployment* ke ponsel Android bertujuan untuk membuat sistem pengenalan dan klasifikasi seragam loreng Tentara Nasional Indonesia (TNI) yang dapat diakses di ponsel Android agar lebih mudah dan efektif.

## 1.2. Rumusan Masalah

Berdasarkan latar belakang penelitian, rumusan masalah dari tugas akhir ini adalah sebagai berikut:

1. Bagaimana cara merancang sistem yang mampu mengklasifikasi seragam loreng TNI AD dan Kostrad dengan menggunakan metode *Convolutional Neural Network (CNN)*?
2. Bagaimana cara melakukan *deployment* model *Convolutional Neural Network (CNN)* ke ponsel Android?
3. Bagaimana performa model *Convolutional Neural Network (CNN)* yang sudah dilatih dalam mengklasifikasi seragam loreng TNI AD dan Kostrad?

## 1.3. Tujuan

Adapun tujuan dari penyusunan tugas akhir tersebut adalah:

1. Merancang sebuah sistem yang mampu mengklasifikasi seragam loreng TNI AD dan Kostrad menggunakan metode *Convolutional Neural Network (CNN)*.
2. Merancang sebuah sistem klasifikasi seragam loreng TNI AD dan Kostrad yang mampu diakses di ponsel Android.
3. Menganalisis dan mengevaluasi performa model *Convolutional Neural Network (CNN)* yang sudah dilatih dalam mengklasifikasi seragam loreng TNI AD dan Kostrad.

## 1.4. Manfaat

Manfaat yang didapat dan diharapkan dalam penyusunan tugas akhir ini adalah:

1. Dapat terbentuknya sistem klasifikasi seragam loreng TNI AD dan Kostrad yang membantu mengurangi risiko penipuan dan penyamaran oleh pihak yang tidak bertanggung jawab.
2. Dapat terbentuknya sistem klasifikasi seragam loreng TNI AD dan Kostrad yang dapat diakses di ponsel Android sehingga dapat mempermudah dalam identifikasi seragam.
3. Dapat mengetahui bagaimana performa model *Convolutional Neural Network (CNN)* dalam mengklasifikasi seragam loreng TNI AD dan Kostrad yang dapat dijadikan referensi untuk penelitian selanjutnya.

## 1.5. Batasan

Adapun batasan masalah dalam penyusunan tugas akhir ini adalah:

1. Objek seragam loreng tentara Indonesia yang difokuskan dalam sistem pengklasifikasian adalah seragam loreng malvinas TNI AD dan seragam loreng Kostrad.
2. *Dataset* untuk *training* menggunakan yang diambil dari Google, video, website resmi TNI AD, dan sosial media resmi TNI AD dan Kostrad.
3. Aplikasi hanya bisa dijalankan di ponsel Android dengan sistem operasi Android 11+.
4. Jarak maksimal kamera ponsel Android ke objek adalah 200 cm
5. Jarak minimal kamera ponsel Android ke objek adalah 100 cm
6. Penelitian ini hanya difokuskan untuk klasifikasi gambar atau foto seragam loreng TNI AD dan Kostrad.

## Bab 2. Tinjauan Pustaka

### 2.1. Pengolahan Citra (*Image Processing*)

Citra atau gambar adalah representasi visual dari objek, individu, atau lanskap. Dalam konteks digital, citra diwakili sebagai matriks dua dimensi, di mana fungsi  $(x, y)$  mengacu pada lokasi elemen gambar atau piksel dan memuat nilai intensitas. Citra digital dibagi menjadi  $M$  baris dan  $N$  kolom, dengan  $x$  dan  $y$  sebagai koordinat spasial, dan amplitudo  $f$  pada titik koordinat  $(x, y)$  mewakili tingkat intensitas [2].

Dalam gambar berformat grayscale, setiap piksel memiliki hanya satu nilai intensitas, yang berkisar dari 0 hingga 255. Nilai intensitas rendah (0) menggambarkan warna hitam, sementara nilai tertinggi (255) menggambarkan warna putih. Untuk gambar berwarna, ada tiga komponen nilai: merah (red), hijau (green), dan biru (blue), dengan masing-masing berkisar dari 0 hingga 255.

Pengolahan citra, juga dikenal sebagai *image processing*, adalah teknik yang memanfaatkan komputer untuk meningkatkan kualitas citra, mengekstraksi informasi penting, dan mengintegrasikan elemen citra untuk berbagai keperluan [3]. Pengolahan citra dapat berupa foto atau video, dan terdiri dari tiga tingkatan yang berbeda [4]:

1. **Low-Level Image Processing:** Pengolahan tingkat rendah ini melibatkan operasi sederhana seperti peningkatan kontras dan pengurangan noise pada gambar.
2. **Mid-Level Image Processing:** Pengolahan tingkat menengah, gambar diolah dengan metode yang lebih kompleks, seperti ekstraksi tepi, kontur, dan wilayah gambar.
3. **High-Level Image Processing:** Pengolahan tingkat tinggi, pengolahan citra melibatkan analisis dan interpretasi konten dalam konteks pengambilan keputusan yang lebih kompleks.

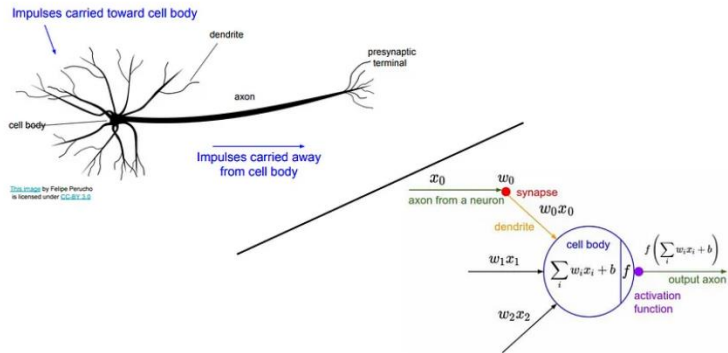
Penerapan teknologi pengolahan citra ini sangat luas, meliputi identifikasi objek, pelacakan objek secara real-time, serta kontribusi dalam berbagai bidang, seperti kecerdasan buatan, machine learning, dan visi komputer. Selain itu, pengolahan citra banyak dimanfaatkan dalam sektor militer untuk berbagai keperluan, termasuk pemantauan, pengenalan objek, serta analisis citra yang canggih. Penelitian ini juga akan memanfaatkan pengolahan citra untuk mengklasifikasi seragam loreng Tentara Indonesia, seperti seragam loreng malvinas TNI AD dan seragam loreng TNI Kostrad. Penelitian ini juga bertujuan untuk mengintegrasikan solusi ini ke dalam platform Android, sehingga pemantauan dan pengenalan seragam dapat dilakukan dengan mudah dan efektif.

## 2.2. Arsitektur Neural Network

*Neural network* atau jaringan saraf adalah model komputasi yang terinspirasi oleh struktur dan fungsi otak manusia. Arsitektur neural network merupakan susunan dari unit-unit pemrosesan informasi, yaitu *neuron*, yang bekerja bersama untuk mengekstraksi pola atau informasi dari data. Salah satu jenis arsitektur yang umum digunakan adalah jaringan saraf tiruan (*artificial neural network* atau ANN) [5].

Komponen utama dari arsitektur neural network melibatkan *neuron/node*, bobot (*weight*), fungsi aktivasi, *layer*, *feedforward*, dan *backpropagation*. *Neuron* merupakan unit dasar yang menerima input, melakukan transformasi menggunakan bobot, dan menghasilkan keluaran melalui fungsi aktivasi. Bobot menentukan kontribusi setiap input terhadap keluaran *neuron* terkait. Fungsi aktivasi diterapkan pada keluaran *neuron* untuk memperkenalkan non-linearitas ke dalam model, memungkinkan pemodelan hubungan yang kompleks [6].

Lapisan (*layer*) digunakan untuk mengelompokkan neuron ke dalam *input layer*, *hidden layers*, dan *output layer*. *Feedforward* adalah proses di mana data mengalir dari input ke keluaran, sedangkan *backpropagation* merupakan teknik pelatihan untuk memperbarui bobot berdasarkan gradien dari fungsi kesalahan, dengan tujuan meminimalkan kesalahan prediksi [7].



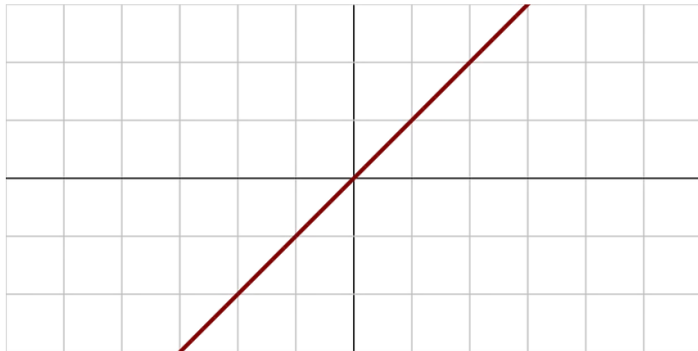
**Gambar 1 Representasi Neural Network [8]**

Gambar 1 merupakan gambar yang merepresentasi *Neural Network*. Pada Gambar 1, terdapat tiga input yang masuk ke dalam *neuron* ( $x_0, x_1, x_3$ ), kemudian dikalikan dengan variabel yang disebut sebagai bobot ( $w_0, w_1, w_2$ ). Setelah itu, hasil perkalian dijumlahkan. Setiap koneksi *neuron* memiliki bobot masing-masing dan nilai bobot dapat berubah selama proses pembelajaran hingga model yang

dihasilkan oleh *Artificial Neural Network* (ANN) mendekati keluaran yang diinginkan [8].

Selanjutnya, bias  $b$  dapat ditambahkan ke hasil penjumlahan tersebut. Nilai bias ini tidak berasal dari lapisan input. Sebagai analogi, bias mirip dengan *intercept* dalam persamaan linear yang ditambahkan untuk menyesuaikan agar perhitungan menjadi lebih akurat [8].

Setelah seluruh penjumlahan tersebut dilakukan, *neuron* akan dimasukkan ke dalam suatu fungsi yang disebut fungsi aktivasi atau *activation function* yang berfungsi untuk mengatur apakah neuron tersebut harus aktif atau tidak [8].



**Gambar 2 Identity Activation Function**

Gambar 2 menunjukkan *identity activation function*  $f(x) = x$ . Dengan demikian, inilah cara kerja *Artificial Neural Network* (ANN), di mana setiap *neuron* saling terhubung untuk menghasilkan keluaran dari input yang diberikan [8].

### **2.3. Machine Learning**

Pembelajaran mesin, juga dikenal sebagai machine learning, merupakan cabang dari kecerdasan buatan (AI) dan ilmu komputer yang berfokus pada penggunaan data dan algoritma untuk meniru proses pembelajaran manusia, dengan tujuan meningkatkan akurasi seiring berjalannya waktu. Machine learning melibatkan analisis sampel data yang telah tersimpan sebelumnya untuk menghasilkan prediksi, dengan kemampuan belajar sendiri sebagai ciri khasnya. Jenis-jenis machine learning mencakup Supervised Learning, Unsupervised Learning, dan Reinforced Learning [9].

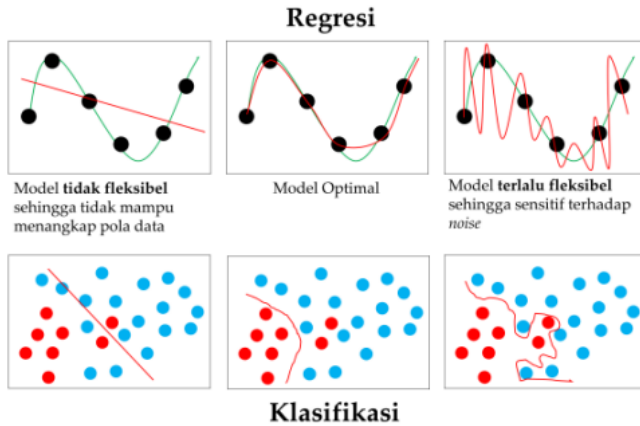
Dalam pengembangan model machine learning, terdapat dua istilah penting yang perlu dipahami, yaitu "training" dan "testing". Training merupakan proses

konstruksi model, sementara testing adalah langkah untuk menguji kinerja model pembelajaran. Dalam konteks *dataset*, *dataset* adalah kumpulan data yang digunakan dalam statistik, dan sampel dalam *dataset* adalah data yang digunakan untuk membangun dan mengevaluasi model machine learning [9].

*Dataset* umumnya dibagi menjadi tiga jenis yang tidak memiliki tumpang tindih, artinya setiap sampel dalam *dataset* tertentu tidak muncul dalam *dataset* lainnya [9]:

1. **Training Set:** Ini adalah himpunan data yang digunakan untuk melatih dan membangun model.
2. **Validation Set:** Ini adalah himpunan data yang digunakan untuk mengoptimalkan model selama proses pelatihan. Model dilatih menggunakan training set, dan biasanya, kinerja model diuji dengan validation set. Ini membantu memastikan bahwa model mampu mengenali pola secara umum.
3. **Testing Set:** Ini adalah himpunan data yang digunakan untuk menguji model setelah proses pelatihan selesai. Dalam buku ini, "testing data" atau "test set" mengacu pada testing set. Penting untuk diingat bahwa testing set berisi data yang tidak pernah dilihat sebelumnya oleh model atau manusia selama proses pelatihan. Melihat testing set selama pelatihan dapat menyebabkan bias yang tidak diinginkan.

Ketiga *dataset* ini berasal dari populasi yang sama dan dihasilkan dari distribusi yang sama. Dengan kata lain, ketiga jenis *dataset* dapat mencerminkan karakteristik yang identik. Oleh karena itu, jika terjadi kerugian atau kesalahan pada tahap *training*, *validation*, dan *testing*, nilainya akan hampir serupa. Meskipun begitu, kemungkinan *underfitting* dan *overfitting* tetap ada. Ilustrasinya dapat dilihat pada gambar 3.

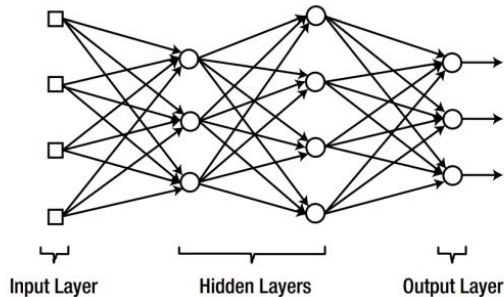


**Gambar 3 Ilustrasi Regresi [9]**

Tantangan utama dalam *machine learning* adalah menghindari underfitting dan overfitting, di mana model memiliki kinerja yang kurang baik atau terlalu baik pada data *training*, *validation*, dan *testing*. Model yang dianggap ideal adalah yang mampu menunjukkan kinerja yang baik pada data pelatihan, validasi, dan pengujian. Dengan kata lain, kita berharap bahwa perbedaan kinerja model pada berbagai *dataset* dapat diminimalkan. Upaya untuk menghindari overfitting atau underfitting dapat dilakukan dengan menambahkan fungsi noise/bias (selanjutnya disebut sebagai noise/bias saja) dan menerapkan teknik regularisasi [9].

## **2.4. Deep Learning**

*Deep learning* atau pembelajaran mendalam merupakan sub bidang dari *machine learning* atau pembelajaran mesin yang menerapkan algoritma *Neural Network* dengan banyak lapisan untuk mempelajari dan memproses data. *Deep learning* bekerja dengan tiga lapisan atau lebih. Setiap lapisan menerima informasi dari lapisan sebelumnya dan meneruskan informasi ke lapisan selanjutnya [10].



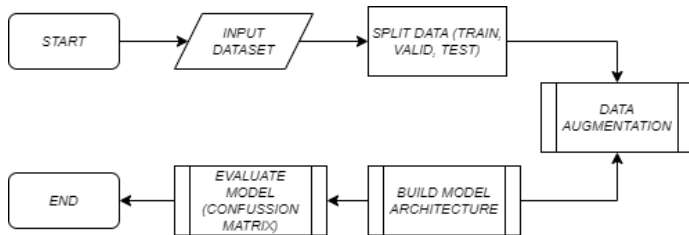
**Gambar 4 Struktur *Neural Network* Pada *Deep Learning* [11]**

Gambar 4 merupakan struktur *neural network* pada *deep learning*. Terdapat tiga lapisan utama, yaitu *input layer*, *hidden layers*, dan *keluaran layer*. *Input layer* atau lapisan masukan memiliki empat *node neuron* yang berfungsi sebagai jalur untuk menstramisikan sinyal masukan ke *node neuron* berikutnya. Dua *hidden layers* atau lapisan tersembunyi sebagai perantara antara *input layer* dan *keluaran layer* yang berfungsi untuk menambah instruksi logika. Peforma *deep learning* dipengaruhi oleh jumlah *neuron* pada lapisan tersembunyi. Model akan menjadi *underfitting* jika jumlah *neuron* yang diterapkan pada arsitektur *deep learning* terlalu sedikit dan *overfitting* jika jumlah *neuron* terlalu banyak. *Keluaran layer* atau lapisan keluaran memiliki tiga *node neuron* yang akan menampilkan hasil akhir dari *neural network*. Dengan memanfaatkan banyak lapisan, *deep learning* dapat mengenali pola dan fitur yang kompleks sehingga sering dimanfaatkan untuk menyelesaikan tugas yang melibatkan data yang sangat kompleks atau besar, seperti deteksi objek, klasifikasi gambar, dan pengenalan suara [12].

