

TUGAS AKHIR - SS234862

**ANALISIS KLASIFIKASI *FEW-SHOT LEARNING* UNTUK
IDENTIFIKASI PRODUK GAMBAR DI *E-COMMERCE*
MENGUNAKAN *SIAMESE CONVOLUTIONAL NEURAL
NETWORK* DAN *TRANSFER LEARNING***

HANIF CHOIRUDDIN

NRP 5003211063

Dosen Pembimbing

Dr. Dra. Kartika Fithriasari, M.Si.

NIP 19691212 199303 2 002

Tintrim Dwi Ary Widhianingsih, S.Si., M.Stat., Ph.D.

NIP 19950520 202406 2 003

Program Studi Sarjana Statistika

Departemen Statistika

Fakultas Sains dan Analitika Data

Institut Teknologi Sepuluh Nopember

Surabaya

2025



TUGAS AKHIR - SS234862

**ANALISIS KLASIFIKASI *FEW-SHOT LEARNING* UNTUK
IDENTIFIKASI PRODUK GAMBAR DI *E-COMMERCE*
MENGUNAKAN *SIAMESE CONVOLUTIONAL NEURAL
NETWORK* DAN *TRANSFER LEARNING***

HANIF CHOIRUDDIN

NRP 5003211063

Dosen Pembimbing

Dr. Dra. Kartika Fithriasari, M.Si.

NIP 19691212 199303 2 002

Tintrim Dwi Ary Widhianingsih, S.Si., M.Stat., Ph.D.

NIP 19950520 202406 2 003

Program Studi Sarjana Statistika

Departemen Statistika

Fakultas Sains dan Analitika Data

Institut Teknologi Sepuluh Nopember

Surabaya

2025



FINAL PROJECT - SS234862

FEW-SHOT LEARNING CLASSIFICATION ANALYSIS FOR IMAGE PRODUCT IDENTIFICATION IN E-COMMERCE USING SIAMESE CONVOLUTIONAL NEURAL NETWORK AND TRANSFER LEARNING

HANIF CHOIRUDDIN

NRP 5003211063

Advisors

Dr. Dra. Kartika Fithriasari, M.Si.

NIP 19691212 199303 2 002

Tintrim Dwi Ary Widhianingsih, S.Si., M.Stat., Ph.D.

NIP 19950520 202406 2 003

Bachelor Program of Statistics

Department of Statistics

Faculty of Science and Data Analytics

Institut Teknologi Sepuluh Nopember

Surabaya

2025

LEMBAR PENGESAHAN

ANALISIS KLASIFIKASI *FEW-SHOT LEARNING* UNTUK IDENTIFIKASI PRODUK GAMBAR DI *E-COMMERCE* MENGGUNAKAN *SIAMESE CONVOLUTIONAL* *NEURAL NETWORK* DAN *TRANSFER LEARNING*

TUGAS AKHIR

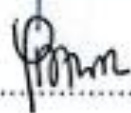
Diajukan untuk memenuhi salah satu syarat
memperoleh gelar Sarjana Statistika pada
Program Studi Sarjana Statistika
Departemen Statistika
Fakultas Sains dan Analitika Data
Institut Teknologi Sepuluh Nopember

Oleh: **HANIF CHOIRUDDIN**
NRP 5003211063

Tanggal Ujian: 29 Juli 2025
Periode Wisuda: September 2025

Disetujui oleh:
Pembimbing:

1. Dr. Dra. Kartika Fithriasari, M.Si.
NIP 19691212 199303 2 002
2. Tintrim Dwi Ary Widhianingsih, S.Si., M.Stat., Ph.D.
NIP 19950520 202406 2 003


.....

.....

Penguji:

3. Prof. Dr.rer.pol. Heri Kuswanto, S.Si., M.Si.
NIP 19820326 200312 1 004
4. Adatul Mukarromah, S.Si., M.Si.
NIP 19800418 200312 2 001


.....

.....

Kepala Departemen Statistika
Fakultas Sains dan Analitika Data



Dr.rer.pol. Dedy Dwi Prastyo, S.Si., M.Si. 
NIP 19831204 200812 1 002

(Halaman Sengaja dikosongkan)

APPROVAL SHEET

FEW-SHOT LEARNING CLASSIFICATION ANALYSIS FOR IMAGE PRODUCT IDENTIFICATION IN E-COMMERCE USING SIAMESE CONVOLUTIONAL NEURAL NETWORK AND TRANSFER LEARNING

FINAL PROJECT

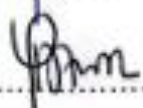
Submitted to fulfill one of the requirements
For obtaining a degree Bachelor of Statistics at
Bachelor Program of Statistics
Department of Statistics
Faculty of Science and Data Analytics
Institut Teknologi Sepuluh Nopember

By: **HANIF CHOIRUDDIN**
NRP 5003211063

Exam Date: 29 July 2025
Graduation Period: September 2025

Approved By:
Advisors:

1. Dr. Dra. Kartika Fithriasari, M.Si.
NIP 19691212 199303 2 002
2. Tintrim Dwi Ary Widhianingsih, S.Si., M.Stat., Ph.D.
NIP 19950520 202406 2 003


.....

.....

Examiners:

3. Prof. Dr.rer.pol. Heri Kuswanto, S.Si., M.Si.
NIP 19820326 200312 1 004
4. Adatul Mukarromah, S.Si., M.Si.
NIP 19800418 200312 2 001


.....

.....

Head of Statistics Departement
Faculty of Science and Data Analytics



Dr.rer.pol. Dedy Dwi Prastyo, S.Si., M.Si.
NIP 19831204 200812 1 002

(Halaman ini sengaja dikosongkan)

PERNYATAAN ORISINALITAS

Yang bertanda tangan di bawah ini:

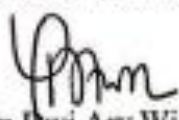
Nama mahasiswa / NRP : Hanif Choiruddin/ 5003211063
Departemen : Statistika FSAD - ITS
Dosen Pembimbing/NIP : Dr. Dra. Kartika Fithriasari, M.Si./
19691212 199303 2 002
Dosen Ko-Pembimbing/NIP : Tintrim Dwi Ary Widhianingsih, S.Si., M.Stat., Ph.D./
19950520 202406 2 003

Dengan ini menyatakan bahwa Tugas Akhir dengan judul "**Analisis Klasifikasi *Few-Shot Learning* Untuk Identifikasi Produk Gambar Di *E-Commerce* Menggunakan *Siamese Convolutional Neural Network* Dan *Transfer Learning***" adalah hasil karya sendiri, bersifat orisinal, dan ditulis dengan mengikuti kaidah penulisan ilmiah.

Bilamana di kemudian hari ditemukan ketidaksesuaian dengan pernyataan ini, maka saya bersedia menerima sanksi sesuai dengan ketentuan yang berlaku di Institut Teknologi Sepuluh Nopember.