## **2.5. Convolutional Neural Network (CNN)**

*Convolutional Neural Network (CNN)* masuk dalam kelas *deep feed-forward Artificial Neural Networks (ANN)*. *Convolutional Neural Network (CNN)* adalah jenis arsitektur jaringan saraf tiruan yang dirancang khusus untuk memproses dan menganalisis data grid, seperti gambar dan video. *CNN* telah menjadi sangat populer dalam tugas-tugas pengenalan pola visual, klasifikasi gambar, deteksi objek, dan tugas-tugas lain yang melibatkan data spasial [13].

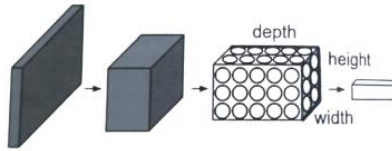
Berbeda dengan jaringan saraf tiruan umum, *CNN* memiliki lapisan-lapisan khusus, seperti lapisan konvolusi dan pooling, yang membantu dalam mengekstraksi fitur-fitur spasial dari data input. Lapisan konvolusi menggunakan filter untuk mendeteksi pola dan fitur-fitur pada berbagai tingkat abstraksi, sementara lapisan pooling mengurangi dimensi data dengan mereduksi resolusi spasial.



**Gambar 5 Tahapan *Convolutional Neural Network* (CNN) [14]**

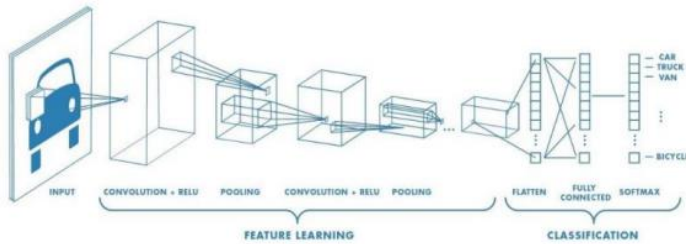
Gambar 5 menunjukkan 6 tahapan CNN, di antaranya adalah tahapan pre-processing, yang melibatkan input data, pembagian data, dan augmentasi data. Data CNN umumnya berupa gambar dan video. Tahapan selanjutnya mencakup pembagian data menjadi kategori train dan valid, yang digunakan untuk menghasilkan loss function dari model. Data test kemudian digunakan untuk menguji model yang telah dibuat [14].

Augmentasi data adalah teknik dalam pembelajaran mesin, khususnya pada pengolahan citra dan data visual, yang bertujuan meningkatkan jumlah dan variasi data pelatihan dengan menciptakan variasi sintesis dari data yang sudah ada. Hal ini membantu meningkatkan kinerja model dan mencegah overfitting dengan memberikan variasi yang lebih banyak pada data pelatihan. Beberapa teknik augmentasi data yang umum digunakan melibatkan perubahan kecil pada data, seperti rotasi, pemotongan, pembalikan horizontal atau vertikal, pergeseran, zoom, dan penambahan noise. Dengan menerapkan teknik ini, variasi tambahan dari setiap gambar asli dapat diciptakan tanpa memerlukan sampel unik tambahan. Augmentasi data bermanfaat untuk melatih model agar lebih umum dan robust, terutama ketika *dataset* terbatas [15] [16]. Selain itu, augmentasi data juga dapat mengatasi masalah ketidakseimbangan kelas di *dataset*. Penerapan augmentasi data umumnya digunakan dalam tugas-tugas seperti klasifikasi gambar, deteksi objek, dan segmentasi gambar, di mana variasi yang lebih besar pada data pelatihan dapat meningkatkan kemampuan model untuk mengenali dan memahami objek dalam berbagai situasi dan kondisi [17]. Setelah tahapan *pre-processing* selesai, langkah berikutnya adalah membuat model arsitektur dan mengevaluasi model menggunakan *confusion matrix*.



**Gambar 6** Gambaran Umum *Convolutional Neural Network (CNN)* [18]

Gambar 6 merupakan gambar umum arsitektur CNN. CNN menggunakan arsitektur tiga dimensi: lebar, tinggi, dan dalam (depth). Setiap lapisan CNN mentransformasikan volume masukan tiga dimensi menjadi volume keluaran tiga dimensi aktivasi sel saraf. Gambar tersebut mengilustrasikan masukan berupa citra berwarna dengan dimensi lebar dan tinggi, serta kedalaman 3 yang mewakili kanal Red, Green, Blue (RGB).



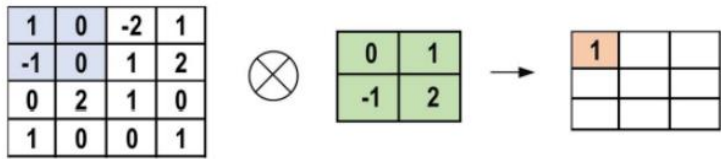
**Gambar 7** Arsitektur Algoritma *Convolutional Neural Network (CNN)* [9]

Gambar 7 merupakan arsitektur algoritma CNN. CNN memiliki tiga lapisan struktur *neural network* yaitu lapisan masukan, lapisan tersembunyi, dan lapisan keluaran. Di dalam lapisan tersembunyi terdapat empat lapisan utama pada arsitektur CNN, yaitu *convolutional layer*, *relu layer*, *pooling layer*, dan *fully connected layer*.

### 2.5.1. *Convolutional Layer*

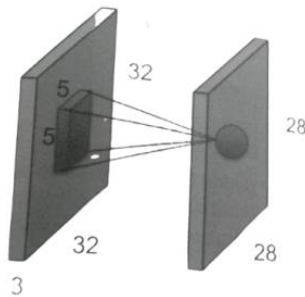
*Convolutional layer* atau lapisan konvolusional merupakan blok bangunan inti pada arsitektur pada CNN yang dimana sebagian besar komputasi dilakukan di lapisan ini. Lapisan ini berisi kumpulan filter konvolusional (kernel) yang bertugas melakukan komputasi dengan menerapkan sebuah fungsi pada fungsi keluaran lainnya secara berulang. Pada gambar *input* dalam gambar, gambar ini dinyatakan

dalam metrik  $N$ -dimensi yang diproses dengan filter untuk menghasilkan peta fitur (*feature map*) keluaran.



**Gambar 8 Ilustrasi Operasi Konvolusi [19]**

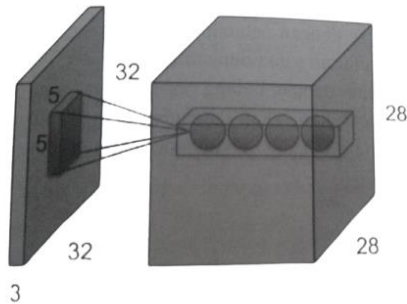
Gambar 8 merupakan ilustrasi operasi konvolusi. Kernel dapat didefinisikan sebagai atau nilai distrik yang terdapat dalam suatu kotak. Setiap nilai pada kotak hijau disebut bobot kernel. Pada awal pelatihan, terdapat nomor acak yang bertindak sebagai bobot kernel. Untuk mengekstraksi fitur – fitur yang penting, bobot tersebut akan disesuaikan pada setiap proses pelatihan. Nilai yang ditandai dengan warna oranye merupakan masuk pada peta fitur keluaran [12].



**Gambar 9 Convolution layer dengan satu buah filter berukuran 5x5 [18]**

Gambar 9 merupakan ilustrasi *convolutional layer* dengan satu buah filter berukuran 5x5. Misalkan kita membangun satu *convolutional layer* dengan satu lembaran *neuron* yang berisi 28x28. Setiap lembaran terhubung dengan suatu area kecil dalam citra masukan, contohnya adalah 5x5 (piksel), yang merupakan bidang reseptif untuk setiap neuron. Hal ini menunjukkan bahwa filter yang digunakan memiliki ukuran 5x5. Seluruh bidang reseptif akan ditelusuri dengan tumpang tindih parsial, sehingga semua neuron dalam proses tersebut pasti berbagi bobot koneksi (*weight sharing*) [18]. Bidang reseptif adalah konsep yang umum digunakan dalam konteks jaringan saraf konvolusional dan digunakan untuk menggambarkan area di input yang mempengaruhi aktivitas suatu neuron dalam jaringan tersebut. Bidang reseptif merujuk pada wilayah spasial di input data yang memengaruhi nilai aktivasi suatu neuron tertentu. Dalam jaringan saraf

konvolusional, setiap neuron pada suatu lapisan memiliki bidang reseptif yang ditentukan oleh ukuran filter (kernel) yang digunakan pada lapisan tersebut. Saat filter bergerak melintasi input, bidang reseptif suatu neuron mencakup area di input yang diproses oleh filter tersebut.



**Gambar 10 Convolution layer dengan satu buah filter berukuran 5x5 [18]**

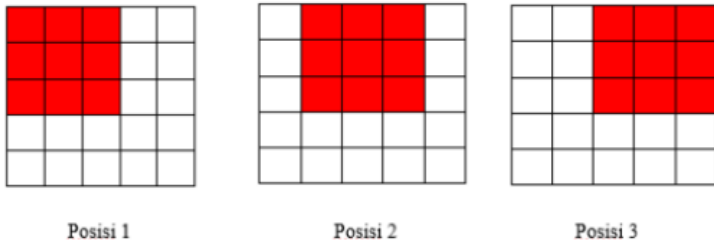
Gambar 10 merupakan ilustrasi *convolutional layer* dengan empat filter berukuran 5x5. Pada umumnya, *convolutional layer* dalam arsitektur *CNN* menggunakan lebih dari satu filter. Apabila kita menggunakan empat filter, *convolutional layer* akan terdiri dari sejumlah *neuron* dalam kisi berukuran 28x28x4. Dapat dilihat pada gambar, ada empat *neuron* yang menghadap ke area yang sama pada citra masukan [18].

Pada umumnya, *Convolutional Neural Network (CNN)* menggunakan interval langkah atau *stride* sebesar satu dan *zero padding*. *Padding* atau penambahan tepi merupakan proses menambahkan piksel tambahan di sekitar gambar atau input data, terutama saat menggunakan konvolusi dalam jaringan saraf konvolusional (*CNN*) atau operasi konvolusi pada data spasial. Tujuan utama dari penggunaan *padding* adalah untuk mempertahankan informasi di pinggiran data selama operasi konvolusi [14].

$$P = \frac{F - 1}{2} \tag{2.1}$$

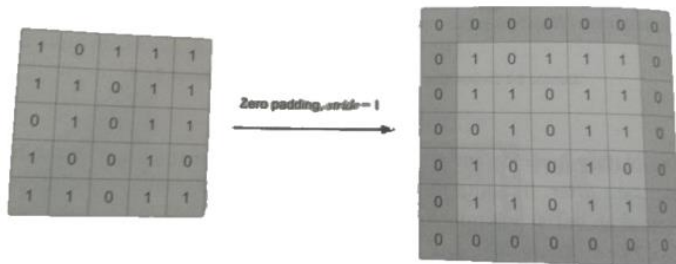
Ukuran *padding* dapat dicari dengan rumus 2.1. Nilai *P* merupakan ukuran *padding* dan *F* merupakan ukuran kernel atau ukuran filter konvolusi yang digunakan. *Stride* atau interval langkah adalah parameter dalam operasi konvolusi yang menentukan sejauh berapa banyak filter atau kernel bergerak melintasi input data setiap kali operasi konvolusi dilakukan. *Stride* mengontrol langkah atau jarak antara satu posisi filter dengan posisi filter berikutnya. Jika nilai *stride* adalah satu, maka filter bergeser secara horizontal dan vertikal sebanyak satu piksel. Nilai *stride*

yang kecil dapat memberikan keluaran dengan resolusi spasial lebih tinggi dan kebutuhan komputasi lebih besar dibandingkan nilai *stride* yang besar [12] [14].

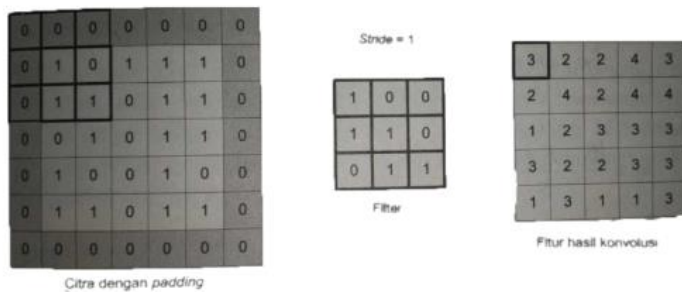


**Gambar 11 Proses Konvolusi *Neural Network* [9]**

Gambar 11 merupakan ilustrasi proses operasi konvolusi dengan menggunakan ukuran filter 3x3 dan *stride* satu.



**Gambar 12 Teknik *zero padding, stride = 1* [18]**



**Gambar 13 Proses konvolusi antara citra dengan *filter, stride = 1* [18]**

Gambar 12 merupakan contoh *zero padding* dengan *stride* sebesar satu. Gambar 13 merupakan proses konvolusi antara citra dengan *filter* menggunakan *stride* sebesar satu. Pada dasarnya, konvolusi merupakan produk dot antara citra masukan dengan filter atau kernel. Dapat dilihat gambar, operasi konvolusi citra dengan filter dimulai dari bidang reseptif (citra masukan) dengan ukuran 3x3 bagian kiri atas.

$$(1 \times 1) + (1 \times 1) + (1 \times 1) = 3 \quad (2.2)$$

Persamaan 2.2 merupakan operasi konvolusi pertama yang menghasilkan nilai 3 dan diletakkan di bagian kiri atas matriks fitur hasil konvolusi. Lalu, operasi konvolusi terus dilakukan dengan menggeser ke kanan dengan *stride* sebesar satu sampai ke seluruh bidang reseptif. Pada umumnya, *stride* yang digunakan sebesar satu.

$$w_1 \times h_1 \times d_1, \quad (2.3)$$

Ukuran atau dimensi sebuah citra masukan dapat dinyatakan seperti persamaan. Dalam persamaan,  $w_1$  adalah lebar citra masukan,  $h_1$  adalah tinggi citra masukan, dan  $d_1$  adalah kedalaman atau jumlah kanal citra masukan.

$$w_2 \times h_2 \times d_2, \quad (2.4)$$

dimana

$$w_2 = \frac{(w_1 - F + 2P)}{S} + 1, \quad (2.5)$$

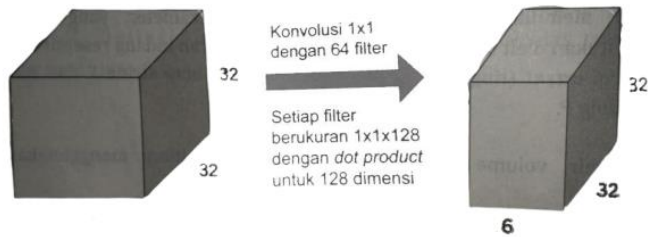
$$h_2 = \frac{(h_1 - F + 2P)}{S} + 1, \quad (2.6)$$

$$d_2 = K, \quad (2.7)$$

*Convolutional Neural Network (CNN)* memiliki empat *hyperparameter* atau parameter yang harus ditentukan oleh *user* sebelum proses pemodelan dan pelatihan data. Empat *hyperparameter* tersebut adalah jumlah filter  $K$ , ukuran bidang reseptif atau *spatial extent*  $F$ , *stride*  $S$ , dan *zero padding*  $P$ . Untuk menghitung dimensi citra keluaran dapat dihitung dengan formula.

Pada umumnya arsitektur *Convolutional Neural Network (CNN)* menggunakan *hyperparameter* sebagai berikut:

1. Jumlah filter  $K$  berupa bilangan pangkat dua, misalkan 32, 64, 128, 512
2. Ukuran filter yang biasanya digunakan,
  - $F = 3, S = 1, P = 1,$
  - $F = 5, S = 1, P = 2,$
  - $F = 5, S = 2, P = \text{berapapun yang sesuai},$
  - $F = 1, S = 1, P = 0$  (*konvolusi*  $1 \times 1$ ).

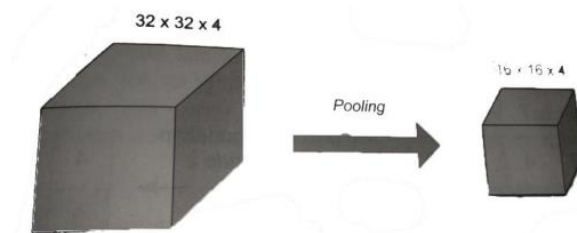


**Gambar 14** Contoh ilustrasi konvolusi 1x1 dengan 64 filter [18]

Gambar 14 merupakan contoh konvolusi 1x1 dengan 64 filter. Dalam beberapa kasus, *Convolutional Neural Network (CNN)* umumnya didesain untuk tidak mereduksi ukuran data terlalu cepat. Mereduksi ukuran data terlalu cepat dapat menyebabkan penurunan performa *Convolutional Neural Network (CNN)*. Jadi, *Convolutional Neural Network (CNN)* terkadang menggunakan konvolusi 1x1 untuk mereduksi kedalaman data tetapi tetap mempertahankan lebar dan tinggi data.

### 2.5.2. Pooling Layer

*Pooling layer* merupakan lapisan dalam arsitektur *Convolutional Neural Network (CNN)* yang bertugas untuk mereduksi dimensi spasial dari peta fitur keluaran dari lapisan konvolusi sebelumnya. *Pooling layer* membantu mempertahankan informasi yang paling penting dari peta fitur dan memberikan representasi yang lebih ringkas tetapi tetap mewakili fitur-fitur penting dalam data. Dengan *pooling* juga, data dapat dipresentasikan lebih kecil, mudah dikelola, dan mudah mengontrol *overfitting* [12] [14].



**Gambar 15** Proses *pooling* untuk mereduksi dimensi data [18]

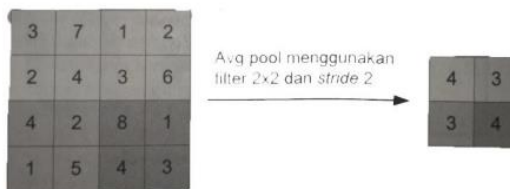
Gambar 15 merupakan proses *pooling* dalam mereduksi dimensi data. Data awalnya berdimensi 32x32x4, melalui proses *pooling*, dimensi data direduksi

menjadi 16x16x4. Metode *pooling* yang paling sering digunakan adalah *max pooling* dan *average pooling*. *Max pooling* dapat dilakukan dengan cara memilih nilai maksimum dan *average pooling* dilakukan dengan cara memilih nilai rata – rata dalam suatu area tertentu.



**Gambar 16 Teknik *max pooling* [18]**

Gambar 16 merupakan ilustrasi *max pooling* menggunakan filter 2x2 dan *stride* 2. Pada gambar, peta fitur pertama dimulai dari bagian kiri atas. Di antara empat nilai tersebut, nilai maksimum adalah 7. Dengan melakukan *stride* sebanyak dua piksel secara horizontal (bagian kanan atas), di antara empat nilai tersebut, nilai maksimum adalah 6. *Stride* secara vertical ke bawah sebanyak dua piksel (bagian kanan bawah), diantara empat nilai tersebut, nilai maksimumnya adalah 8. *Stride* sebanyak dua piksel secara horizontal ke kiri (bagian kiri bawah), diantara empat nilai tersebut, nilai maksimumnya adalah 5.



**Gambar 17 Teknik *average pooling* [18]**

Gambar 17 merupakan ilustrasi *average pooling* menggunakan filter 2x2 dengan *stride* 2. Keluaran dihasilkan melalui proses yang sama dengan *max pooling*, tetapi nilai yang diambil merupakan nilai rata – rata. Sebagai contoh, dari bagian kiri atas, nilai rata – rata dari keempat nilai adalah 4. Nilai tersebut didapat dengan menjumlahkan keempat nilai lalu membaginya dengan empat.

### 2.5.3. Activation Function / ReLU Layer

Secara umum, ReLU (Rectified Linear Unit) bukanlah lapisan (layer) sendiri, melainkan fungsi aktivasi yang sering digunakan di dalam lapisan-lapisan jaringan saraf. Pada *Convolutional Neural Network (CNN)*, ReLU biasanya diterapkan sebagai fungsi aktivasi pada keluaran dari lapisan-lapisan konvolusi atau lapisan-lapisan tersembunyi [12].

$$f(x) = \max(0, x) \quad (2.8)$$

Persamaan 2.8. merupakan persamaan fungsi ReLU yang dimana setiap nilai  $f(x)$  yang positif, keluarannya adalah  $x$ , dan untuk setiap nilai  $x$  yang negatif, keluarannya adalah 0. Keuntungan dari fungsi aktivasi ReLU adalah beban komputasi yang lebih rendah [12].

### 2.5.4. Fully Connected Layer

Proses akhir dalam arsitektur adalah *fully connected layer*, yang bertujuan untuk menghubungkan representasi fitur yang diekstraksi oleh lapisan konvolusi dan pooling ke lapisan keluaran, terutama untuk tugas klasifikasi. Sebelum melanjutkan ke langkah *fully connected layer*, keluaran dari peta fitur masih berupa *array multidimensional*, sehingga perlu dilakukan *flatten* atau transformasi peta fitur menjadi vektor, yang akan menjadi masukan untuk *fully connected layer*. Pada lapisan ini, semua neuron aktivasi dari lapisan sebelumnya terhubung ke neuron di lapisan selanjutnya, mirip dengan jaringan syaraf tiruan biasa. Lapisan ini menghasilkan keluaran berupa klasifikasi citra yang diinginkan [12] [14].

## 2.6. Arsitektur NasNetMobile

Dikembangkan oleh Google Inc., NASNetMobile adalah sebuah varian dari arsitektur NASNet dengan jumlah parameter yang lebih sedikit dibandingkan dengan arsitektur MobileNet. Ini menonjol sebagai salah satu varian arsitektur NASNet yang dirancang khusus untuk sistem terbenam dan perangkat seluler [20].

## 2.7. Tentara Negara Indonesia (TNI)

Tentara adalah warga negara yang dipersiapkan dan dipersenjatai untuk melakukan tugas pertahanan negara dalam menghadapi ancaman militer maupun ancaman non-militer. Tentara Negara Indonesia (TNI) merupakan salah satu institusi negara di bidang pertahanan yang memiliki tugas pokok untuk menegakkan kedaulatan negara, mempertahankan keutuhan wilayah Negara Kesatuan Republik Indonesia yang berdasarkan Pancasila dan Undang-Undang Dasar 1945, serta melindungi segenap bangsa dan seluruh tumpah darah Indonesia dari ancaman dan gangguan terhadap keutuhan bangsa dan negara [21].

Tentara Negara Indonesia (TNI) dibagi menjadi tiga sektor, yaitu Tentara Negara Indonesia Angkatan Darat (TNI AD), Tentara Negara Indonesia Angkatan Laut (TNI AL), dan Tentara Negara Indonesia Angkatan Udara (TNI AU). Dengan adanya tiga sektor ini, maka seragam yang digunakan oleh para anggota tentara setiap sektor ada berbagai jenis sesuai lingkungan dan tugas yang akan dilaksanakan [21].

Seragam TNI dibagi menjadi tiga jenis, yaitu Pakaian Dinas Harian (PDH), Pakaian Dinas Upacara (PDU), dan Pakaian Dinas Lapangan (PDL) [22]. Seragam PDL merupakan seragam yang menjadi ciri khas dari seorang tentara yaitu seragam bermotif loreng. Seragam bermotif loreng digunakan pada saat tentara melakukan penyaraman atau kamuflase untuk melindungi diri dari musuh di medan perang. Setiap sektor memiliki seragam motif loreng yang berbeda. TNI AD memiliki seragam motif loreng dengan warna hijau, coklat, dan krem. TNI AL, memiliki seragam motif loreng dengan warna dasar abu – abu tua kebiruan ditambah dengan perpaduan warna biru dongker, putih, dan coklat tua. TNI AU memiliki seragam motif loreng dengan warna dasar biru muda dipadukan dengan warna biru tua, putih kebiruan, dan biru dongker [23].



**Gambar 18 Seragam PDL TNI AL, TNI AD, dan TNI AU [23]**



**Gambar 19 Seragam Loreng TNI AD [24] dan TNI Kostrad [25]**

Gambar 18 merupakan gambar seragam PDL tantara setiap sektor. Di dalam setiap Angkatan terdapat pasukan – pasukan khusus yang dibentuk dan memiliki seragam khusus juga. Di Angkatan Darat, terdapat pasukan seperti Komando Cadangan Strategis Angkatan Darat (Kostrad) yang bertugas untuk melakukan operasi gabungan pertahanan udara dan pantai bersama matra lain,serta menyelenggarakan operasi darat berupa operasi serangan dan operasi intelijen. Gambar 19 merupakan seragam loreng TNI AD dan Kostrad yang akan digunakan sebagai objek klasifikasi untuk penelitian tugas akhir ini. Dapat dilihat seragam loreng TNI AD dan seragam loreng Kostrad memiliki warna mirip yaitu hijau, yang dimana seragam loreng TNI AD bewarna hijau tua dan seragam loreng Kostrad bewarna hijau lumut muda. Penelitian ini menjadikan dua seragam loreng ini menjadi objek penelitian untuk melihat bagaimana peforma model CNN dalam mengklasifikasi seragam loreng dengan warna yang mirip. Oleh karena keterbatasan dana, jenis seragam loreng yang dijadikan objek penelitian hanya dua.

## 2.8. Penelitian yang Relevan

Tabel 1 *State of The Art*

No	Tahun	Penulis	Jurnal	Metode	Kelemahan
1	2014	Martins E. Irhebhude, Eran A. Edirisinghe	<i>Military Personnel Recognition System using Texture, Color, and SURF Features</i> [26]	<i>Support Vector Machine (SVM), Texture Features, GLCM &amp; Color Histogram Bin Features, Speed Up Robust (SURF), Correlation-based Feature</i>	Dalam hasil penelitian yang dipaparkan, terdapat kelemahan pada metode yang digunakan untuk mengklasifikasikan seragam kamuflase. Pertama, terlihat adanya variasi dalam tingkat akurasi antara berbagai kombinasi fitur tekstur dan warna (paling rendah pada kombinasi fitur tekstur dan intensitas). Selanjutnya, penurunan akurasi terjadi ketika dilakukan seleksi fitur menggunakan <i>Correlation-based Feature Selection (CFS)</i> , terutama terlihat pada kombinasi fitur tekstur dan intensitas. Sensitivitas terhadap variasi orientasi seragam kamuflase juga merupakan tantangan, meskipun eksperimen dilakukan pada berbagai orientasi.

				<p><i>Selection (CFS)</i></p>	<p>Jumlah sampel pelatihan dan pengujian yang relatif kecil dapat mempengaruhi kemampuan model untuk mengenali variasi yang lebih kompleks pada seragam militer. Penelitian ini juga hanya menggunakan data dari sumber tertentu seperti gambar dari Google dan situs web militer yang terpilih, sehingga hasilnya mungkin tidak berlaku untuk situasi operasional yang lebih luas. Selain itu, penelitian bergantung pada fitur seperti tekstur, saturasi, intensitas, dan warna, yang mungkin tidak selalu efektif dalam menghadapi variasi kompleks pada seragam militer. Oleh karena itu, perlu dilakukan penelitian dan eksperimen lebih lanjut untuk mengatasi kelemahan ini dan meningkatkan keandalan metode tersebut.</p>
--	--	--	--	-------------------------------	--

2	2015	Martins E. Irhebhude, Eran A. Edirisinghe	<i>Personnel Recognition in the Military using Multiple Features</i> [27]	<i>Support Vector Machine (SVM), GLCM (22 Features) &amp; Color Histogram Bin Features, Speed Up Robust (SURF), Correlation-based Feature Selection (CFS)</i>	Berdasarkan hasil eksperimen penelitian dengan berbagai metode kombinasi fitur tekstur <i>GLCM</i> dan fitur histogram warna untuk klasifikasi seragam loreng tentara, nilai akurasi yang dihasilkan berbeda – beda dan nilai akurasi paling rendah yaitu pada kombinasi fitur tekstur <i>GLCM</i> dengan fitur intensitas pada warna (77%). Dengan adanya variasi akurasi yang berbeda, hal ini menunjukkan bahwa metode yang diusulkan tidak selalu dapat mengenali dengan baik dalam berbagai situasi atau kondisi yang berbeda, seperti perubahan pencahayaan, sudut pandang, atau variasi lainnya. Penggunaan berbagai fitur seperti tekstur dan warna membuat metode yang digunakan menjadi lebih rumit. Kerumitan ini bisa mempengaruhi kinerja sistem dalam dua hal utama: kecepatan pemrosesan (sistem mungkin menjadi lebih lambat) dan keandalan (sistem
---	------	---	---	---	---

					<p>mungkin menjadi kurang konsisten atau akurat). Penelitian ini juga hanya menggunakan data dari sumber tertentu seperti gambar dari Google dan situs web militer yang terpilih, sehingga hasilnya mungkin tidak berlaku untuk situasi operasional yang lebih luas. Selain itu, penelitian bergantung pada fitur seperti tekstur, saturasi, intensitas, dan warna, yang mungkin tidak selalu efektif dalam menghadapi variasi kompleks pada seragam militer. Oleh karena itu, perlu dilakukan penelitian dan eksperimen lebih lanjut untuk mengatasi kelemahan ini dan meningkatkan keandalan metode tersebut.</p>
3	2017	Hasbi Naufal Ashshidiqi, Suprayogi, Hertiana Bethaningtyas	Identifikasi Seragam Personel Militer Menggunakan	Ekstraksi Ciri Warna (Penyebaran <i>Hue Color</i> ) & Ekstraksi Ciri Bentuk	Berdasarkan hasil pengujian data ekstraksi ciri warna, pengujian dengan 80 data seragam loreng malvinas menghasilkan akurasi sebesar 92,5% dengan 6 data <i>error</i> . Pengujian untuk 35 data seragam loreng lain menghasilkan akurasi sebesar

		Dyah Kusumaningrum	<i>Image Processing</i> [28]	( <i>Eccentricity</i> dan <i>Metric</i> )	20% dengan 28 data <i>error</i> . Hal ini menunjukkan bahwa metode ekstraksi fitur warna tidak baik dalam membedakan seragam loreng lain dengan seragam loreng malvinas. Hasil pengujian data ekstraksi ciri bentuk menghasilkan nilai akurasi yang lebih baik daripada ekstraksi ciri warna, tetapi untuk pola bentuk dua seragam malvinas nilai akurasinya kurang memuaskan. Selain itu, pada saat pola bentuk dua ini diuji dengan seragam loreng lain, nilai akurasinya mengalami peningkatan. Hal ini menunjukkan bahwa metode ini untuk uji pola bentuk dua kurang memuaskan dalam mengenal pola bentuk dua itu sendiri. Berdasarkan hasil pengujian sistem terhadap perubahan jarak saat pengambilan data dengan akurasi/resolusi data citra yang berbeda – beda sesuai jarak menunjukkan bahwa semakin jauh jarak dan semakin kecil ukuran/resolusi citra, sistem
--	--	--------------------	------------------------------	---	---

					semakin tidak dapat mengenali data. Dengan adanya berbagai kelemahan yang terdapat pada metode penelitian ini, maka diperlukan penelitian lebih lanjut untuk mengatasi kelemahan yang ada.
4	2017	Hertiana Bethaningtyas, Hasbi Naufal, Gama Wisnu Fajarianto	Pengenalan Jenis Seragam Loreng TNI Menggunakan Kombinasi <i>Eccentricity</i> Dan <i>Metric</i> [29]	<i>Eccentricity</i> dan <i>Metric</i>	Berdasarkan hasil pengujian data ekstraksi ciri warna, pengujian dengan 80 data seragam loreng malvinas menghasilkan akurasi sebesar 92,5% dengan 6 data <i>error</i> . Pengujian untuk 35 data seragam loreng lain menghasilkan akurasi sebesar 20% dengan 28 data <i>error</i> . Hal ini menunjukkan bahwa metode ekstraksi fitur warna tidak baik dalam membedakan seragam loreng lain dengan seragam loreng malvinas. Maka dari itu, diperlukan penelitian lebih lanjut untuk mengatasi kelemahan dari metode ini.