Surabaya, 12 Juni 2025

Mengetahui,
Dosen Ko-Pembimbing


(Tintrim Dwi Ary Widhianingsih,
S.Si., M.Stat., Ph.D.)
NIP 19950520 202406 2 003

Mahasiswa


(Hanif/Choiruddin)
NRP 5003211063

Mengetahui,
Dosen Pembimbing


(Dr. Dra. Kartika Fithriasari, M.Si.)
NIP 19691212 199303 2 002

(Halaman ini sengaja dikosongkan)

STATEMENT OF ORIGINALITY

The undersigned below:

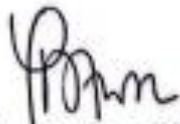
Name of student / NRP : Hanif Choiruddin/5003211063
Department : Statistics F-Scientics ITS
Advisor/NIP : Dr. Dra. Kartika Fithriasari, M.Si./
19691212 199303 2 002
Co-Advisor/NIP : Tintrim Dwi Ary Widhianingsih, S.Si., M.Stat., Ph.D./
19950520 202406 2 003

hereby declare that the Final Project with the title of "**Few-Shot Learning Classification Analysis For Image Product Identification In E-Commerce Using Siamese Network Convolutional Neural Network And Transfer Learning**" is the result of my own work, is original, and is written by following the rules of scientific writing.

If in the future there is a discrepancy with this statement, then I am willing to accept sanctions in accordance with the provisions that apply at Institut Teknologi Sepuluh Nopember.

Surabaya, 12 June 2025

Acknowledged,
Co-Advisor



(Tintrim Dwi Ary Widhianingsih,
S.Si., M.Stat., Ph.D.)
NIP 19950520 202406 2 003

Student



(Hanif Choiruddin)
NRP 5003211063

Acknowledged,
Advisor



(Dr. Dra. Kartika Fithriasari, M.Si.)
NIP 19691212 199303 2 002

(Halaman ini sengaja dikosongkan)

ABSTRAK

ANALISIS KLASIFIKASI *FEW-SHOT LEARNING* UNTUK IDENTIFIKASI PRODUK GAMBAR DI *E-COMMERCE* MENGGUNAKAN *SIAMESE CONVOLUTIONAL NEURAL NETWORK* DAN *TRANSFER LEARNING*

Nama Mahasiswa / NRP : Hanif Choiruddin / 5003211063
Departemen : Statistika FSAD ITS
Dosen Pembimbing : Dr. Dra. Kartika Fithriasari, M.Si.
Tintrim Dwi Ary Widhianingsih, S.Si., M.Stat., Ph.D.

Abstrak:

Pertumbuhan pesat e-commerce mendorong kebutuhan akan sistem pencarian produk berbasis gambar yang efisien, terutama untuk mengidentifikasi produk baru dengan jumlah data yang terbatas. Penelitian ini mengusulkan pendekatan *Few-Shot Learning* menggunakan arsitektur *Siamese Network* untuk tugas klasifikasi kemiripan gambar produk pada platform e-commerce. Penelitian ini mengkaji pendekatan *Few-Shot Learning* berbasis arsitektur *Siamese Network* dalam mengidentifikasi kemiripan citra produk e-commerce dengan keterbatasan data. Dua arsitektur backbone dibandingkan, yakni VGG-16 pretrained dan CNN, pada skenario *5-Way K-Shot* ($K = 1$ hingga 5). Data diperoleh dari kumpulan gambar produk Shopee di platform Kaggle, yang kemudian dipasangkan menjadi support-query untuk membentuk episode pelatihan. Proses preprocessing mencakup resize ke 256×256 piksel dan normalisasi sesuai standar VGG-16. Evaluasi dilakukan pada data validasi dan uji menggunakan metrik akurasi dan presisi antar kelas. Hasil eksperimen menunjukkan bahwa model dengan CNN memperoleh akurasi validasi tertinggi sebesar 54,17% pada konfigurasi *4-Shot*, serta akurasi uji tertinggi 49,90% pada konfigurasi yang sama. Sementara itu, VGG-16 menunjukkan performa yang lebih stabil dan efisien secara komputasi, terutama pada skenario *1-Shot* dan *5-Shot*. CNN unggul pada jumlah support menengah, namun menunjukkan fluktuasi yang lebih besar. Berdasarkan analisis menyeluruh terhadap akurasi, presisi antar kelas, serta efisiensi pelatihan, dapat disimpulkan bahwa pemilihan backbone sangat dipengaruhi oleh jumlah data dan kebutuhan sistem, CNN cocok untuk kondisi data terbatas, sedangkan VGG-16 lebih unggul untuk prediksi yang konsisten dan seimbang.

Kata kunci: *E-Commerce, CNN, Few-Shot Learning, Siamese Network, VGG-16, Identifikasi Produk.*

(Halaman ini sengaja dikosongkan)

ABSTRACT

FEW-SHOT LEARNING CLASSIFICATION ANALYSIS FOR IMAGE PRODUCT IDENTIFICATION IN E-COMMERCE USING SIAMESE CONVOLUTIONAL NEURAL NETWORK AND TRANSFER LEARNING

Name Of Student / NRP : Hanif Choiruddin / 5003211063
Department : Statistics F-Scientics ITS
Advisors : Dr. Dra. Kartika Fithriasari, M.Si
Tintrim Dwi Ary Widhianingsih, S.Si., M.Stat., Ph.D.

Abstract:

The rapid growth of e-commerce has driven the need for efficient image-based product search systems, particularly for identifying new products with limited data availability. This study proposes a Few-Shot Learning approach using a Siamese Network architecture to classify product image similarity on e-commerce platforms. Two backbone architectures are compared: a pretrained VGG-16 and a CNN, evaluated under a 5-Way K-Shot setting ($K = 1$ to 5). The dataset is derived from Shopee product images available on the Kaggle platform, which are paired into support-query episodes for training. Preprocessing includes resizing images to 256×256 pixels and normalizing them according to the VGG-16 standard. Evaluation is performed on both validation and test data using accuracy and per-class precision as performance metrics. Experimental results show that the CNN model achieved the highest validation accuracy of 54.17% and the highest test accuracy of 49.90%, both under the 4-Shot configuration. In contrast, VGG-16 demonstrated more stable and computationally efficient performance, especially in the 1-Shot and 5-Shot scenarios. While the CNN excelled in mid-Shot settings, it exhibited greater performance fluctuations. Overall, a comprehensive analysis of accuracy, class-wise precision, and training efficiency suggests that the choice of backbone is highly dependent on data volume and system requirements. The CNN is better suited for limited-data scenarios, whereas VGG-16 offers more consistent and balanced predictions.

Keywords: *E-Commerce, CNN, Few-Shot Learning, Siamese Network, VGG-16, Product Identification.*

(Halaman ini sengaja dikosongkan)

KATA PENGANTAR

Alhamdulillah, puji syukur kehadirat Allah SWT yang telah melimpahkan nikmat serta karunia-Nya yang tak terhingga, sehingga penulis dapat menyelesaikan laporan Tugas Akhir dengan judul “Analisis Klasifikasi *Few-Shot Learning* Untuk Identifikasi Produk Gambar Di *E-Commerce* Menggunakan *Siamese Convolutional Network* Dan *Transfer Learning*” dengan baik dan lancar.

Penulisan laporan Tugas Akhir ini tidak dapat diselesaikan secara lancar tanpa bantuan serta dukungan dari berbagai pihak. Oleh karena itu, penulis ingin menyampaikan ucapan terima kasih dan penghargaan kepada:

1. Ibunda tercinta Agus Setyaningsih, ayah Misturan Adipati, adik tersayang Husnaini Choirunnisak, saudara sepupu Odyha Hafidz Pangestu dan M. Haqqi Al Faris, serta segenap keluarga dan kerabat yang memberikan dukungan dan do’a kepada penulis.
2. Dr. Dra. Ismaini Zain, M.Si. selaku dosen wali yang memberikan saran dan arahan pada seluruh proses pembelajaran di Departemen Statistika Fakultas Sains dan Analitika Data ITS.
3. Ibu Dr. Kartika Fithriasari, M.Si. selaku dosen pembimbing atas arahan, saran, serta bimbingannya selama penyusunan Tugas Akhir.
4. Ibu Widhianingsih Tintrim Dwi Ary, S.Si., M.Stat., Ph.D. selaku dosen pembimbing kedua yang memberikan saran dan masukan terkait penulisan Tugas Akhir.
5. Bapak Dr.rer.pol. Dedy Dwi Prastyo, S.Si., M.Si. selaku Kepala Departemen Statistika Fakultas Sains dan Analitika Data ITS.
6. Ibu Dr. Wibawati, S.Si., M.Si. selaku Sekretaris Departemen Statistika Fakultas Sains dan Analitika Data.
7. Ibu Shofi Andari, S.Stat., M.Si., Ph.D. selaku Kepala Program Studi Sarjana Statistika Departemen Statistika Fakultas Sains dan Analitika Data ITS yang telah memberikan sarana dan prasarana selama perkuliahan.
8. Bapak Prof. Dr.rer.pol. Heri Kuswanto, S.Si., M.Si. dan Ibu Adatul Mukarromah, S.Si., M.Si., selaku dosen penguji yang memberikan masukan selama penyusunan Tugas Akhir.
9. Bapak dan Ibu dosen Departemen Statistika Fakultas Sains dan Analitika Data ITS yang telah memberikan ilmu yang bermanfaat bagi penulis.
10. Bapak/Ibu staf dan karyawan di Departemen Statistika Fakultas Sains dan Analitika Data ITS, terima kasih atas segala bantuan selama masa perkuliahan saya.
11. Supporter penulis di grup “Pondok Pesantren Statistika”, dan “Kimia Zelerett” yang tersayang yang senantiasa mendampingi perjalanan penulis,
12. Teman-teman Statistika ITS Angkatan 2021 “Bima Sakti”, serta semua pihak yang turut membantu dalam pelaksanaan Tugas Akhir.

Penulis menyadari bahwa dalam penulisan tugas akhir ini masih belum sempurna. Ketidaksempurnaan tersebut menjadi celah untuk pembelajaran dan pengembangan keilmuan kedepannya. Semoga Tugas Akhir ini bisa menjadi bahan referensi dan memberikan manfaat bagi pengembangan keilmuan statistika.

Surabaya 12 Juli 2025

Penulis

(Halaman ini sengaja dikosongkan)

DAFTAR ISI

LEMBAR PENGESAHAN.....	i
APPROVAL SHEET	iii
PERNYATAAN ORISINALITAS	v
STATEMENT OF ORIGINALITY.....	vii
ABSTRAK	ix
ABSTRACT	xi
KATA PENGANTAR.....	xiii
DAFTAR ISI	xv
DAFTAR GAMBAR	xvii
DAFTAR TABEL	xix
DAFTAR SIMBOL	xxi
DAFTAR LAMPIRAN	xxiii
BAB 1 PENDAHULUAN.....	1
1.1 Latar Belakang.....	1
1.2 Rumusan Masalah.....	3
1.3 Batasan Masalah	3
1.4 Tujuan.....	3
1.5 Manfaat	3
BAB 2 TINJAUAN PUSTAKA.....	5
2.1 Hasil Penelitian Terdahulu	5
2.2 <i>Artificial Neural Network</i>	6
2.2.1 <i>Single Layer Perceptron</i>	6
2.2.2 <i>Multi Layer Perceptron</i>	7
2.2.3 Fungsi Aktivasi	8
2.2.4 <i>Dropout</i>	9
2.2.5 <i>Batch Normalization</i>	10
2.3 Few-Shot Learning.....	11
2.4 Convolutional Neural Network.....	12
2.5 Transfer Learning Visual Geometry Group 16 (VGG-16).....	15
2.6 <i>Siamese Network</i>	16
2.6.1 <i>Framework Siamese Network</i> Secara Umum	17
2.6.2 Implementasi <i>Siamese Network</i> pada Model VGG-16	19
2.6.3 Implementasi <i>Few-Shot Learning</i> Pada Arsitektur <i>Siamese Network</i> ..	20
2.7 Evaluasi Model	21
2.8 <i>E-Commerce</i>	22
BAB 3 METODOLOGI	23
3.1 Sumber Data	23
3.2 Data Penelitian.....	23
3.3 Struktur Data.....	24
3.4 Langkah Analisis	26
3.5 Diagram Alir.....	27
BAB 4 HASIL DAN PEMBAHASAN.....	29
4.1 Spesifikasi <i>Cloud Service</i> Penunjang Eksperimen	29
4.2 Arsitektur <i>Siamese Network</i> dengan VGG-16 dan CNN.....	30

4.2.1	<i>Backbone VGG-16</i>	30
4.2.2	<i>Backbone CNN</i>	31
4.2.3	Konfigurasi <i>Siamese Network</i>	32
4.3	Pengolahan Data	34
4.3.1	Pra-Preprocessing Data	34
4.3.2	Preprocessing Data.....	35
4.4	Analisis Perbandingan Performa <i>Backbone VGG-16</i> dan CNN	36
4.4.1	<i>Few-Shot Learning Siamese Network Backbone VGG-16</i>	37
4.4.2	<i>Few-Shot Learning Siamese Network Backbone CNN</i>	38
4.4.3	Model Terbaik.....	40
BAB 5	PENUTUP	45
5.1	KESIMPULAN.....	45
5.2	SARAN.....	45
	DAFTAR PUSTAKA	47
	LAMPIRAN	53
	BIODATA PENULIS	61

DAFTAR GAMBAR

Gambar 2.1 Ilustrasi Arsitektur Perceptron Tunggal.....	6
Gambar 2.2 Ilustrasi Arsitektur Perceptron Multi-Class	7
Gambar 2.3 Ilustrasi Multilayer Perceptron 2 Layer	8
Gambar 2.4 Ilutstrasi Dropout	10
Gambar 2.5 Ilustrasi Paradigma Few-Shot Learning N-Way-K-Shot.....	11
Gambar 2.6 Arsitektur Convolutional Neural Network.....	12
Gambar 2.7 Ilustrasi Proses Convolutional Layer.....	13
Gambar 2.8 Max Pooling Layers dengan Ukuran 2x2 dan Stide 2	14
Gambar 2.9 Arsitektur VGG-16	15
Gambar 2.10 Ilustrasi Arsitektur Siamese Network	17
Gambar 2.11 Ilustrasi Contrastive Loss.....	18
Gambar 2. 12 Ilutstrasi Arsitektur Siamese Network Menggunakan Backbone VGG-16	19
Gambar 2.13 3-Way-3-Shot Pada Arsitektur Siamese Network Pada Training Task.....	20
Gambar 3.1 Diagram Alir Penelitian	27
Gambar 4.1 Arsitektur VGG-16	30
Gambar 4.2 Arsitektur CNN.....	31
Gambar 4.3 Konfigurasi Siamese Network terhadap Arsitektur	32
Gambar 4.4 Alur Pra-Preprocessing Data (5 Way 5 Shot)	34
Gambar 4.5 Proses pembuatan pasangan untuk Arsitektur Siamese Network.....	35
Gambar 4.6 Process Input Menggunakan VGG-16.....	36
Gambar 4.7 Akurasi Validasi pada Model Siamese VGG-16 (5-Way K-Shot).....	37
Gambar 4.8 Contrastive LossValidasi pada Model Siamese VGG-16 (5-Way K-Shot).....	38
Gambar 4.9 Akurasi Validasi pada Model Siamese CNN (5-Way K-Shot).....	39
Gambar 4. 10 Contrastive Loss Validasi pada Model Siamese CNN (5-Way K-Shot)	39