5	2018	Manuel Eugenio Morocho Cayamcela, and Wansu Lim	<i>Learning The Patterns of Soldier Uniforms from The Weights of a Semantic Segmentation Network</i> [30]	VGG-16, Segmentation Network	<p>Penelitian ini membuat <i>segmentation network</i> melalui algoritma CNN model arsitektur VGG-16. Untuk membangun jaringan, <i>dataset</i> untuk <i>training</i> dibagi menjadi dua kelas, yaitu tentara dan latar belakang. Dari statistik kumpulan data, 23% gambar berisi kelas tentara dan 77% piksel sisanya berisi latar belakang. Hasil simulasi menghasilkan akurasi untuk tentara sekitar 81% dan latar belakang sekitar 78%. Walaupun, hasil akurasi ini menjanjikan tetapi perlu diperhatkjan bahwa data yang digunakan tidak seimbang. Peningkatan bobot kelas memang dapat mengurangi ketidakseimbangan kelas tetapi model dapat meningkatkan risiko overfitting. Selain itu, menggunakan akurasi saja mungkin tidak cukup informatif dalam kasus distribusi kelas yang tidak seimbang. Metrik seperti precision, recall, dan F1-score dapat memberikan</p>
---	------	---	---	------------------------------	---

					pemahaman yang lebih baik tentang kinerja model.
6	2019	Manuel Eugenio Morocho Cayamcela, Wansu Lim	<i>Research On Discriminating the Patterns of Soldier Uniforms Using Deep Learning</i> [31]	<i>Deep Learning</i>	Penelitian ini membuat <i>segmentation network</i> melalui algoritma CNN model arsitektur VGG-16. Untuk membangun jaringan, <i>dataset</i> untuk <i>training</i> dibagi menjadi dua kelas, yaitu tentara dan latar belakang. Dari statistik kumpulan data, 23% gambar berisi kelas tentara dan 77% piksel sisanya berisi latar belakang. Hasil simulasi menghasilkan akurasi untuk tentara sekitar 81% dan latar belakang sekitar 82%. Walaupun, hasil akurasi ini menjanjikan tetapi perlu diperhatikan bahwa data yang digunakan tidak seimbang. Peningkatan bobot kelas memang dapat mengurangi ketidakseimbangan kelas tetapi model dapat meningkatkan risiko overfitting. Selain itu, menggunakan akurasi saja mungkin tidak cukup informatif dalam kasus distribusi kelas yang tidak

					seimbang. Metrik seperti precision, recall, dan F1-score dapat memberikan pemahaman yang lebih baik tentang kinerja model. Lalu, penerapan metode dalam pengaturan real-time atau dengan sumber daya terbatas dapat menjadi tantangan yang signifikan.
7	2019	Risda Monitawati, Drs. Suwandi, M.Si, Hertiana Bethaningtyas Dyah K. ST., MT	<i>Tracking Seragam Militer Berbasis Image Processing Secara Real Time</i> [32]	Ekstraksi Ciri Warna dan Bentuk (Algoritma SIFT)	Berdasarkan hasil pengujian dengan menggunakan 21 data citra, dimana terdapat variasi nilai pixel maksimum dan minimum, metode yang dijelaskan sebelumnya mencapai tingkat akurasi sebesar 80%. Meskipun secara keseluruhan memberikan hasil yang memuaskan, terdapat 5 dari 25 data citra yang mengalami kesalahan identifikasi. Kelemahan utama metode ini dapat diidentifikasi dari data error yang menunjukkan bahwa ada citra yang tidak dapat teridentifikasi atau tidak sesuai dengan parameter data latih. Potensi kelemahan dapat berasal dari

					<p>keterbatasan metode dalam mengatasi variasi yang signifikan dalam nilai pixel maksimum dan minimum, terutama pada citra-citra tertentu. Selain itu, hasil yang kurang optimal mungkin juga dipengaruhi oleh kecocokan parameter yang tidak optimal atau kurangnya representasi variasi yang memadai dalam <i>dataset</i> pelatihan.</p>
8	2021	Manuel Eugenio Morocho-Cayamcela & Wansu Lim	<p><i>Pattern Recognition of Soldier Uniforms with Dilated Convolutions And A Modified Encoder-Decoder Neural Network Architecture</i> [33]</p>	VGG-16, Encoder-Decoder, Segmentation Network	<p>Beberapa kelemahan yang perlu diperhatikan. Pertama, terdapat ketidakseimbangan kelas pada data <i>ground truth</i> yang digunakan untuk melatih jaringan segmentasi. Hal ini dapat mengakibatkan bias dalam model, mengingat hanya 23% dari gambar mengandung kelas "tentara" sementara 77% mengandung kelas "latar belakang." Selanjutnya, meskipun jaringan segmentasi mencapai akurasi di atas 80% (81.49% tentara dan 82.64% latar belakang), masih terdapat perbedaan</p>

					<p>antara hasil segmentasi dan <i>ground truth</i> yang diharapkan. Selain itu, akurasi yang dihasilkan hasil segmentasi dengan model lainnya rendah, yaitu di bawah 79% bahkan ada yang sekitar 50%. Ini menunjukkan adanya keterbatasan dalam melakukan segmentasi dengan akurat pada semua bagian gambar. Penggunaan gambar berukuran tetap untuk setiap patch juga menjadi kendala, karena dapat membatasi performa model untuk menangani gambar dengan berbagai ukuran. Hal ini dapat memengaruhi kinerjanya dalam skenario dunia nyata dengan dimensi gambar yang beragam. Selain itu, ketergantungan pada model <i>pre-trained</i> seperti VGG-16 untuk mentransfer bobot yang telah dipelajari ke jaringan segmentasi dapat membatasi performa model terhadap <i>dataset</i> baru, memengaruhi</p>
--	--	--	--	--	--

					kemampuannya untuk mengenali data yang belum pernah dilihat sebelumnya.
9	2022	Benjamin Dubetsky, Kevin Fernandez, Garrett Christopher, Lakhan Singh, Jason Hughes, Jeremy Cole, Michael Novitzky	<i>Military Uniform Identification for Search and Rescue (SAR) through Machine Learning</i> [34]	<i>You Only Look Once (YOLO)</i>	Salah satu kelemahan pada penelitian ini adalah terbatasnya data pelatihan yang tersedia untuk melatih model. Kel data ini mendorong peneliti untuk membuat <i>dataset</i> mereka sendiri, yang mungkin tidak seberagam yang diinginkan. Kelemahan lainnya terletak pada kesulitan model dalam mengklasifikasikan warga sipil (Kelas 3) secara akurat. Model sering kali salah mengklasifikasikan warga sipil sebagai personel militer (Kelas 1 atau Kelas 2) karena kurangnya contoh pelatihan untuk Kelas 3 dalam <i>dataset</i> , menyebabkan akurasi yang lebih rendah dalam mengidentifikasi warga sipil.
10	2023	Ilkay Karatepe, Vasif Nabiyev	<i>Military Camouflage Classification</i>	<i>Mask R-CNN</i>	Penelitian ini model yang dibuat memprediksi dengan rata – rata akurasi 95%. Meskipun hasil penelitian

			with Mask R-CNN Algorithm [35]		menunjukkan tingkat keberhasilan yang tinggi dalam klasifikasi dan segmentasi, hasil pengujian yang disajikan tidak begitu rinci sehingga diperlukan lagi analisis mendalam sendiri untuk mengetahui faktor – faktor yang mempengaruhi performa model.
--	--	--	--------------------------------	--	--

Tabel 1 merupakan *state of the art* atau tabel sumber literatur yang relevan. Berdasarkan tabel tersebut, penelitian yang menerapkan metode deep learning, seperti *Convolutional Neural Network (CNN)*, menunjukkan hasil akurasi yang lebih baik dibandingkan dengan metode ekstraksi fitur tekstur dan warna. Hal ini menunjukkan bahwa metode deep learning efektif untuk klasifikasi gambar. Selain itu, tabel tersebut juga menunjukkan bahwa belum ada penelitian yang menggunakan metode CNN dengan arsitektur NasNetMobile yang dapat diakses melalui ponsel Android.

*Convolutional Neural Network (CNN)* adalah salah satu metode yang paling banyak digunakan dan terbukti efektif dalam berbagai tugas pengenalan gambar, seperti klasifikasi, deteksi objek, dan segmentasi. Metode ini dikenal karena kemampuannya dalam menangani variasi yang kompleks dalam data gambar dan kemampuannya untuk mempelajari representasi yang lebih abstrak dan hierarkis dari fitur-fitur visual. CNN memiliki kelebihan yaitu dapat secara otomatis mengekstraksi ciri-ciri penting dari setiap gambar tanpa bantuan manusia. CNN juga lebih efisien dalam penggunaan memori dan kompleksitas dibandingkan dengan metode jaringan saraf lainnya. Namun, kekurangan metode ini adalah memerlukan banyak data latih, proses pelatihan yang memakan waktu lama, dan risiko *overfitting* [36] [37] [38].

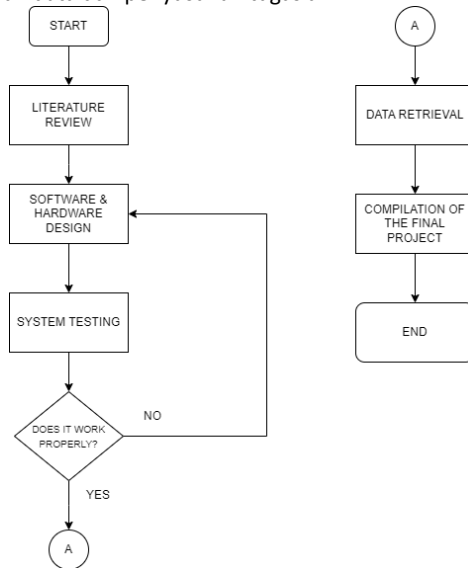
Kelebihan lainnya adalah tersedianya model CNN yang sudah ada atau telah dilatih sebelumnya (pre-trained) dalam ukuran kecil, sehingga dapat digunakan dalam implementasi pada ponsel, contohnya NasNetMobile. NasNetMobile adalah salah satu arsitektur CNN yang telah terbukti efektif dalam tugas-tugas pengenalan gambar [20]. Keberhasilannya dalam mengatasi tantangan dalam pengenalan gambar menjadikan NasNetMobile pilihan menarik untuk diterapkan dalam aplikasi pengenalan

gambar di berbagai platform, termasuk ponsel Android. Diharapkan bahwa dengan metode yang diusulkan, model ini dapat menunjukkan kinerja yang lebih baik dibandingkan dengan penelitian sebelumnya. Selain itu, dengan metode yang diusulkan, klasifikasi seragam loreng Malvinas TNI AD dan Kostrad dapat diakses dengan mudah menggunakan ponsel Android. Hal ini tidak hanya memberikan kemudahan akses bagi pengguna, tetapi juga memanfaatkan potensi teknologi yang tersedia untuk memastikan penggunaan yang efisien dan praktis dalam pengenalan seragam tersebut.

## Bab 3. Metodologi Penelitian

### 3.1. Tahapan Penelitian

Metodologi penelitian adalah suatu pendekatan atau serangkaian langkah-langkah sistematis yang digunakan untuk merancang, melaksanakan, dan mengevaluasi suatu penelitian. Metodologi ini membantu peneliti untuk mengumpulkan data dengan cara yang terstruktur, menjalankan analisis yang sesuai, dan menghasilkan temuan yang dapat diandalkan. Dalam penelitian ini ada beberapa tahapan penelitian yang dimulai dari studi literatur dari penelitian sebelumnya yang relevan, perancangan perangkat lunak dan keras, pengujian sistem, pengambilan data dan penyusunan tugas akhir.



**Gambar 20 Tahapan Penelitian**

Tahapan penelitian dapat dilihat pada gambar 20 yang dimana ada langkah – langkah dari awal studi literatur hingga penyusunan tugas akhir.

1. *Literature Review* / Studi literatur.

Pada tahap ini, dilakukan analisis terhadap literatur, jurnal, buku, laporan, dan penelitian sebelumnya yang relevan dengan penelitian saat ini. Tujuannya agar dapat mencari metode yang lebih baik untuk diterapkan dalam menyelesaikan penelitian. Metode yang dipilih kemudian diimplementasikan pada sistem yang harus

kompatibel dengan perangkat lunak dan keras yang akan digunakan. Pemilihan metode juga berperan dalam merancang perangkat lunak dan keras yang akan digunakan.

2. *Software and Hardware Design* / Perancangan perangkat keras dan lunak

Pada tahap ini, dilakukan perancangan perangkat lunak dengan merinci bagaimana sistem perangkat lunak akan diimplementasikan dan cara kerja dari sistem sesuai dengan spesifikasi metode yang diterapkan. Selain itu, perancangan perangkat keras juga dilakukan dengan merinci perangkat keras yang akan digunakan yang mendukung kinerja perangkat lunak, sehingga sistem dapat berjalan secara optimal dan sesuai dengan kebutuhan.

Pada tahap ini, dibuat perancangan perangkat lunak untuk program dengan menggunakan aplikasi web jupyter notebook dan Android Studio. Jupyter Notebook menggunakan bahasa pemrograman Python sedangkan Android Studio menggunakan bahasa pemrograman Java. Program akan diawali dengan *training dataset* objek seragam loreng TNI AD menggunakan beberapa *library* untuk *machine learning* di python, seperti tensorflow, keras, scikit-learn, dan yang lainnya. *Training data* ini akan menghasilkan sebuah model dari arsitektur CNN. Setelah itu, model ini diimplementasikan ke ponsel Android menggunakan Android Studio agar sistem klasifikasi seragam loreng TNI AD bisa digunakan di ponsel Android.

3. *System Testing* / Pengujian Sistem

Tahap ini sistem yang telah dirancang akan diuji. Jika sistem bekerja dengan baik, maka akan diambil datanya dan dianalisis. Jika sistem tidak bekerja dengan baik, maka akan dimulai ulang ke perancangan perangkat lunak dan keras.

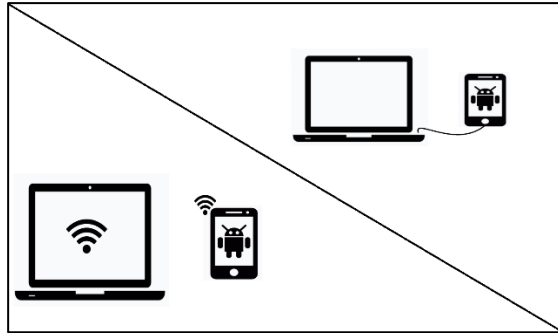
4. *Data Retrieval* / Pengambilan data

Pada tahap ini dilakukan pengambilan data sesuai dengan tabel pengujian. Setelah itu, data tersebut dianalisis untuk mengetahui apakah sistem dapat bekerja dengan baik dan tepat.

5. Penulisan tugas akhir

Pada tahap ini, dilakukan penyusunan tugas akhir berupa penulisan hasil dan kesimpulan penelitian dari sistem dan metode yang telah diuji dan diambil datanya.

### 3.1.1. Perancangan Perangkat Keras



**Gambar 21 Perancangan Perangkat Keras**

Gambar 21 merupakan perancangan perangkat keras sistem klasifikasi seragam tentara Indonesia. Perangkat keras yang digunakan adalah laptop dan ponsel Android. Ponsel ini digunakan untuk mengambil foto untuk *dataset* dan untuk pengujian.

Untuk mengakses aplikasi model yang telah dibuat pada Android Studio, ponsel ini disambungkan ke Android Studio dalam laptop melalui Wi-Fi atau kabel USB. Untuk menghubungkan dengan Wi-Fi, pastikan ponsel Android dan laptop terhubung dengan Wi-Fi yang sama.

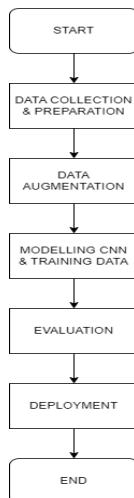
- Menu *Developer Options* perlu untuk diaktivasi dengan cara ketuk menu *Settings*, lalu ketuk *About phone*, ketuk *Software Information*, ketuk *Build Number* sebanyak tujuh kali untuk mengaktivasi menu *Developer Options*, masukan PIN atau *password* ponsel, menu *Developer Options* telah aktif.
- Setelah aktif, ketuk menu *Developer Options*, ketuk menu *Wireless Debugging*, lalu ketuk menu *Pair using QR code* untuk menghubungkan melalui *scan QR barcode* yang ada di Android Studio atau ketuk menu *Pair using pair code* untuk menghubungkan melalui kode yang ada di Android Studio. Jika berhasil terkoneksi, aplikasi klasifikasi dapat dijalankan di ponsel Android.

**Tabel 2 Speksifikasi Perangkat Keras**

No	Perangkat Keras	Speksifikasi	
1	Laptop	Nama	Acer Nitro 5 AN515-57
		RAM	8 GB
		SSD	512 GB
		Processor	11th Gen Intel® Core™ i5-11400H
		Kartu Grafis	NVIDIA® GeForce RTX™ 3050
2	Ponsel	Samsung Galaxy A23	

Tabel 2 merupakan speksifikasi perangkat keras yang digunakan. Perangkat keras yang digunakan laptop Acer Nitro 5 dan ponsel Samsung Galaxy A23.

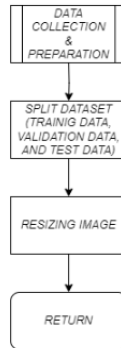
### 3.1.2. Perancangan Perangkat Lunak



**Gambar 22 Diagram Alir**

Gambar 22 merupakan diagram alir dari sistem klasifikasi seragam tentara Indonesia menggunakan metode *Convolutional Neural Network (CNN)*. Diagram alir dimulai dari pengumpulan dan pra proses data, augmentasi data, pemodelan dan *training* data, evaluasi sistem, dan penyebaran. Tahapan alur kerja dari perangkat lunak secara rinci sebagai berikut :

1. Data Collection & Preparation (Pengumpulan dan Pra Proses Data)



**Gambar 23 Tahapan Pengumpulan dan Pra Proses Data**

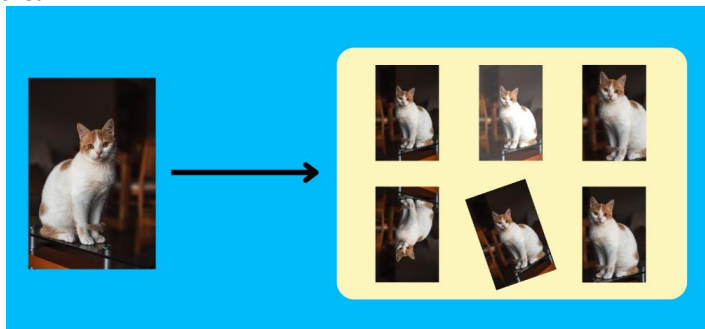
Gambar 23 merupakan tahapan yang dilakukan dalam pengumpulan dan pra proses data. Dalam tahapan ini diawali dengan pengumpulan *dataset* berupa citra atau gambar dari seragam loreng TNI AD dan seragam loreng Kostrad. Lalu, *dataset* yang sudah dikumpulkan dibagi menjadi data *training*, data *validation*, dan data *test*. Setiap *dataset* dilakukan penyeragaman ukuran citra sehingga dapat mempercepat proses *training*.