(Halaman ini sengaja dikosongkan)

DAFTAR TABEL

Tabel 2.1 Daftar Penelitian yang Memiliki Penelitian Menggunakan Dataset E-commerce Untuk Penerapan Pada Sistem Identifikasi.....	5
Tabel 2.2 Daftar Penelitian yang Menggunakan Few-Shot Learning Berbasis Metode Siamese Network Dalam Analisisnya.....	5
Tabel 2.3 Arsitektur model Siamese Network Menggunakan VGG-16.....	19
Tabel 2.4 Confusion Matrix.....	21
Tabel 3.1 Data Penelitian.....	23
Tabel 3.2 Label Data yang digunakan	24
Tabel 3.3 Struktur Data Gambar.....	24
Tabel 3.4 Struktur Data Convolutional.....	25
Tabel 4.1 Spesifikasi Cloud Service.....	29
Tabel 4.2 Arsitektur Siamese Network VGG-16.....	33
Tabel 4.3 Arsitektur Siamese Network CNN	33
Tabel 4.4 Jumlah pairs yang terbentuk pada kombinasi setiap task.	35
Tabel 4.5 Data Training dan Validasi.....	36
Tabel 4.6 Hasil Akurasi Model VGG-16 Sebagai Backbone pada 5-Way K-Shot.....	38
Tabel 4.7 Hasil Akurasi Model CNN Sebagai Backbone pada 5-Way K-Shot	40
Tabel 4.8 Data Testing.....	40
Tabel 4.9 Hasil Perbandingan Model Siamese VGG-16 dan Siamese CNN pada K-Shot	41
Tabel 4.10 Hasil Perbandingan Presisi Pada Model Siamese VGG-16 dan Siamese CNN pada K-Shot.....	41

(Halaman ini sengaja dikosongkan)

DAFTAR SIMBOL

Simbol	Keterangan
x_i	Nilai <i>input</i> ke- i , biasanya berupa <i>pixel</i> atau fitur dari data gambar
w_i	Bobot/parameter yang terkait dengan <i>input</i> x_i
w_0	Bias atau <i>intercept</i> dalam <i>perceptron</i>
$\vec{w}, \vec{x}$	Vektor parameter dan <i>input</i> yang sudah diaugmentasi (dalam bentuk vektor kolom)
Q	Jumlah total <i>input</i> (dimensi fitur)
al	Hasil dari aktivasi linear (dot product antara bobot dan <i>input</i>)
$g(\cdot)$	Fungsi aktivasi pada <i>hidden layer</i> (ReLU, Softmax)
$g'(\cdot)$	Fungsi aktivasi pada <i>Output layer</i>
z_p	<i>Output</i> dari <i>perceptron</i> ke- p di <i>hidden layer</i>
p	Jumlah <i>perceptron</i> di <i>layer</i> ke- p
y_c	<i>Output</i> dari <i>perceptron</i> kelas ke- c , hasil klasifikasi.
C	Jumlah kelas dalam klasifikasi
$\mathcal{A}_{ReLU}(\cdot)$	Fungsi aktivasi ReLU
$\mathcal{A}_c(y_i)$	Fungsi aktivasi Softmax pada <i>Output</i> kelas ke- c
$s()$	Fungsi diskriminasi untuk probabilitas yang dihasilkan oleh fungsi aktivasi softmax
$\mathcal{A}_{Sigmoid}(y)$	Fungsi aktivasi Sigmoid
y_c	<i>Output</i> dari <i>perceptron</i> kelas ke- c , hasil klasifikasi
μ_h	Nilai rata-rata channel ke- h dalam proses <i>batch normalization</i>
F	Tensor berdimensi <i>Height</i>
G	Tensor berdimensi <i>Width</i>
H	Jumlah channel
f, g, h	Ukuran dimensi filter (panjang, lebar, dan kedalaman)
σ_c^2	Varians channel ke- c dalam <i>batch normalization</i>
γ, β	Parameter yang dapat dilatih dalam proses <i>batch normalization</i>
$\hat{x}_{ifgh}$	Nilai <i>input</i> yang sudah dinormalisasi pada koordinat (I, f, g, h)
y_{ifgh}	<i>Output</i> akhir <i>batch normalization</i> setelah dikalikan dan ditambah γ, β
d	Kedalaman (<i>depth</i>) filter/kernel dalam <i>convolutional layer</i>
s	<i>Stride</i> atau langkah pergeseran filter
K	Kernel Padding
p	Padding (jumlah piksel yang ditambahkan di tepi <i>input</i>)
$B_{(i',j',u')}$	Nilai <i>Output</i> dari proses konvolusi pada lokasi (i', j', u')
i', j', u'	Indeks posisi <i>Output</i> feature map (hasil konvolusi), masing-masing menunjukkan:
L	<i>Layer</i> didalam Convolutional Neural Network
$Z_{i,j,u}^{(L)}$	<i>Output layer</i> terakhir CNN sebelum flattening
$F_{flatten}$	Vektor satu dimensi hasil flattening dari matriks fitur
D_{12}	Nilai jarak <i>Euclidean</i> antara dua vektor <i>embedding</i> pada <i>Siamese Network</i>
z_1, z_2	Komponen dari vektor <i>Output</i> (embedding) dari dua <i>subnetwork Siamese</i>
m	Margin pada fungsi <i>contrastive loss</i>
Y	Label biner (0 jika tidak mirip, 1 jika mirip)

Simbol	Keterangan
p	Nilai probabilitas <i>Output</i> akhir dari <i>Siamese Network</i>
β	Parameter bobot pada <i>layer</i> akhir untuk perhitungan probabilitas kesamaan pada <i>Siamese Network</i>
$S_{i,j}$	Elemen pada <i>Support Set</i> (data latih <i>Few-Shot Learning</i>)
$V_{i,j}$	Elemen pada <i>Query Set</i> (data uji <i>Few-Shot Learning</i>)
N	Jumlah kelas (<i>N-Way</i>)
K	Jumlah gambar per kelas (<i>K-Shot</i>)
TP,TN,FP,FN	<i>True Positive, True Negative, False Positive, False Negative</i>