## 2. Data Augmentation (Augmentasi Data)



**Gambar 24 Tahapan Augmentasi Data**

Gambar 24 merupakan tahapan yang dilakukan dalam augmentasi data. Augmentasi merupakan salah satu cara yang dilakukan untuk mengatasi *overfitting*. Dalam tahap ini, data dimodifikasi menggunakan beberapa teknik, seperti *rotation*, *zoom*, *horizontal flip*, *width shift*, *height shift*, dan *shear*.



**Gambar 25 Contoh Augmentasi Data [16]**

Gambar 25 merupakan contoh augmentasi data. Pada sisi kiri terdapat gambar asli dan pada sisi kanan terdapat gambar hasil augmentasi.

3. *Modelling & Training Data* (Pemodelan dan Pelatihan Data)

Tahap pemodelan dan pelatihan data adalah tahap menentukan metode *deep learning* yang digunakan, menentukan algoritma *deep learning* yang digunakan, dan menentukan *hyperparameter* yang sesuai agar menghasilkan nilai akurasi yang baik. Metode yang digunakan adalah metode klasifikasi pada *deep learning* dengan algoritma *CNN* dan model arsitektur NASNetMobile.

**Tabel 3 Available Models on Keras Applications [39]**

Model	Size (MB)	Top-1 Accuracy	Top-5 Accuracy	Parameters	Depth	Time(ms) per inference step (CPU)	Time (ms) per inference step (GPU)
Xception	88	79.0%	94.5%	22.9M	81	109.4	8.1
VGG16	528	71.3%	90.1%	138.4M	16	69.5	4.2
VGG19	549	71.3%	90.0%	143.7M	19	84.8	4.4
ResNet50	98	74.9%	92.1%	25.6M	107	58.2	4.6
ResNet50V2	98	76.0%	93.0%	25.6M	103	45.6	4.4
ResNet101	171	76.4%	92.8%	44.7M	209	89.6	5.2
ResNet101V2	171	77.2%	93.8%	44.7M	205	72.7	5.4
ResNet152	232	76.6%	93.1%	60.4M	311	127.4	6.5
ResNet152V2	232	78.0%	94.2%	60.4M	307	107.5	6.6
InceptionV3	92	77.9%	93.7%	23.9M	189	42.2	6.9

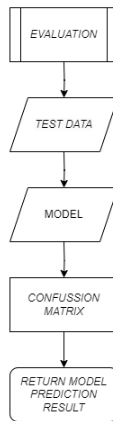
InceptionResNetV2	215	80.3%	95.3%	55.9M	449	130.2	10.0
MobileNet	16	70.4%	89.5%	4.3M	55	22.6	3.4
MobileNetV2	14	71.3%	90.1%	3.5M	105	25.9	3.8
DenseNet121	33	75.0%	92.3%	8.1M	242	77.1	5.4
DenseNet169	57	76.2%	93.2%	14.3M	338	96.4	6.3
DenseNet201	80	77.3%	93.6%	20.2M	402	127.2	6.7
NASNetMobile	23	74.4%	91.9%	5.3M	389	27.0	6.7
NASNetLarge	343	82.5%	96.0%	88.9M	533	344.5	20.0
EfficientNetB0	29	77.1%	93.3%	5.3M	132	46.0	4.9
EfficientNetB1	31	79.1%	94.4%	7.9M	186	60.2	5.6
EfficientNetB2	36	80.1%	94.9%	9.2M	186	80.8	6.5

EfficientNetB3	48	81.6%	95.7%	12.3M	210	140.0	8.8
EfficientNetB4	75	82.9%	96.4%	19.5M	258	308.3	15.1
EfficientNetB5	118	83.6%	96.7%	30.6M	312	579.2	25.3
EfficientNetB6	166	84.0%	96.8%	43.3M	360	958.1	40.4
EfficientNetB7	256	84.3%	97.0%	66.7M	438	1578.9	61.6
EfficientNetV2B0	29	78.7%	94.3%	7.2M	-	-	-
EfficientNetV2B1	34	79.8%	95.0%	8.2M	-	-	-
EfficientNetV2B2	42	80.5%	95.1%	10.2M	-	-	-
EfficientNetV2B3	59	82.0%	95.8%	14.5M	-	-	-
EfficientNetV2S	88	83.9%	96.7%	21.6M	-	-	-
EfficientNetV2M	220	85.3%	97.4%	54.4M	-	-	-

EfficientNetV2L	479	85.7%	97.5%	119.0M	-	-	-
ConvNeXtTiny	109.42	81.3%	-	28.6M	-	-	-
ConvNeXtSmall	192.29	82.3%	-	50.2M	-	-	-
ConvNeXtBase	338.58	85.3%	-	88.5M	-	-	-
ConvNeXtLarge	755.07	86.3%	-	197.7M	-	-	-
ConvNeXtXLarge	1310	86.7%	-	350.1M	-	-	-

Tabel 3 merupakan daftar model yang tersedia pada aplikasi Keras. Oleh karena sistem klasifikasi seragam tentara ini akan ditampilkan ke dalam sebuah aplikasi Android, maka model arsitektur *CNN* yang tepat adalah *NasNetMobile* dengan *size* tidak begitu berat untuk sebuah ponsel yaitu 23 MB.

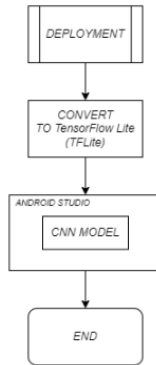
4. Evaluation (Evaluasi)



**Gambar 26 Tahapan Evaluasi**

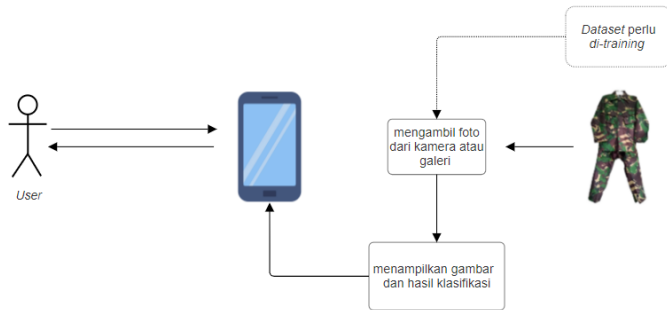
Gambar 26 merupakan tahap evaluasi dari hasil *deep learning* yang digunakan dalam proses pemodelan di tahap sebelumnya. Tahap evaluasi dilakukan untuk menyesuaikan model yang didapat agar sesuai dengan tujuan yang ingin dicapai dalam penelitian. *Dataset* yang digunakan untuk evaluasi adalah data test yang telah dibagi sebelumnya menggunakan metode *confusion matrix* sebagai acuan untuk menentukan baik atau tidaknya performa model klasifikasi yang telah dilatih.

## 5. *Deployment* / Penyebaran



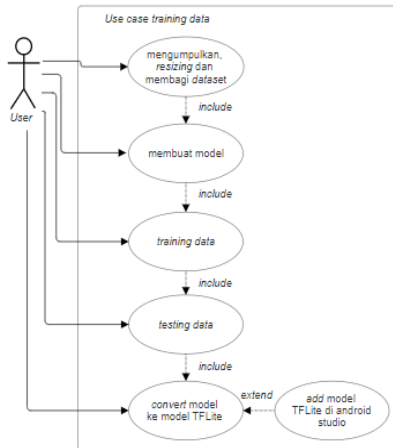
**Gambar 27 Tahapan Penyebaran**

Gambar 27 merupakan tahap penyebaran. *Deployment* atau penyebaran merupakan tahap dimana hasil pemodelan yang dihasilkan dari proses *machine learning* sebelumnya ditampilkan dalam sebuah aplikasi Android menggunakan *Integrated Development Environment (IDE)* Android Studio. Model yang diperoleh dari proses *training* diubah ke *TensorFlow Lite (TF Lite)* menjadi format model *machine learning* yang lebih kecil dan efisien agar dapat dijalankan di ponsel Android.



**Gambar 28 Diagram Alir Kerja Sistem Klasifikasi**

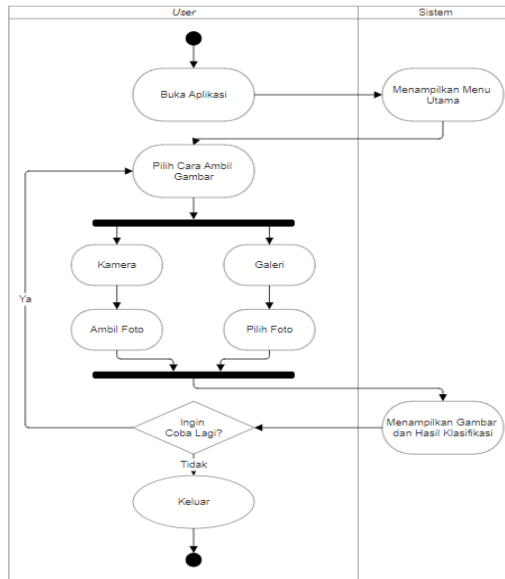
Gambar 28 adalah diagram alir cara kerja sistem klasifikasi. Sebelum *user* melakukan pengujian menggunakan ponsel Android, *user* harus melakukan *training* pada *dataset* terlebih dahulu. Setelah itu, *user* membuka aplikasi ponsel Android. *User* mengambil foto dari kamera atau galeri, dan gambar yang diambil serta hasil klasifikasi dari seragam tersebut ditampilkan.



**Gambar 29 Use Case untuk training data**

Gambar 29 merupakan diagram *use case* untuk *training data*. Diagram *use case* merupakan bentuk gambaran *user* dengan sistem. *User* disini merupakan orang yang melakukan melatih data. Mengumpulkan, *resizing*, dan membagi

*dataset* merupakan langkah pertama untuk melatih data. Setelah itu, *user* dapat membuat model dari menentukan model dan *hyperparameter* yang sesuai. *Training data* merupakan suatu kegiatan yang dilakukan *user* untuk melatih data dengan model yang sudah ditentukan. *Testing data* merupakan suatu kegiatan yang dilakukan *user* untuk menguji data dengan model yang telah dihasilkan dari proses *training data*. Setelah itu, *user* mengubah model tersebut menjadi *format* TFLite dan menambahkan model TFLite ke Android Studio.



**Gambar 30 Diagram Aktivitas Klasifikasi**

Gambar 30 merupakan diagram aktivitas untuk klasifikasi pada ponsel Android. Diawali dengan *user* membuka aplikasi, menu utama akan muncul, *user* memilih cara untuk ambil gambar dari kamera atau galeri, untuk kamera data diperoleh dari ambil foto dan galeri data diperoleh dari memilih gambar yang sudah tersimpan di galeri. Setelah itu, hasil klasifikasi dan gambar yang diambil ditampilkan. Jika ingin ambil foto lagi, *user* dapat memulai lagi dari memilih menu, jika tidak, *user* dapat keluar dari aplikasi.

### 3.2. Alat dan Bahan (Opsional)

Tabel 4 Estimasi Biaya

No.	Alat/bahan	Harga Satuan (Rp.)	Jumlah	Total (Rp.)	Keterangan
1	Laptop	-	1	-	Milik Pribadi
2	Seragam Loreng Malvinas PDL TNI AD	Rp225.000	1	RP225.000	Dana Pribadi
3	Seragam Loreng PDL TNI Kostrad	Rp160.000	1	RP160.000	Dana Pribadi
4	Tripod Ponsel	Rp68.000	1	RP68.000	Dana Pribadi
Total				Rp453.000	

Tabel 4 merupakan tabel estimasi biaya untuk tugas akhir. Dalam tugas akhir ini, semua kebutuhan alat/bahan yang digunakan menggunakan milik pribadi dan dana pribadi.

### 3.3. Pengujian

Setelah dilaksanakan perancangan pada sistem, tahap selanjutnya adalah pengujian. Pengujian dilakukan untuk melihat bagaimana performa dari model hasil *training* menggunakan *Convolutional Neural Network (CNN)* dalam mengklasifikasi seragam loreng TNI AD dan Kostrad.

Berikut pengujian yang akan dilakukan untuk penelitian ini:

1. Pengujian model menggunakan *confusion matrix*
2. Pengujian klasifikasi menggunakan data pelatihan
3. Pengujian klasifikasi menggunakan kamera
4. Pengujian klasifikasi terhadap perubahan jarak
5. Pengujian klasifikasi terhadap perubahan waktu (pagi dan malam)
6. Pengujian klasifikasi menggunakan jenis seragam loreng yang lain

## Bab 4. Hasil dan Pembahasan

### 4.1. Alur Kerja Sistem

Pada tahap awal yaitu pengumpulan *dataset* yang akan digunakan untuk proses *modelling* CNN dan *training* data. Pengambilan *dataset* dilakukan dengan cara mengumpulkan data citra yang berasal dari website resmi tentara Indonesia, video tentara Indonesia, dan sosial media resmi tentara Indonesia. Total keseluruhan *dataset* berjumlah 1002 yang dibagi menjadi dua kelas, yaitu Seragam Loreng Kostrad dan Seragam Loreng TNI AD. Setelah itu, ukuran semua citra akan diubah menjadi citra dengan ukuran 224x224.

**Tabel 5 Tabel Jumlah Citra**

No	Kelas Citra Seragam Loreng	Jumlah Citra
1	Seragam Loreng Kostrad	501
2	Seragam Loreng TNI AD	501

Tabel 5 merupakan tabel untuk jumlah citra pada setiap kelas. Kelas citra seragam loreng Kostrad berjumlah 501 citra dan kelas citra seragam loreng TNI AD berjumlah 501 citra.

Tabel 6 Contoh Citra *Dataset*

No	Citra	Kelas
1		Seragam Loreng Kostrad



Tabel 6 merupakan contoh citra tiap kelas yang digunakan sebagai *dataset* untuk proses *modelling* CNN dan *training* data. Pada proses ini, citra akan dibagi menjadi data *train*, data *test*, dan data *validation*.

```
In [3]: train_split=.8
test_split=.1
dummy_split=test_split/(1-train_split)
train_df, dummy_df=train_test_split(df, train_size=train_split, shuffle=True, random_state=123)
test_df, valid_df=train_test_split(dummy_df, train_size=dummy_split, shuffle=True, random_state=123)
print ('train_df length: ', len(train_df), ' test_df length: ', len(test_df), ' valid_df length: ', len(valid_df))

train_df length: 801 test_df length: 100 valid_df length: 101
```

**Gambar 31 Pembagian data *train*, data *test*, dan data *validation***

Gambar 31 merupakan jumlah pembagian data dalam proses *modelling* CNN dan *training* data. Pada proses ini, pembagian data ini menentukan seberapa banyak data yang akan digunakan untuk *training*, *testing*, dan *validation*. Penelitian kali ini, pembagian datanya adalah 80% data digunakan untuk data *train*, 10% data digunakan untuk data *test*, dan 10% data untuk data *validation*.

**Tabel 7 Tabel Hyperparameter**

Hyperparameter	
Split Data	80% data <i>train</i> , 10% data <i>test</i> , 10% data <i>validation</i>
Batch Size	64
Layers	6
Epoch	15
Optimizer	Adam
Learning Rate	0.001

Tabel 7 merupakan tabel *hyperparameter* yang akan digunakan dalam proses *training*. Dalam proses *training*, data dibagi menjadi 80% data *train*, 10%, data *test*, dan 10% data *validation*. *Batch size* atau sampel data yang disebar ke *neural network* sebanyak 64. Hal ini berarti, dari 1002 jumlah *dataset* maka algoritma CNN akan menggunakan 64 sampel data pertama dari 1002 data untuk disebar atau *training* oleh *neural network* sampai selesai. Setelah itu, mengambil kembali 64 sampel selanjutnya dari 1002 data tersebut, dan begitu seterusnya.

```
In [11]: model.summary()
Model: "sequential"
Layer (type)                Output Shape                Param #
-----
NASNet (Functional)         (None, 7, 7, 1056)         4269716
conv2d (Conv2D)             (None, 7, 7, 32)           304160
max_pooling2d (MaxPooling2D) (None, 3, 3, 32)           0
dropout (Dropout)           (None, 3, 3, 32)           0
flatten (Flatten)           (None, 288)                 0
dense (Dense)                (None, 2)                   578
-----
```

**Gambar 32 Model Summary**

Gambar 32 merupakan *model summary* pada proses *training*. Dapat dilihat bahwa proses *training* ini menggunakan arsitektur CNN dengan 6 layers yang terdiri dari *layer* dengan *base model* NASNetMobile, *layer* konvolusi, *layer max pooling*, *layer dropout*, *layer flatten*, dan *layer fully connected*. *Epoch* untuk proses *training* sebanyak 15. Hal ini berarti, proses *training* dilakukan sebanyak 15 kali iterasi atau 15 kali perulangan. *Optimizer* yang digunakan untuk *training* adalah optimizer Adam. Optimizer ini digunakan untuk memperbarui bobot *neural network* selama *training*. Optimizer Adam merupakan algoritma optimasi yang sering digunakan untuk proses *training* karena memiliki laju pembelajaran yang lebih cepat daripada optimasi yang lain. *Learning rate* yang digunakan sebesar

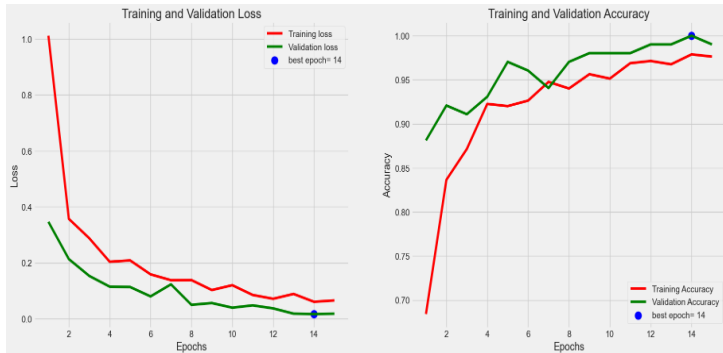
0.001, yang dimana semakin kecil *learning rate* maka *model* akan mempelajari *data training* semakin lambat, tetapi lebih stabil dan detail.