DAFTAR LAMPIRAN

Lampiran 1	Arsitektur VGG-16	53
Lampiran 2	Ilustrasi Setiap Task dari 1-Shot sampai 5-Shot	54
Lampiran 3	Ilustrasi 2 Shot	54
Lampiran 4	Lanjutan Ilustrasi 2 Shot	55
Lampiran 5	Akurasi Validasi pada Model Siamese VGG-16 (5-Way K-Shot) Replikasi Ke-1	55
Lampiran 6	Contrastive Loss Validasi pada Model Siamese VGG-16 (5-Way K-Shot) Replikasi Ke-1	55
Lampiran 7	Akurasi Validasi pada Model Siamese CNN (5-Way K-Shot) Replikasi Ke-1.	56
Lampiran 8	Contrastive Loss Validasi pada Model Siamese CNN (5-Way K-Shot) Replikasi Ke-1	56
Lampiran 9	Akurasi Validasi pada Model Siamese VGG-16 (5-Way K-Shot) Replikasi Ke-10	56
Lampiran 10	Contrastive Loss Validasi pada Model Siamese VGG-16 (5-Way K-Shot) Replikasi Ke-10	57
Lampiran 11	Akurasi Validasi pada Model Siamese CNN (5-Way K-Shot) Replikasi Ke-10	57
Lampiran 12	Contrastive Loss Validasi pada Model Siamese CNN (5-Way K-Shot) Replikasi Ke-10	57
Lampiran 13	Akurasi Validasi pada Model VGG-16.....	58
Lampiran 14	Waktu Training Validasi pada Model VGG-16.....	58
Lampiran 15	Akurasi Validasi pada Model CNN	58
Lampiran 16	Waktu Validasi pada Model CNN	59
Lampiran 17	Code Analisis VGG-16 dan CNN.....	59
Lampiran 18	Surat Pernyataan Data.....	60

(Halaman ini sengaja dikosongkan)

BAB 1 PENDAHULUAN

1.1 Latar Belakang

Perkembangan *e-commerce* di Indonesia menunjukkan tren yang sangat positif dan menjanjikan. Menurut laporan e-Conomy SEA 2023, nilai ekonomi sektor *e-commerce* Indonesia mencapai US\$59 miliar pada tahun 2022, meningkat 22% dibandingkan tahun sebelumnya. Hal tersebut menjadikan perkembangan industri *e-commerce* yang pesat telah mendorong kebutuhan akan sistem identifikasi produk yang lebih efisien dan intuitif. Saat ini, metode pencarian berbasis kata kunci masih mendominasi, namun sering kali kurang efektif dalam menghubungkan konsumen dengan produk yang mereka inginkan. Dalam kondisi ini, pendekatan visual berbasis gambar menjadi solusi yang menjanjikan, bukan dalam bentuk *pencarian*, melainkan dalam bentuk pengukuran kesamaan antara gambar produk. Dengan pendekatan ini, sistem dapat membandingkan dua gambar dan mengukur tingkat kemiripan visual, untuk mengidentifikasi apakah keduanya mewakili produk dari kategori yang sama atau tidak (Togashi et al. 2017). Sistem ini memungkinkan pengguna untuk mengidentifikasi produk menggunakan gambar sebagai input, yang kemudian dibandingkan dengan produk yang ada di platform *e-commerce*. Penerapan teknologi ini dapat meningkatkan akurasi pencarian dan memberikan pengalaman berbelanja yang lebih memuaskan. Kepercayaan pelanggan terhadap produk yang dibeli secara online masih menjadi masalah utama dalam industri *e-commerce*. Hal ini disebabkan oleh beberapa faktor, termasuk ketidakmampuan untuk melihat produk secara langsung sebelum membelinya. Dalam beberapa kasus, gambar produk yang ditampilkan di situs web *e-commerce* tidak akurat atau bahkan palsu (Agustina 2017). Oleh karena itu, dibutuhkan teknologi yang mampu mengidentifikasi kemiripan citra secara objektif, bukan hanya mencocokkan teks deskriptif, sebagai bagian dari sistem verifikasi produk berbasis visual.

Solusi untuk masalah ini adalah dengan menggunakan computer vision di sistem identifikasi. *Computer vision* (CV) adalah teknologi yang memungkinkan komputer untuk memahami dan menganalisis gambar dan video (Guntara 2023). Dalam *e-commerce*, CV dapat digunakan untuk memperbaiki pengalaman pembelian online dengan memberikan gambaran yang akurat tentang produk yang dibeli. Untuk menguji efektivitas pemanfaatan CV pada *e-commerce*, khususnya pada pengenalan citra produk dan klasifikasi citra produk (*fashion*). Menggunakan teknologi CV dan *machine learning* untuk mengklasifikasikan produk secara otomatis pada situs web *e-commerce* (Villalba-Diez et al. 2019). *Machine learning* untuk mengenali produk berdasarkan ciri-ciri visual yang ada, lalu pengenalan citra produk dan klasifikasi citra produk dapat meningkatkan akurasi gambar produk pada situs web *e-commerce*, serta meningkatkan pengalaman belanja pelanggan (Yu et al. 2024).

Perkembangan signifikan dalam implementasi sistem identifikasi berbasis gambar pada platform *e-commerce*. Penelitian yang dilakukan oleh (Guntara 2023) mengungkapkan bahwa implementasi CV pada *e-commerce* memberikan dampak positif dalam meningkatkan akurasi gambar produk, efisiensi proses klasifikasi, serta pengalaman pengguna secara keseluruhan. Kemudian, studi yang dilakukan (Panggabean et al. 2025) menghasilkan temuan bahwa sistem identifikasi berbasis gambar yang diintegrasikan dengan AI mampu meningkatkan efektivitas pencarian produk secara signifikan, yang berdampak pada peningkatan engagement pengguna. Sementara itu, (Devi 2016) telah berhasil mengembang-

kan sistem identifikasi jenis produk *smartphone*, yang berbasis gambar khusus untuk produk *smartphone* dengan menggunakan pendekatan fitur morfologi, sebuah aplikasi mesin pencari untuk mengetahui spesifikasi dari sebuah *smartphone*. Namun, Aplikasi ini merupakan dapat mendeteksi tipe *smartphone* berdasarkan gambar yang diambil dari kamera *smartphone* pengguna.

Deep learning merupakan model yang di mana banyak lapisan tahap pemrosesan informasi dalam arsitektur hirarkis dieksploitasi untuk pembelajaran fitur tanpa pengawasan dan untuk klasifikasi pola. Teknik ini berada di persimpangan antara area penelitian *neural network*, pemodelan grafis, optimasi, pengenalan pola, dan pemrosesan sinyal (Mishra et al. 2021). Dalam penelitian ini digunakan model *deep learning* yang dapat mengenali setiap warna, bentuk, dan pola di setiap gambar, yang sering digunakan oleh kebanyakan orang adalah *convolutional neural network* (CNN) mempunyai kemampuan untuk mengenali fitur citra digital (*feature learning*) melalui konvolusi filter selama pelatihan (Azmi et al. 2023). Adapun metode yang sesuai untuk digunakan dalam penelitian ini adalah *Siamese Network* (Ma'Rufah et al. 2023). Menggunakan CNN untuk mengklasifikasikan produk berdasarkan citra yang dimilikinya (Smetanin and Komarov 2019). Dengan penggunaan CNN dapat meningkatkan akurasi klasifikasi produk secara signifikan, dan lebih efektif dibandingkan dengan metode *support vector machine*.

Untuk mengatasi tantangan dalam identifikasi gambar dan besarnya basis data yang dimiliki, *few-Shot learning* dan arsitektur *Siamese Network* menjadi pilihan yang tepat. *Few-Shot Learning* memungkinkan model mengenali produk baru dengan hanya sedikit contoh pelatihan, menjadikannya efektif dalam situasi di mana data pelatihan terbatas (Ma'Rufah et al. 2023). *Siamese Network*, dengan kemampuannya membandingkan dua gambar untuk menentukan kesamaan di antara mereka, sangat cocok diterapkan dalam sistem identifikasi gambar (Alqasir, Muselet, and Ducottet 2024).