## 4.2. Hasil Proses *Training Model*

Setelah semua proses pengumpulan, *resizing*, dan penentuan *hyperparameter* selesai, tahapan selanjutnya adalah proses *training* data. Pada tahapan ini, dilakukan pembentukan model yang akan digunakan untuk proses pengujian. *Hyperparameter* yang telah ditentukan pada tabel 7 akan menentukan nilai akurasi selama *training*.

**Tabel 8** Tabel *Training*

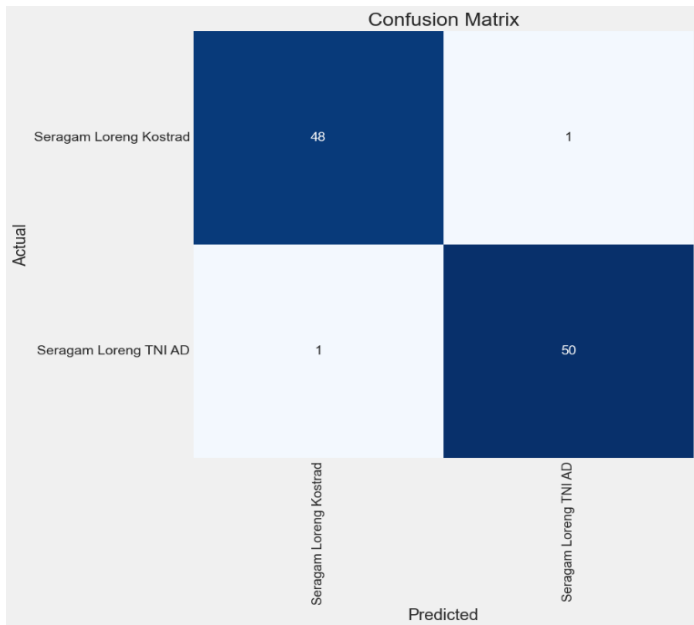
No	Epoch	Loss		Accuracy	
		loss	validation loss	accuracy	validation accuracy
1	Epoch 1	1.0114	0.3463	0.6841	0.8812
2	Epoch 2	0.3569	0.2132	0.8365	0.9208
3	Epoch 3	0.2876	0.1530	0.8714	0.9109
4	Epoch 4	0.2035	0.1146	0.9226	0.9307
5	Epoch 5	0.2084	0.1134	0.9201	0.9703
6	Epoch 6	0.1586	0.0798	0.9263	0.9604
7	Epoch 7	0.1377	0.1229	0.9476	0.9406
8	Epoch 8	0.1381	0.0498	0.9401	0.9703
9	Epoch 9	0.1025	0.0560	0.9563	0.9802
10	Epoch 10	0.1198	0.0396	0.9513	0.9802
11	Epoch 11	0.0849	0.0478	0.9688	0.9802
12	Epoch 12	0.0713	0.0370	0.9713	0.9901
13	Epoch 13	0.0886	0.0181	0.9675	0.9901
14	Epoch 14	0.0605	0.0167	0.9788	1.0000
15	Epoch 15	0.0654	0.0180	0.9763	0.9901



**Gambar 33** Grafik Hasil Proses *Training*

Tabel 8 merupakan tabel *training* dengan 15 *epoch* dan gambar 33 merupakan grafik hasil proses *training*. Dari tabel dan gambar grafik terlihat bahwa *epoch* atau iterasi menghasilkan nilai akurasi dan *loss* untuk data pelatihan dan pengujian. Nilai akurasi menunjukkan seberapa baik model berhasil melakukan prediksi dengan benar. Sebaliknya, nilai *loss* mengukur besarnya kesalahan yang dihasilkan oleh jaringan, dan tujuan dari pelatihan model adalah untuk meminimalisir nilai *loss*. Berdasarkan gambar grafik, garis berwarna merah menunjukkan pergerakan data *train* dan garis hijau menunjukkan pergerakan data *validation*. Pergerakan garis ini menunjukkan bahwa model mempelajari data *training* secara perlahan dan mendetail. Dapat dilihat bahwa pergerakan grafik *loss* semakin menurun, hanya saja pada saat *epoch* ke 7, grafik *loss* untuk data *validation* menaik dari 0.0789 ke 0.1229, lalu kembali menurun lagi. Untuk pergerakan grafik akurasi dapat dilihat bahwa pergerakan semakin menaik, namun untuk data *validation*, pergerakan grafik mulai stabil menaik setelah *epoch* ke 7.

### 4.3. Hasil Pengujian Model Menggunakan *Confusion Matrix*



**Gambar 34 *Confusion Matrix***

Gambar 34 merupakan gambar dari *confusion matrix*. *Confusion matrix* merupakan metode untuk mengukur bagaimana performa suatu model dalam melakukan klasifikasi [40]. *Confusion matrix* berbentuk tabel matriks dengan empat kombinasi nilai aktual dan prediksi yang berbeda. Terdapat 4 istilah sebagai representasi hasil proses klasifikasi. Berikut keempat istilah:

- *True Positive (TP)* = data positif yang diprediksi benar.
- *True Negative (TN)* = data negatif yang diprediksi benar.
- *False Positive (FP)* = data negatif yang diprediksi sebagai data positif.
- *False Negative (FN)* = data positif namun diprediksi sebagai data negatif.

Berdasarkan *confusion matrix* pada gambar 33, dapat dilihat *TP* sebesar 48 citra, *TN* sebesar 50 citra, *FP* sebesar 1 citra, dan *FN* sebesar 1 citra. Dengan tabel matriks ini berbagai *performance metrics* dapat dihitung untuk mengukur performa model yang telah dibuat. Berikut *performance metrics* yang digunakan akan digunakan untuk model yang dilatih:

- *Accuracy*  
*Accuracy* merupakan akurasi model secara keseluruhan. Persamaan 4.1 merupakan cara menghitung *accuracy*.

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \quad (4.1) [40]$$

$$Accuracy = \frac{48 + 50}{48 + 50 + 1 + 1}$$

$$Accuracy = \frac{98}{100} = 0.98$$

$$Accuracy (\%) = 0.98 \times 100\% = 98\%$$

- *Precision*  
*Precision* menggambarkan akurasi data yang diminta dengan hasil yang diberikan oleh model. Persamaan 4.2 merupakan cara menghitung *precision*.

$$Precision = \frac{TP}{TP + FP} \quad (4.2) [40]$$

$$Precision = \frac{48}{48 + 1}$$

$$Precision = \frac{48}{49} = 0.979$$

$$Precision (\%) = 0.979 \times 100\% = 97.9\%$$

- *Recall*  
*Recall* menggambarkan keberhasilan model dalam menemukan kembali sebuah informasi. Persamaan 4.3 merupakan cara menghitung *recall*.

$$Recall = \frac{TP}{TP + FN} \quad (4.3) [40]$$

$$Recall = \frac{48}{48 + 1}$$

$$Recall = \frac{48}{49} = 0.979$$

$$Recall (\%) = 0.979 \times 100\% = 97.9\%$$

- *F1-Score*

*F-1 Score* merupakan perbandingan rata – rata *precision* dan *recall* yang dibobotkan. Berikut rumus untuk menghitung *F-1 Score*:

$$F - 1 \text{ Score} = 2 * \frac{Precision * Recall}{Precision + Recall} \quad (4.4) [40]$$

$$F - 1 \text{ Score} = \frac{2TP}{2TP + FP + FN} \quad (4.5) [40]$$

$$F - 1 \text{ Score} = \frac{2(48)}{2(48) + 1 + 1}$$

$$F - 1 \text{ Score} = \frac{96}{96 + 1 + 1}$$

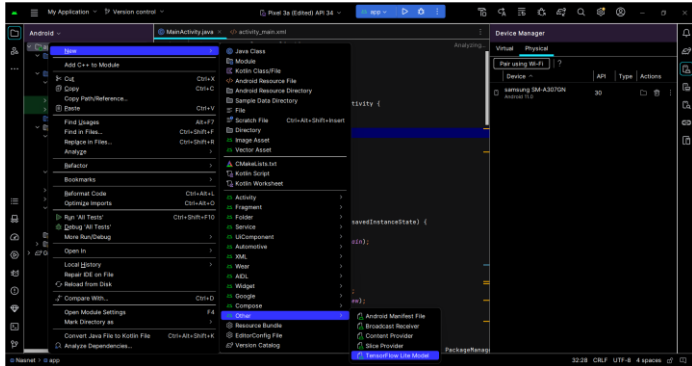
$$F - 1 \text{ Score} = \frac{96}{98} = 0.979$$

$$F - 1 \text{ Score} (\%) = 0.979 \times 100\% = 97.9\%$$

Dari pengujian yang telah dilakukan dengan sampel data *test* sebanyak 100 citra, maka diperoleh presentase *accuracy* sebesar 98%, *precision* sebesar 97.9%, *recall* sebesar 97.9%, dan *F-1 score* sebesar 97.9%. Model ini menunjukkan kinerja yang sangat baik dengan *precision*, *recall*, dan *F-1 score* yang tinggi untuk kedua kelas. *Accuracy* keseluruhan juga sangat tinggi. Hal ini menunjukkan bahwa model ini sangat efektif dalam mengklasifikasikan gambar Seragam Loreng Kostrad dan Seragam Loreng TNI AD.

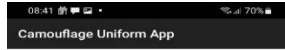
#### 4.4. Hasil Deployment

Pada tahap ini, model CNN yang telah dihasilkan dan dievaluasi menggunakan *confusion matrix*, disebar atau di-*deploy* ke sebuah aplikasi Android menggunakan *Integrated Development Environment (IDE)* Android Studio. Model yang diperoleh dari proses *training* diubah ke TensorFlow Lite (TFLite).



**Gambar 35 Cara Menambahkan Model TFLite**

Gambar 35 merupakan cara untuk menambahkan model TFLite di Android Studio. Untuk menambahkan model hasil *training* yang telah dikonversi ke format TFLite dengan cara klik kanan ke folder “app” yang berada di folder project, lalu ketuk *New, Other, TensorFlow Lite Model*. Tambahkan model tersebut. Ketika dibuka file model tersebut, akan terdapat dua versi code model, yaitu kode yang menggunakan bahasa pemrograman Java dan Kotlin. Setelah itu, dapat dipilih salah satu versi sesuai bahasa pemrograman yang digunakan untuk membangun aplikasi Android. Masukkan kode tersebut ke *MainActivity* (Java atau Kotlin Class) dan dapat menambahkan kode sesuai kebutuhan.



CLASSIFIED AS:

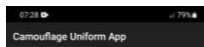
TAKE PICTURE

LAUNCH GALLERY



**Gambar 36** Tampilan Utama Aplikasi

Gambar 36 adalah tampilan utama aplikasi klasifikasi seragam loreng Kostrad dan TNI AD. Di dalam tampilan utama aplikasi, terdapat dua pilihan menu untuk mengambil gambar, yaitu *take picture* untuk mengambil gambar dari kamera langsung dan *launch gallery* untuk mengambil gambar dari foto yang tersimpan di *gallery*.



CLASSIFIED AS:  
Seragam Loreng TNI AD

TAKE PICTURE

LAUNCH GALLERY



**Gambar 37** Contoh Hasil Klasifikasi

Gambar 37 merupakan contoh hasil klasifikasi seragam loreng TNI AD. Klasifikasi ini menggunakan data pelatihan dan prediksinya benar bahwa gambar tersebut terdapat tentara yang menggunakan seragam loreng TNI AD.

#### 4.5. Hasil Pengujian Klasifikasi Menggunakan Data Pelatihan

**Tabel 9** Tabel Pengujian Klasifikasi Menggunakan Data Pelatihan

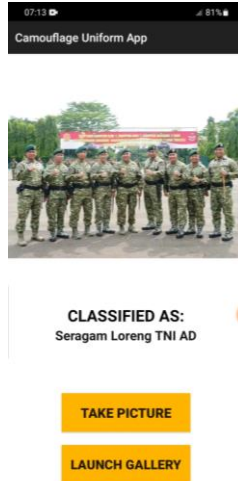
Nama Kelas	Jumlah Data	Sukses	Gagal	Akurasi
Seragam Loreng TNI AD	30	30	0	100%
Seragam Loreng Kostrad	30	29	1	96.6%
Total	60	59	1	98.3%

Tabel 9 merupakan tabel pengujian pertama, yaitu pengujian menggunakan data pelatihan. Dari semua data yang diuji, akan didapatkan data yang berhasil diklasifikasi dan data yang gagal diklasifikasi. Berdasarkan data-data ini, dapat diperoleh akurasi. Akurasi dapat dihitung dengan menggunakan persamaan 4.6.

$$Akurasi (\%) = \frac{jumlah\ data - data\ gagal}{jumlah\ data} \times 100\% \quad (4.6) [28]$$

$$Akurasi (\%) = \frac{59}{60} \times 100\% = 98.3\%$$

Untuk pengujian klasifikasi menggunakan data pelatihan, total jumlah data yang diuji dari dua kelas ada 60 data citra yang dimana data diambil secara acak dari *dataset*. Untuk kelas seragam loreng TNI AD, dari 30 data citra yang diuji, diperoleh semua data sukses diprediksi dan untuk kelas seragam loreng Kostrad, dari 30 data citra yang diuji, diperoleh 29 data citra yang sukses diprediksi dan 1 data citra yang gagal diprediksi. Jadi, akurasi keseluruhan data citra, diperoleh sebesar 98.3%.



**Gambar 38 Hasil Klasifikasi yang Gagal (Data Pelatihan)**

Gambar 38 merupakan hasil pengujian klasifikasi menggunakan data pelatihan yang gagal. Seharusnya hasil klasifikasi tersebut adalah seragam loreng Kostrad. Hal ini dikarenakan beberapa datataset seragam loreng Kostrad yang digunakan pada proses *training* kurang banyak variasi sehingga model kurang efektif dalam mengklasifikasi seragam loreng Kostrad.

## 4.6. Hasil Pengujian Klasifikasi Menggunakan Kamera

**Tabel 10 Tabel Pengujian Klasifikasi Menggunakan Kamera**

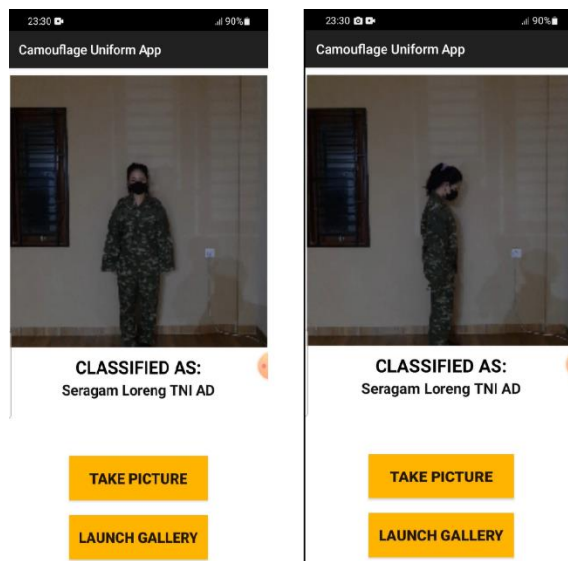
Nama Kelas	Jumlah Data	Sukses	Gagal	Akurasi
Seragam Loreng TNI AD	28	28	0	100%
Seragam Loreng Kostrad	28	26	2	92.8%
Total	56	54	2	96.4%

Tabel 10 merupakan tabel pengujian pertama, yaitu pengujian menggunakan data dari kamera. Dari semua data yang diuji, akan didapatkan data yang berhasil diklasifikasi dan data yang gagal diklasifikasi. Berdasarkan data-data ini, dapat

diperoleh akurasi. Akurasi dapat dihitung dengan menggunakan persamaan 4.6.

$$Akurasi (\%) = \frac{jumlah\ data - data\ gagal}{jumlah\ data} \times 100\% \quad (4.6) [28]$$
$$Akurasi (\%) = \frac{54}{56} \times 100\% = 96.4\%$$

Untuk pengujian klasifikasi menggunakan kamera, total jumlah data yang diuji dari dua kelas ada 56 data citra, yang dimana data merupakan data di luar *dataset*. Data yang digunakan merupakan data citra dari penulis yang menggunakan seragam loreng dari dua kelas tersebut. Untuk kelas seragam loreng TNI AD, dari 28 data citra yang diuji, diperoleh semua data sukses diprediksi dan untuk kelas seragam loreng Kostrad, dari 28 data citra yang diuji, diperoleh 26 data citra yang sukses diprediksi dan 2 data citra yang gagal diklasifikasi. Akurasi keseluruhan data citra, diperoleh sebesar 96.4%.