Dalam penerapan teknologi *few-Shot learning* dan *Siamese Network* (Zhou et al. 2021) mengembangkan metode *siamese convolutional neural network* yang berhasil meningkatkan akurasi deteksi dan mengurangi false alarm rate pada sistem keamanan *cyber-physical*. (Fedele, et al. 2024) memperdalam pemahaman tentang mekanisme *Siamese Network* dalam konteks *few-Shot learning*, dengan temuan penting terkait kemampuannya mengidentifikasi karakteristik diskriminatif pada level intra-kelas dan antar-kelas. Sementara itu, (Argüeso et al. 2020) menghadirkan perspektif baru dengan membuktikan superioritas pada penerapan *few-Shot learning* berbasis *Siamese Network* dibandingkan transfer learning konvensional seperti *prototypical network*, mencapai akurasi hingga 24% untuk satu gambar per kelas dan 77,3% untuk 80 gambar per kelas. Hasil-hasil penelitian ini menjadi landasan yang kuat untuk pengembangan sistem pencarian berbasis gambar yang lebih mutakhir dengan mengintegrasikan teknologi *few-Shot learning* dan *Siamese Network* dalam konteks e-commerce kontemporer.

Dengan memanfaatkan kedua metode ini, sistem dapat lebih akurat dalam menemukan produk serupa berdasarkan gambar yang diunggah oleh pengguna. Implementasi sistem identifikasi gambar berbasis *few-Shot learning* dan *Siamese Network* di e-commerce diharapkan tidak hanya memudahkan konsumen dalam menemukan produk yang mereka inginkan tetapi juga meningkatkan pengalaman berbelanja secara keseluruhan. Dengan demikian, penerapan teknologi ini dapat memberikan keunggulan kompetitif bagi platform e-commerce dalam menjawab kebutuhan konsumen modern yang semakin mengandalkan visual dalam proses pembelian mereka.

1.2 Rumusan Masalah

Berdasarkan latar belakang yang telah disusun, rumusan masalah yang relevan untuk tugas akhir dengan fokus pada penerapan *few-Shot learning* dan *Siamese Network* dalam sistem identifikasi gambar di e-commerce dapat dirumuskan sebagai berikut:

1. Bagaimana pembentukan model *Siamese Network* menggunakan *backbone* Visual Geometri Group 16 (VGG-16) yang digunakan untuk klasifikasi dan identifikasi produk di *platform e-commerce*?
2. Bagaimana pembentukan model *Siamese Network* menggunakan *backbone* CNN yang digunakan untuk klasifikasi dan identifikasi produk di *platform e-commerce*?
3. Bagaimana hasil evaluasi untuk klasifikasi produk di *platform e-commerce* antara model *Siamese Network* dengan menggunakan *backbone* VGG-16 dan model CNN ?

1.3 Batasan Masalah

Dalam penelitian Untuk memastikan penelitian tetap fokus dan dapat diselesaikan dalam waktu yang tersedia, batasan masalah perlu ditetapkan dengan jelas. Berikut adalah beberapa batasan masalah yang relevan pada penelitian ini:

1. Ruang Lingkup *e-commerce* pada penelitian ini akan dibatasi pada platform *e-commerce* shopee.
2. Jenis produk sistem identifikasi gambar akan difokuskan pada kategori produk tertentu.

1.4 Tujuan

Berdasarkan latar belakang, rumusan masalah, dan batasan masalah yang telah ditentukan, tujuan penelitian ini dapat dirumuskan sebagai berikut:

1. Mendapatkan model klasifikasi untuk identifikasi gambar produk di *platform e-commerce* menggunakan *Siamese Network* menggunakan *backbone* VGG-16.
2. Mendapatkan model klasifikasi untuk identifikasi gambar produk di *platform e-commerce* menggunakan model VGG-16.
3. Mendapatkan hasil evaluasi untuk klasifikasi produk di *platform e-commerce* antara model *Siamese Network* dengan menggunakan *backbone* VGG-16 dan *backbone* CNN.

1.5 Manfaat

Penelitian ini diharapkan dapat memperkaya khazanah ilmiah dalam bidang computer vision dan e-commerce dengan menyoroti penerapan arsitektur *Siamese Network* berpadu *few-Shot learning* untuk identifikasi kemiripan visual produk. Dengan melakukan komparasi empiris antara *backbone* VGG-16 dan CNN, studi ini akan memberikan data uji yang konkret mengenai performa kedua arsitektur dalam tugas klasifikasi dan identifikasi gambar, sekaligus menguji efektivitas *few-Shot learning* pada kondisi data terbatas yang sering ditemukan di banyak aplikasi vision. Modul identifikasi visual ini berpotensi mendukung layanan customer support, misalnya pada verifikasi produk di proses retur atau klaim garansi, sehingga mempercepat dan mempermudah operasional. Selain itu, diharapkan dari hasil penelitian ini dapat menjadi acuan bagi penelitian selanjutnya dan dapat menjadi referensi dalam penelitian lain yang terkait pengembangan sistem identifikasi.

(Halaman ini sengaja dikosongkan)

BAB 2 TINJAUAN PUSTAKA

2.1 Hasil Penelitian Terdahulu

Penelitian mengenai sistem pencarian berbasis gambar telah banyak dilakukan dalam beberapa tahun terakhir, terutama seiring dengan perkembangan *e-commerce* dan teknologi kecerdasan buatan. Beberapa penelitian yang berkaitan langsung dengan penggunaan *Siamese Network* dan *Few-Shot Learning* dalam konteks pencarian produk dapat diringkas pada Tabel 2.1 dan 2.2.

Tabel 2.1 Daftar Penelitian yang Memiliki Penelitian Menggunakan Dataset E-commerce Untuk Penerapan Pada Sistem Identifikasi

Penelitian Sebelumnya	E-commerce dan Produk	Metode	Temuan	Kekurangan
(Ma'Rufah et al. 2023)	Shopee, OLX, eBay. Produk : Aksesoris Pakaian	Siamese Convolutional Neural Network dan Multi-Scale Convolutional Neural Network	Skenario <i>Siamese Network</i> menggunakan Model Convolutional Network (VGG-16, Inception-V3, dan CNN) didapatkan yang terbaik adalah VGG-16 dengan akurasi 97%.	Hanya menggunakan beberapa label Aksesoris Pakaian. <i>Output</i> Biner (Mirip dan Tidak Mirip)
(Panggabean et al. 2025)	Tiktok shop Produk : <i>Make-up</i>	Convolutional Neural Network	Convolutional Neural Networks (CNN) dapat mengidentifikasi dan mengklasifikasikan produk dengan akurasi tinggi.	Hanya menggunakan beberapa label <i>Make-up</i> .
(Devi 2016)	<i>E-commerce</i> buatan sendiri Produk : <i>smartphone</i>	Fitur Morfologi, Scale-Invariant Feature Transform (SIFT), Speeded Up Robust Features (SURF).	Perbandingan 3 metode tersebut dilakukan pada gambar <i>smartphone</i> yang memiliki 10 label, di usulkan metode Fitur Morfologi.	Akurasi yang didapatkan oleh metode Fitur Morfologi sekitar 82%, dan hanya menggunakan 10 label

Tabel 2.2 Daftar Penelitian yang Menggunakan *Few-Shot Learning* Berbasis Metode *Siamese Network* Dalam Analisisnya