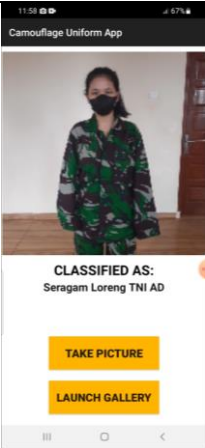
**Gambar 39 Hasil Klasifikasi yang Gagal (Kamera)**

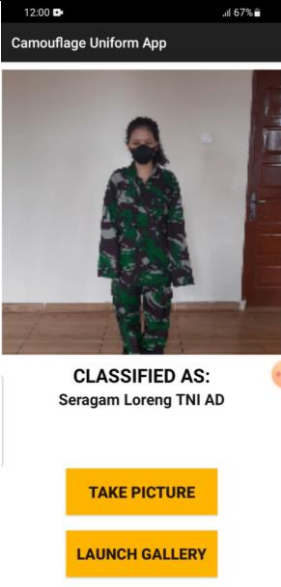
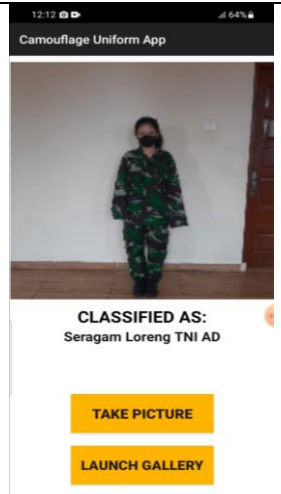
Gambar 39 merupakan hasil klasifikasi yang gagal menggunakan data yang diambil langsung dari kamera (data di luar data pelatihan). Pengambilan gambar dilakukan dengan jarak yang lebih dari 200 cm dan dilakukan pada malam hari.

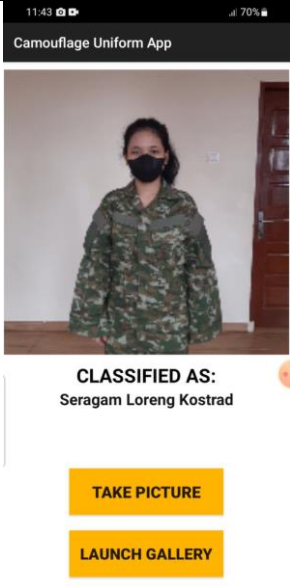
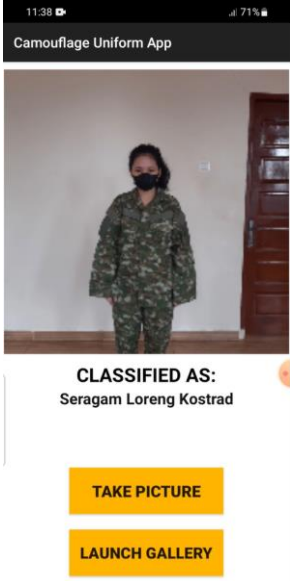
Hasil klasifikasi yang benar adalah seragam loreng Kostrad. Oleh karena pada saat malam hari pencahayaan kurang dan jarak kamera ke objek lebih dari 200 cm sehingga model salah dalam mengklasifikasi. Selain itu, kegagalan ini juga berasal dari faktor *dataset* seragam loreng Kostrad yang digunakan pada proses *training* kurang banyak variasinya sehingga menyebabkan model kurang efektif dalam mengklasifikasi seragam loreng Kostrad dan diperlukan pengambilan gambar lebih dari satu kali untuk mendapatkan hasil klasifikasi yang tepat.

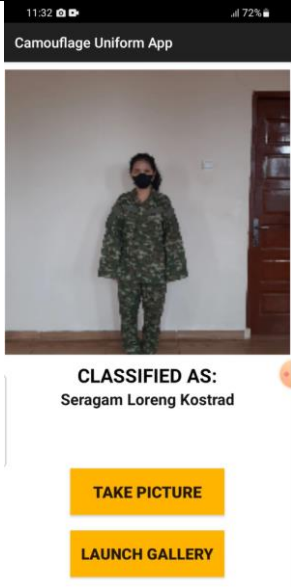
#### 4.7. Hasil Pengujian Klasifikasi Terhadap Perubahan Jarak

Tabel 11 Tabel Pengujian Klasifikasi Terhadap Perubahan Jarak

Nama Kelas	Jarak (cm)	Hasil (Gambar)	Keterangan
Seragam Loreng TNI AD	dekat (100)		Sukses

	<p>sedang (150)</p>		<p>Sukses</p>
	<p>jauh (200)</p>		<p>Sukses</p>

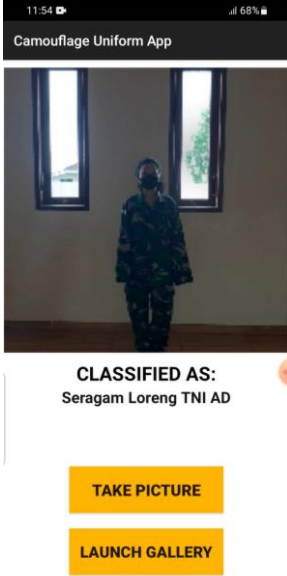
Seragam Loreng Kostrad	dekat (100)		Sukses
	sedang (150)		Sukses

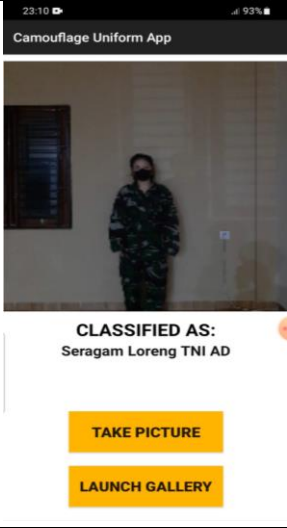
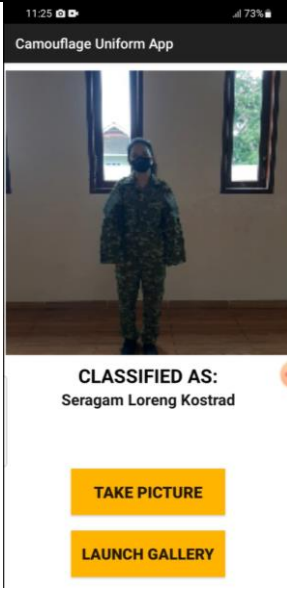
	<p>jauh (200)</p>		<p>Sukses</p>
--	-------------------	---	---------------

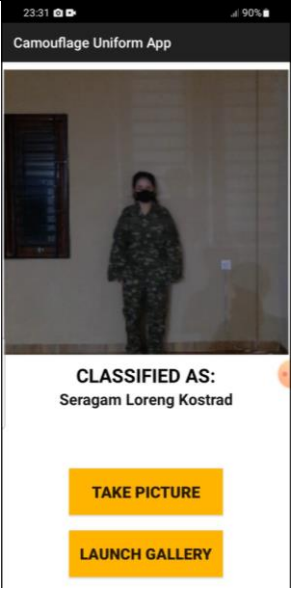
Tabel 11 merupakan tabel pengujian klasifikasi terhadap perubahan jarak. Pengujian dilakukan dengan berbagai jarak. Jarak dekat yaitu 100 cm dari kamera ponsel, jarak sedang 150 cm dari kamera ponsel, dan jarak jauh yaitu 200 cm dari kamera ponsel. Dari hasil klasifikasi, diperoleh bahwa model dapat mengklasifikasi seragam loreng TNI AD dan Kostrad dengan tepat dari berbagai jarak yang ditentukan. Hal ini menunjukkan bahwa performa model baik dalam mengklasifikasi walaupun adanya perubahan jarak dari kamera ke objek.

## 4.8. Hasil Pengujian Klasifikasi Terhadap Perubahan Waktu

Tabel 12 Tabel Pengujian Klasifikasi Terhadap Perubahan Waktu

Nama Kelas	Waktu	Hasil (Gambar)	Keterangan
Seragam Loreng TNI AD	Pagi	 <p>The screenshot shows the 'Camouflage Uniform App' interface. At the top, the title 'Camouflage Uniform App' is displayed. Below the title is a photo of a person wearing a TNI AD camouflage uniform. Underneath the photo, the text 'CLASSIFIED AS: Seragam Loreng TNI AD' is shown. At the bottom of the app interface, there are two yellow buttons: 'TAKE PICTURE' and 'LAUNCH GALLERY'.</p>	Sukses


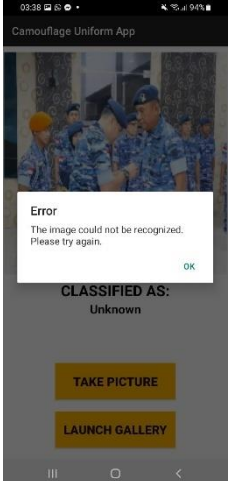
	Malam	 <p>23:10 93%</p> <p>Camouflage Uniform App</p> <p><b>CLASSIFIED AS:</b> Seragam Loreng TNI AD</p> <p>TAKE PICTURE</p> <p>LAUNCH GALLERY</p>	Sukses
Seragam Loreng Kostrad	Pagi	 <p>11:25 73%</p> <p>Camouflage Uniform App</p> <p><b>CLASSIFIED AS:</b> Seragam Loreng Kostrad</p> <p>TAKE PICTURE</p> <p>LAUNCH GALLERY</p>	Sukses

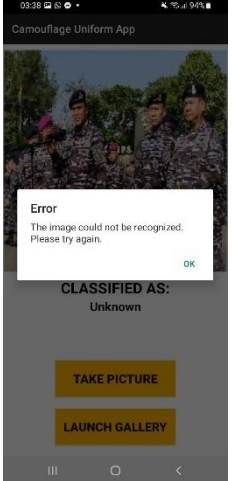
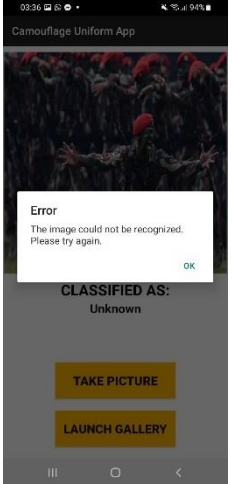
	Malam		Sukses
--	-------	---	--------

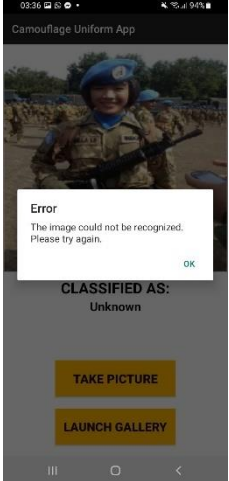
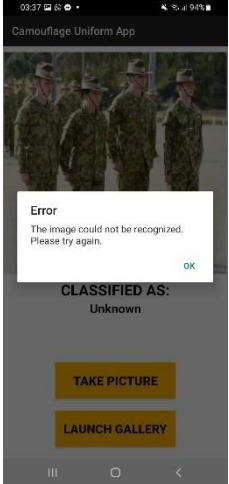
Tabel 12 merupakan tabel pengujian klasifikasi terhadap perubahan waktu. Pengujian dilakukan pagi dan malam. Hasil klasifikasi akan dibandingkan dengan data aslinya dan diberi keterangan apakah klasifikasi gagal atau sukses. Dari hasil klasifikasi, diperoleh bahwa model dapat mengklasifikasi seragam loreng TNI AD dan Kostrad dengan tepat di waktu pagi dan malam. Untuk pada saat malam hari, pengujian pada klasifikasi seragam loreng Kostrad harus dilakukan lebih dari satu kali agar model CNN dapat mengklasifikasi dengan tepat. Pada saat pengambilan gambar di malam hari ini juga jarak lebih dari 200 cm dari kamera ke objek. Oleh karena itu, pencahayaan pada malam hari dan jarak yang jauh (lebih dari 200 cm) mempengaruhi hasil pengujian seragam loreng Kostrad, sehingga pengambilan gambar perlu diulang lebih dari satu kali.

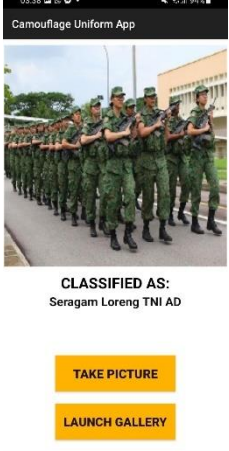

## 4.9. Hasil Pengujian Klasifikasi Menggunakan Jenis Seragam Loreng Lain





Tabel 13 Tabel Hasil Klasifikasi Menggunakan Jenis Seragam Loreng Lain

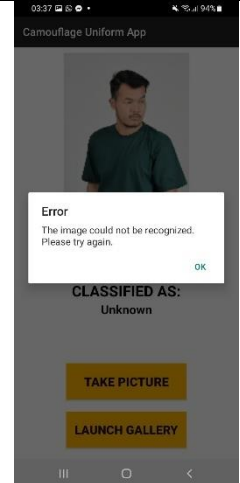
Nama Seragam Lainnya	Hasil Klasifikasi	Keterangan (Benar/Salah)
Seragam Loreng Brimob	 <p>03:37 94% Camouflage Uniform App</p> <p>CLASSIFIED AS: Seragam Loreng TNI AD</p> <p>TAKE PICTURE</p> <p>LAUNCH GALLERY</p>	✘
Seragam Loreng TNI AU	 <p>03:38 94% Camouflage Uniform App</p> <p>Error The image could not be recognized. Please try again.</p> <p>OK</p> <p>CLASSIFIED AS: Unknown</p> <p>TAKE PICTURE</p> <p>LAUNCH GALLERY</p>	✔

<p>Seragam Loreng TNI AL</p>		<p>✓</p>
<p>Seragam Loreng Kopasus</p>		<p>✓</p>

<p>Seragam Loreng Pasukan Kontingen Garuda</p>		<p>✓</p>
<p>Seragam Loreng Tentara Cadangan Australia (<i>Reverse Army</i>)</p>		<p>✓</p>

<p>Seragam Loreng Tentara Singapura</p>	 <p>09:38 94%</p> <p>Camouflage Uniform App</p> <p><b>CLASSIFIED AS:</b> Seragam Loreng TNI AD</p> <p>TAKE PICTURE</p> <p>LAUNCH GALLERY</p>	<p>x</p>
<p>Seragam Loreng Tentara Amerika Serikat</p>	 <p>09:38 94%</p> <p>Camouflage Uniform App</p> <p><b>CLASSIFIED AS:</b> Seragam Loreng Kostrad</p> <p>TAKE PICTURE</p> <p>LAUNCH GALLERY</p>	<p>x</p>

<p>Seragam Loreng Tentara Inggris (<i>British Army</i>)</p>	 <p>02:36 [signal] [wifi] [battery] 94%</p> <p>Camouflage Uniform App</p>  <p><b>CLASSIFIED AS:</b> Seragam Loreng Kostrad</p> <p>TAKE PICTURE</p> <p>LAUNCH GALLERY</p> <p>[android navigation bar]</p>	<p>x</p>
<p>Seragam Loreng Gurun Tentara Australia</p>	 <p>03:58 [signal] [wifi] [battery] 91%</p> <p>Camouflage Uniform App</p>  <p><b>CLASSIFIED AS:</b> Seragam Loreng Kostrad</p> <p>TAKE PICTURE</p> <p>LAUNCH GALLERY</p> <p>[android navigation bar]</p>	<p>x</p>

<p>Seragam Non Loreng (Kaos hijau polos)</p>		<p>✓</p>
--	---	----------

Tabel 13 merupakan tabel hasil pengujian menggunakan data seragam loreng yang lain. Pengujian kali ini menggunakan data yang di luar kelas, yaitu seragam loreng selain seragam loreng TNI AD dan seragam loreng Kostrad. Ketika diuji dengan seragam yang bukan termasuk dalam kelas, sistem seharusnya menunjukkan hasil klasifikasi tidak terdeteksi. Tanda centang (✓) menunjukkan bahwa model benar mengklasifikasi data yang di luar kelas seragam loreng TNI AD dan seragam loreng TNI Kostrad sebagai data yang tidak diketahui (*unknown*) dan aplikasi akan menampilkan pesan *error*. Tanda silang (✗) menunjukkan bahwa model salah mengklasifikasi data seragam yang di luar dua kelas seragam loreng TNI AD dan seragam loreng Kostrad dan data tersebut diklasifikasikan sebagai salah satu dari kelas yang ada,

Untuk mengklasifikasi data yang di luar kelas tanpa melatih dan menambah kelas baru, dapat menggunakan metode *threshold on softmax output*. Setelah model menghasilkan probabilitas *softmax (confidences)* untuk suatu prediksi, dapat menetapkan sebuah nilai *threshold*. *Confidences* ini merupakan nilai array yang dihasilkan oleh model yang mewakili probabilitas atau tingkat keyakinan bahwa input tertentu termasuk dalam masing-masing kelas. Jika, *confidences* yang diberikan kurang dari nilai *threshold*, maka model mengklasifikasi data tersebut sebagai data diluar kelas. Penggunaan *threshold* ini tidak begitu efektif jika *confidences* yang diberikan tinggi pada data yang di luar kelas karena model dapat mengklasifikasi data di luar kelas tersebut sebagai salah satu dari kelas.