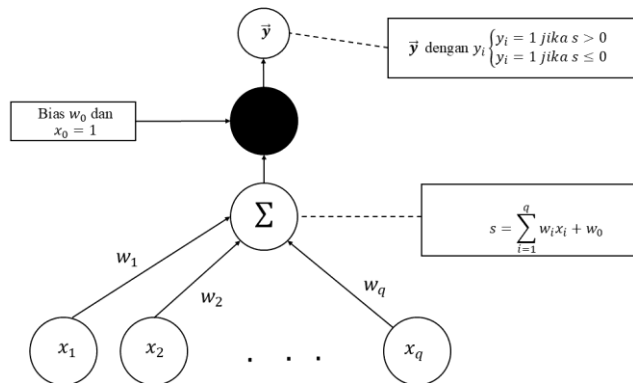
Penelitian Sebelumnya	Analisis	Dataset dan Metode	Temuan	Kekurangan
(Zhou et al. 2021)	Deteksi anomali di industri <i>cyber-physical sistem</i> menggunakan metode <i>Siamese Network</i>	<i>traffic packets, normal activity & contemporary attack behavior packets</i> Metode : Siamese Neural Network	Mengusulkan metode yang digunakan agar dapat menangani data berlabel yang tidak seimbang berdasarkan dataset yang digunakan. Didapatkan nilai akurasi yang tinggi 93%	Berdasarkan akurasi, model yang didapatkan memang tinggi, namun hanya bisa membedakan normal dan anomali saja (2 label).
(Heidari et al. 2020)	Pengenalan wajah menggunakan metode <i>Siamese Network</i>	Labeled Faces in the Wild (LFW) <i>dataset</i> Metode : <i>Siamese Network</i> dan VGG-16.	Dengan menggunakan data LFW menggunakan tersebut didapatkan akurasi sebesar 95.62%, dengan mengurangi total dataset sebesar 80% data.	Backbone yang digunakan hanya model pretrained VGG-16, tidak menggunakan seperti Imagenet, dsb.
(Argüeso et al. 2020)	Perbandingan <i>Few-Shot Learning</i> berbasis <i>Siamese Network</i> (FSL-SCNN) dengan transfer learning (TL) pada klasifikasi penyakit tanaman	Transfer Learning dan <i>Few-Shot Siamese Neural Network</i>	Dengan menggunakan metode TL lebih tinggi dibanding FSL-SCNN dengan selisih 4%, namun ukuran dataset yang digunakan di FSL-SCNN dapat dikurangi 89.1%	Berdasarkan penelitian tersebut ada saran untuk menambah label pada tanaman.

2.2 Artificial Neural Network

Artificial Neural Networks (ANN) adalah jenis *deep learning* yang didasarkan pada organisasi neural otak dengan parameter dinamis untuk memproses informasi secara efektif, dengan menghubungkan node input dan output secara sederhana dalam pemodelan data non-linier (Isabona et al. 2022). *Neural Network* digunakan bukan tanpa sebab, kemampuannya dalam mengidentifikasi pola dan tren yang terlalu rumit untuk dikenali manusia atau metode komputasi lainnya. Karena kemampuannya dalam mengenali pola dan tren, *neural network* sangat cocok untuk model prediksi atau peramalan (Ouali et al. 2020). Seiring dengan berkembangnya penelitian terhadap ANN, komponen penyusun ANN pun semakin kompleks, sehingga penggunaan akan *peceptron* dimultiplikasi sedemikian rupa membentuk *multilayer perceptron*. Pada subbab ini akan dibahas keduanya.

2.2.1 Single Layer Perceptron

Jenis paling dasar dari *artificial neuron* adalah *perceptron*. *Perceptron* terdiri dari input internal (bias), input eksternal, threshold, dan single output eksternal. *Perceptron* menerima sebuah vektor yang terdiri dari nilai input berupa bilangan riil dimana masing-masing nilai diberi bobot menggunakan pengali. Selama fase *training*, *perceptron* mempelajari bobot berdasarkan data *training*. *Perceptron* menjumlahkan nilai input berbobot dan "aktif" jika hasilnya melebihi ambang batas tertentu. Output dari *perceptron* berupa nilai boolean, dianggap "aktif" jika outputnya adalah '1'. Nilai "nonaktif" dari *perceptron* adalah '-1', dan nilai ambang batasnya adalah '0' (Staudemeyer 2019). Arsitektur *peceptron* tunggal terlihat pada Gambar 2.1.



Gambar 2.1 Ilustrasi Arsitektur Perceptron Tunggal
Telah diolah Kembali dari (Staudemeyer 2019)

Misalkan terdapat input data $x_i \in \mathbb{R}, i = 1, \dots, Q$; Q merupakan jumlah data input, lalu terdapat parameter $w_i \in \mathbb{R}$, dan y merupakan *output*. Maka didapatkan persamaan matematis pada Persamaan (2.1).

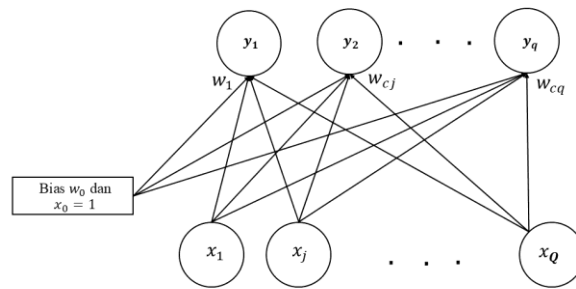
$$al = \sum_{i=1}^Q w_i x_i + w_0 \quad (2.1)$$

berdasarkan hal tersebut, w_0 adalah *intercept* atau parameter internal (bias) yang digunakan untuk menggeneralisasi pengaruh parameter pada data *input*. Pada penelitian kita ketahui bahwa *input* yang digunakan merupakan vektor, maka dari itu parameter dan *input* yang digunakan adalah berupa vektor $\vec{w} \in \mathbb{R}, \vec{w} = (w_0, w_1, \dots, w_q)^T$ dan $\vec{x} \in \mathbb{R}, \vec{x} =$

$(1, x_1, \dots, x_q)^T$, dimana vektor tersebut sudah teraugmentasi pada parameter memuat bias pada *input*, maka dari itu proses matematis perkalian vektor parameter $\vec{w}$ dengan input $\vec{x}$ pada Persamaan (2.2).

$$\vec{al} = \vec{w} \cdot \vec{x} \quad (2.2)$$

Pada kasus klasifikasi apabila data *input* $\vec{x}$ memiliki jumlah klasifikasi C berbeda dengan $C > 2$, maka jumlah dari *perceptron* yang digunakan sebanyak C . setiap *perceptron* memiliki vektor parameter $\vec{w}_c, c = 1, \dots, C$. persamaan matematis untuk klasifikasi $C > 2$ pada Persamaan (2.2). Arsitektur *peceptron Multi-Class* tunggal terlihat pada Gambar 2.2.



Gambar 2.2 Ilustrasi Arsitektur Perceptron Multi-Class

Berdasarkan persamaan (2.3) dan (2.4) $\mathbf{W}$ merupakan matriks berdimensi $C \times (Q + 1)$, yang berisi parameter yang digunakan dengan memuat w_{ci} . y_c merupakan kelas ke- c . Arsitektur *perceptron tunggal* dengan $C > 2$ pada Gambar 2.1.