Berdasarkan tabel, terdapat 5 data yang salah diklasifikasi dan 6 gambar yang benar diklasifikasi. Seharusnya 5 data tersebut tidak dapat diklasifikasi tetapi sistem mengklasifikasi data tersebut. Dari 5 data yang salah diklasifikasi dapat

dilihat ada terdapat gambar yang memiliki warna yang mirip dengan seragam loreng TNI AD dan Kostrad, seperti contoh seragam loreng brimob. Kalau dilihat lebih detail, warna corak yang dimiliki seragam loreng brimob agak sedikit berbeda dengan seragam loreng TNI AD. Hanya saja warna keseluruhan sama. Hal ini yang menyebabkan seragam loreng lain yang mirip warna dengan seragam loreng TNI AD dan Kostrad dapat diklasifikasi.

Tidak hanya itu, model CNN yang dihasilkan tidak selalu efektif mengklasifikasi seragam loreng yang berbeda juga. Terdapat satu seragam yang berbeda warna dan corak, tetapi model mengklasifikasi seragam tersebut sebagai seragam dari salah satu kelas, contoh seragam loreng gurun Australia. Jika diperhatikan lebih detail, warna dan corak yang dimiliki berbeda dengan seragam loreng Kostrad, tetapi model mengklasifikasi data tersebut sebagai seragam loreng kostrad. Hal ini merupakan dampak penggunaan nilai *threshold*. Untuk data yang di luar kelas yang memiliki nilai *confidences* yang tinggi akan diklasifikasi sebagai salah satu data dari kelas.

Untuk nilai akurasi dapat dihitung menggunakan rumus:

$$Akurasi (\%) = \frac{jumlah\ data - data\ gagal}{jumlah\ data} \times 100\% \quad (4.6) [28]$$
$$Akurasi (\%) = \frac{6}{11} \times 100\% = 54.54\%$$

Nilai akurasi yang dihasilkan untuk pengujian ini adalah 54.54%, yang dimana model CNN yang dihasilkan belum baik mengklasifikasi data yang di luar kelas. Dibutuhkan *dataset* yang lebih banyak lagi dan bervariasi untuk mempresentasikan seragam loreng TNI AD dan Kostrad secara spesifik sehingga model CNN dapat dengan tepat mengklasifikasi dan dapat membedakan mana seragam loreng jenis lain dengan seragam loreng TNI AD dan Kostrad meskipun seragam loreng jenis lain memiliki warna yang mirip.

## Bab 5. Kesimpulan dan Saran

### 5.1. Kesimpulan

Berdasarkan hasil pengujian klasifikasi seragam loreng TNI AD dan seragam loreng Kostrad yang telah dilakukan, dapat disimpulkan bahwa:

1. Pembuatan sistem klasifikasi seragam loreng TNI AD dan Kostrad menggunakan metode *Convolutional Neural Network* terdiri dari beberapa tahap, yaitu diawali dari pengumpulan dan praposes data, augmentasi data, menambahkan model NASNetMobile (model Keras), *convolutional*, *max pooling*, *dropout*, *flatten*, *dense (fully connected)* dengan melakukan training sebanyak 15 *epoch*. Setelah itu evaluasi peforma model untuk melihat bagaimana peforma model dalam mengklasifikasi.
2. *Deployment* model *Convolutional Neural Network* ke ponsel Android yaitu dengan cara membuat aplikasi menggunakan *Integrated Development Environment (IDE)* Android Studio.
3. Model arsitektur CNN yang dibangun dengan menambah model NASNetMobile mampu mengklasifikasi seragam loreng TNI AD dan Kostrad. Model ini juga dapat mengklasifikasi data selain dari *dataset*. Hasil akurasi dari data validasi menggunakan *confusion matrix* sebesar 98% yang menunjukkan bahwa model ini memiliki peforma yang baik dalam mengklasifikasi. Hasil pengujian klasifikasi menggunakan data pelatihan menghasilkan akurasi sebesar 98.3%, pengujian klasifikasi menggunakan data kamera menghasilkan akurasi sebesar 96.4%, pengujian klasifikasi terhadap perubahan jarak berhasil dilakukan, dan pengujian klasifikasi terhadap perubahan waktu juga sukses dilakukan. Kegagalan klasifikasi terjadi ketika menguji seragam loreng Kostrad. Hal ini disebabkan oleh pencahayaan pada malam hari dan jarak lebih dari 200 cm, yang mempengaruhi hasil pengujian klasifikasi. Selain itu, variasi yang kurang dalam dataset kelas seragam Kostrad juga mempengaruhi hasil pengujian klasifikasi, sehingga menyebabkan model kurang efektif dalam mengklasifikasi dan memerlukan lebih dari satu kali pengambilan gambar untuk mendapatkan hasil klasifikasi yang tepat.
4. Model CNN yang dihasilkan belum dapat membedakan seragam loreng TNI AD dan Kostrad dengan jenis seragam loreng yang lain. Nilai akurasi yang dihasilkan untuk pengujian ini adalah 54.54%. Hal ini disebabkan oleh dataset yang ada kurang bervariasi dan kurang mempresentasikan seragam loreng TNI AD dan Kostrad secara spesifik sehingga model belum dapat membedakan warna dan corak yang mirip dari jenis seragam loreng lain dengan seragam loreng TNI AD dan Kostrad.

Penggunaan nilai *threshold* tidak selalu efektif dalam mengklasifikasi data di luar kelas yang menyebabkan seragam loreng yang berbeda pun dapat diklasifikasi sebagai salah satu dari kelas yang ada.

## 5.2. Saran

Berdasarkan hasil penelitian yang telah dilakukan, adapun saran yang dapat diberikan oleh penulis untuk penelitian selanjutnya, yaitu:

1. Untuk penelitian selanjutnya dapat menerapkan arsitekur CNN dengan *hyperparameter* yang berbedabeda.
2. Untuk penelitian selanjutnya dapat menambahkan *dataset* yang lebih banyak pada proses *training* agar mendapatkan hasil akurasi yang lebih baik lagi.
3. Untuk penelitian selanjutnya dapat menambahkan kelas seragam loreng seperti menambahkan kelas seragam loreng dari tentara luar negeri.
4. Untuk penelitian selanjutnya diharapkan model klasifikasi dapat diakses dan dijalankan selain di ponsel Android.
5. Untuk penelitian selanjutnya diharapkan menggunakan metode lain untuk mengklasifikasi data yang di luar kelas.

## Daftar Pustaka

- [1] Y. Ernes, "Letkol TNI Gadungan Ditangkap di Depok, Tipu Eks Camat Rp 38 Juta," *detiknews*, 16 September 2023. [Online]. Available: <https://news.detik.com/berita/d-6934728/letkol-tni-gadungan-ditangkap-di-depok-tipu-eks-camat-rp-38-juta>. [Accessed 30 May 2024].
- [2] R. C. Gonzalez and R. E. Woods, *Digital Image Processing Third Edition*, Pearson Education, Inc, 2008.
- [3] T. Svoboda, J. Kybic and V. Hlavac, *Image Processing, Analysis, and Machine Vision: A MATLAB Companion*, Thomson Learning, 2008.
- [4] V. Tyagi, *Understanding Digital Understanding Digital*, Florida: CRC Press Taylor & Francis Group, 2018.
- [5] F. Zola, G. W. Nurcahyo and J. Santony, "Jaringan Syaraf Tiruan Menggunakan Algoritma Backpropagation Untuk Memprediksi Prestasi Siswa," *Jurnal Teknologi dan Open Source*, vol. 1, no. 1, pp. 58-72, 2018.
- [6] L. A. Wulandhari, "Artificial Neural Network Part 1," BINUS University, 27 February 2017. [Online]. Available: <https://socs.binus.ac.id/2017/02/27/artificial-neural-network-part-1/>. [Accessed 14 January 2024].
- [7] SaffronEdge, "Linkedin," SaffronEdge, 24 February 2023. [Online]. Available: <https://www.linkedin.com/pulse/feedforward-vs-backpropagation-ann-saffronedge1>. [Accessed 14 January 2024].
- [8] M. Yunus, "#6 Artificial Neural Network (ANN) — Part 1 (Pengenalan)," *Medium*, 8 April 2020. [Online]. Available: <https://yunusmuhammad007.medium.com/6-artificial-neural-network-ann-part-1-pengenalan-db487b8f8d85>. [Accessed 14 January 2024].
- [9] J. W. G. Putra, *Pengenalan Konsep Pembelajaran Mesin dan Deep Learning*, 2020.
- [10] R. H. Pramestya, "Deteksi dan Klasifikasi Kerusakan Jalan Aspal Menggunakan Metode Yolo Berbasis Citra Digital," *Thesis : Institut Teknolgi Sepuluh : Surabaya*, 2018.
- [11] R. Nima and F. Shila, "Crack classification in rotor-bearing system by means of wavelet transform and deep learning methods: an experimental investigation," *Journal Of Mechanical Engineering, Automation And Control Systems*, vol. 1, no. 2, pp. 102-113, 2020.

- [12] R. E. Putra, "Penerapan Algoritma Convolutional Neural Network Dan Long Short-Term Memory Pada Human Activity Recognition Berbasis Pengolahan Visual Pada Video," *Thesis, Universitas Siliwangi, Indonesia*, 2022.
- [13] Adminlp2m, "Convolutional Neural Networks (CNNs) – Definisi dan Penjelasan," LP2M Universitas Medan Area, 18 June 2023. [Online]. Available: <https://lp2m.uma.ac.id/2023/06/18/convolutional-neural-networks-cnns-definisi-dan-penjelasan/>. [Accessed 17 January 2024].
- [14] C. L. Nazalia, "Implementasi Algoritma Convolutional Neural Network Untuk Deteksi Hama Di Tanaman Sawi Hijau," *Thesis, Institut Teknologi PLN, Indonesia*, 2022.
- [15] Y. Zhou, C. Guo, X. Wang, Y. Chang and Y. Wu, "A Survey on Data Augmentation in Large Model Era," *Journal of Latex Class Files*, vol. 14, no. 8, pp. 1-33, 2015.
- [16] S. Pal, "An Intuitive Guide On Data Augmentation In Deep Learning – Techniques With Examples," GeekPython, 30 November 2022. [Online]. Available: <https://geekpython.in/data-augmentation-in-deep-learning>. [Accessed 2 February 2024].
- [17] M. Farhan, "Analisis Perbandingan Pengaruh Variasi Data Augmentasi Terhadap Kinerja MobileNetV2 Dalam Klasifikasi Penyakit Daun Teh," *Thesis, Universitas Islam Negeri Indonesia, Indonesia*, 2024.
- [18] Suyanto, K. N. Ramadhani and S. Mandala, *Buku Deep Learning Modernisasi Machine Learning Untuk Big Data*, Bandung: Informatika, 2019.
- [19] L. Alzubaidi, J. Zhang, A. J. Humaidi, A. Al-Dujaili, Y. Duan, O. Al-Shamma, J. Santamaría, M. A. Fadhel, M. Al-Amidie and L. Farhan, "Review of deep learning: concepts, CNN architectures, challenges, applications, future directions," *Journal of Big Data*, vol. 8, no. 53, pp. 1-74, 2021.
- [20] A. Fuadi and A. Suharso, "Perbandingan Arsitektur Mobilenet Dan Nasnetmobile Untuk Klasifikasi Penyakit Pada Citra Daun Kentang," *JIP (Jurnal Ilmiah Penelitian dan Pembelajaran Informatika)*, vol. 7, no. 3, pp. 701-710, 2022.
- [21] Undang Undang Republik Indonesia Nomor 34 Tahun 2004 tentang Tentara Nasional Indonesia.
- [22] Keputusan Menteri Pertahanan Nomor : KEP/568NI/2012 tentang Standar Militer Indonesia Nomor : SMI-STD-83-1 Pakaian Seragam Militer.

- [23] A. Romadoni, "TNI AD Punya Seragam Baru, Lihat Beda Lorengnya dengan TNI AU dan TNI AL," *kumparanNEWS*, 2 March 2022. [Online]. Available: <https://kumparan.com/kumparannews/tni-ad-punya-seragam-baru-lihat-beda-lorengnya-dengan-tni-au-dan-tni-al-1xbhTnBgRJT/full>. [Accessed 2023].
- [24] A. T. AD, "Blusukan Hingga ke Dusun, Kadispenad Cek Pelaksanaan TMMD di Bantul," *TNI Angkatan Darat*, 7 March 2024. [Online]. Available: <https://tniad.mil.id/blusukan-hingga-ke-dusun-kadispenad-cek-pelaksanaan-tmmd-di-bantul/>. [Accessed 15 March 2024].
- [25] A. Kostrad, "Pangdivif 2 Kostrad Pimpin Acara Penerimaan Jabatan Komandan Brigif 9 Kostrad," *KOMANDO CADANGAN STRATEGIS ANGKATAN DARAT*, 15 March 2024. [Online]. Available: [https://kostrad.mil.id/post\\_berita/pangdivif-2-kostrad-pimpin-acara-penerimaan-jabatan-komandan-brigif-9-kostrad/](https://kostrad.mil.id/post_berita/pangdivif-2-kostrad-pimpin-acara-penerimaan-jabatan-komandan-brigif-9-kostrad/). [Accessed 15 March 2024].
- [26] M. E. Irhebhude and E. A. Edirisinghe, "Military Personnel Recognition System using Texture, Colour and SURF features," in *SPIE - The International Society for Optical Engineering*, Baltimore, 2014.
- [27] M. E. Irhebhude and E. A. Edirisinghe, "Personnel Recognition in the Military using Multiple Features," *International Journal of Computer Vision and Signal Processing*, vol. 5, no. 1, pp. 23-30, 2015.
- [28] H. N. Ashshidiqi, Suprayogi and H. B. D. Kusumaningrum, "Identifikasi Pada Seragam Personel Militer Menggunakan Image Processing," vol. 4, no. 1, pp. 563-571, 2017.
- [29] H. Bethaningtyas, H. Naufal and G. W. Fajarianto, "Pengenalan Jenis Seragam Loreng TNI menggunakan Eccentricity dan Metric," *Jurnal Penelitian dan Pengembangan Telekomunikasi, Kendali, Komputer, Elektrik, dan Elektronika*, vol. 2, no. 2, pp. 1-8, 2017.
- [30] M. E. Morocho-Cayamcela and W. Lim, "Learning the patterns of soldier uniforms from the weights of a Semantic Segmentation Network," in *2018 International Workshop on Industrial IT Convergence (WIITC)*, Gumi, 2018.
- [31] M. E. Morocho-Cayamcela and W. Lim, "Research On Discriminating The Patterns of Soldier Uniforms Using Deep Learning," in *2019 Korean Institute of Communications and Information Sciences Conference (KICS)*, Pyeongchang, 2019.
- [32] R. Monitawati, Suwandi and H. B. Dyah, "Tracking Seragam Militer Berbasis Image Processing Secara Real Time," vol. 6, no. 1, pp. 1364-1369, 2019.

- [33] M. E. Morocho-Cayamcela and W. Lim, "Pattern Recognition of Soldier Uniforms With Dilated Convolutions and A Modified Encoder-Decoder Neural Network Architecture," *Applied Artificial Intelligence*, vol. 35, no. 6, pp. 476-487, 2021.
- [34] B. Dubetsky, K. Fernandez, G. Christopher, L. Singh, J. Hughes, J. Cole and M. Novitzky, "Military Uniform Identification For Search And Rescue (SAR) through," in *2022 IEEE International Symposium on Technologies for Homeland Security (HST)*, Boston, 2022.
- [35] I. Karatepe and V. Nabyev, "Military Camouflage Classification With Mask R-CNN," *Physical Sciences and Engineering*, vol. 65, no. 1, pp. 69-78, 2023.
- [36] H. i. Hatun, "Convolutional Neural Networks (CNNs) Understanding The Basic CNN Structure," Medium, 16 Juni 2018. [Online]. Available: <https://halil7hatun.medium.com/convolutional-neural-networks-cnns-95321b1f63ff>. [Accessed 19 Juli 2024].
- [37] A. H. Makarim, "Menggunakan Convolutional Neural Network (CNN) untuk Klasifikasi Gambar Produk," Medium, 15 Mei 2024. [Online]. Available: <https://medium.com/@21611129/menggunakan-convolutional-neural-network-cnn-untuk-klasifikasi-gambar-produk-c9f75c2446cb>. [Accessed 19 Juli 2024].
- [38] M. F. Herlambang, "Pengenalan Karakter Huruf Braille Dengan Metode Convolutional Neural Network," *Thesis, Institut Teknologi Nasional Bandung, Indonesia*, 2020.
- [39] Keras, "Keras Applications," [Online]. Available: <https://keras.io/api/applications/>. [Accessed 2023].
- [40] Prinzy and C. Lubis, "KLASIFIKASI BUAH SEGAR DAN BUSUK MENGGUNAKAN CONVOLUTIONAL NEURAL NETWORK BERBASIS ANDROID," *Jurnal Ilmu Komputer dan Sistem Informasi*.