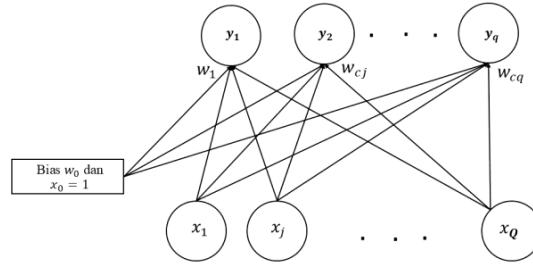
$$al = \sum_{i=1}^q w_{ci}x_i + w_{c0} = \vec{w}_c^T \cdot \vec{x} \quad (2.3)$$

$$al = \mathbf{W} \cdot \vec{x} \quad (2.4)$$

(James et al. 2023)

2.2.2 Multi Layer Perceptron

Pada umumnya ANN sering menggunakan *Multi Layer Perceptron* (MLP), Menurut Sarker MLP adalah jaringan yang terhubung penuh yang terdiri dari lapisan masukan yang menerima data *input*, lapisan *output* yang membuat keputusan atau prediksi tentang sinyal masukan, dan satu atau lebih lapisan tersembunyi di antara keduanya yang dianggap sebagai mesin komputasi jaringan (Sarker 2021). Keluaran jaringan MLP ditentukan menggunakan berbagai fungsi aktivasi, juga dikenal sebagai fungsi transfer, seperti ReLU (Rectified Linear Unit), Tanh, Sigmoid, dan Softmax. Selama proses pelatihan, berbagai pendekatan optimasi diantaranya adalah Adaptive Moment Estimation (Pedregosa et al. 2011). Arsitektur MLP pada Gambar 2.3.



Gambar 2.3 Ilustrasi Multilayer Perceptron 2 Layer

Misalkan terdapat MLP yang memiliki dua *layer*, lalu terdapat data *input* $\vec{x} = (1, x_1, \dots, x_q)^T, k = 1, 2, \dots, Q$, $\vec{w}_p = (w_{p0}, w_{p1}, \dots, w_{pq})^T$ adalah bobot dari *perceptron* ke- p di *layer* pertama, dan $\mathcal{G}(\cdot)$ merupakan suatu fungsi *non-linear*, dan *output layer* pertama dinyatakan sebagai $z_p; p = 1, 2, \dots, P$, dengan P merupakan jumlah *perceptron* yang ada di *layer* pertama. Lalu, didapatkan persamaan *output layer* pertama pada Persamaan (2.5).

$$z_p = \mathcal{G}(\vec{w}_p^T \cdot \vec{x}) = \mathcal{G}(\sum_{i=1}^q w_{pi}x_i + w_{p0}) \quad (2.5)$$

Berdasarkan *output* $\vec{z} = (z_1, z_2, \dots, z_p)^T$ yang didapatkan akan menjadi *input* di *layer* dua, dengan $\vec{v}_c = (v_{c0}, v_{c1}, \dots, v_{cq})^T; c = 1, 2, \dots, C$, dengan C merupakan jumlah kelas yang dihasilkan, dan $\mathcal{G}'(\cdot)$ merupakan suatu fungsi *non-linear* yang digunakan di *layer* dua. Lalu didapatkan *output* yaitu y_c didapatkan Persamaan (2.6)

$$y_c = \mathcal{G}'(\vec{v}_c^T \cdot \vec{z}) = \mathcal{G}'(\sum_{p=1}^P v_{cp}z_p + v_{c0}) \quad (2.6)$$

Pada model MLP dimana jumlah *layer* yang dihasilkan $L > 2$, maka dengan menyatakan $\vec{y}^{(l-1)} = \vec{z}^{(l)}$; $\vec{y}^{(l-1)}$ merupakan hasil dari *output layer* ke- $(l-1)$ dan $\vec{z}^{(l)}$ merupakan *input layer* ke- l , dan $l = 1, 2, \dots, L$, lalu jumlah *perceptron* disetiap *layer* ke- l dinyatakan sebagai $c^{(l)} = 1, \dots, Q^{(l)}$. Hasil *output* untuk *layer* L pada Persamaan (2.7).

$$y_c^{(l)} = \mathcal{G}'(u_{c^{(l)}}^{(l)}) = \mathcal{G}'\left(\sum_{c^{(l-1)}=1}^{Q^{(l-1)}} v_{c^{(l)}c^{(l-1)}}^{(l-1)} z_{c^{(l-1)}}^{(l-1)} + v_{c^{(l)}0}^{(l-1)}\right) \quad (2.7)$$

(James et al. 2023)

2.2.3 Fungsi Aktivasi

Fungsi Aktivasi secara khusus digunakan dalam *neural network* untuk mengubah sinyal *input* menjadi sinyal *output* yang selanjutnya dimasukkan sebagai masukan ke *layer* berikutnya dalam tumpukan. Dalam *neural network*, kita menghitung jumlah *input* dan bobotnya yang sesuai dan akhirnya menerapkan fungsi aktivasi untuk mendapatkan *output* dari *layer* tertentu dan memasukkannya sebagai *input* ke *layer* berikutnya (Sharma et al. 2020). Tujuan fungsi aktivasi adalah mentransformasi nilai *input* ke dalam rentang tertentu sesuai karakteristik fungsinya. Diantara berbagai fungsi aktivasi yang ada dan digunakan dalam penelitian ini adalah, *Rectified Linear Unit* (ReLU), *Softmax*, dan Sigmoid. Fungsi ini sering digunakan pada *Multilayer Perception*.

1. Rectified Linear Unit (ReLU)

Pada dasarnya fungsi ReLU melakukan “threshold” dari 0 hingga infinity (Firmansyah et al. 2022). Fungsi ini menjadi salah satu fungsi yang populer saat ini. fungsi aktivasi ReLU diterapkan. ReLU berfungsi untuk memperkenalkan *non-linearitas* ke dalam sistem. Fungsi ReLU didefinisikan pada Persamaan (2.8).

$$\mathcal{A}_{ReLU}(x_i) = \max(0, x_i) \quad (2.8)$$

Berdasarkan Persamaan (2.9), x_i merupakan *input* data $\mathbf{X}$, lalu $f(x_i)$ adalah *output* yang sudah ditransformasikan. ReLU mengubah semua nilai *input* negatif menjadi nol, sehingga *output* dari layer yang tidak menangkap pola penting dalam data tidak akan diproses lebih lanjut. Berdasarkan fungsi aktivasi yang ditunjukkan dalam persamaan (2.8), ReLU memiliki tingkat kompleksitas yang efisien, menjadikannya pilihan yang populer dalam arsitektur ANN.

2. Softmax

Fungsi aktivasi *softmax* merupakan komponen penting dalam *neural network*, terutama untuk tugas klasifikasi *multiclass*. Fungsi ini mengubah *output* $\vec{z}$ *hidden layer* menjadi distribusi probabilitas di berbagai kelas (Priya 2023).

$$\mathcal{A}_c(z_i) = \frac{e^{z_i}}{\sum_{c=1}^{C'} e^{z_j}}, c = 1, \dots, Q, -\infty < o_i < \infty \quad (2.9)$$

Berdasarkan Persamaan (2.9). probabilitas yang dihasilkan dari *output* tergolong kelas C_c dengan memandang setiap kelas $C_c, c = 1, 2, \dots, C'$ tidak bebas saling linier. $o_i = \mathcal{A}_c(y_i)$, fungsi diskriminasi untuk probabilitas yang dihasilkan oleh fungsi aktivasi *softmax* persamaan (2.10).

$$s(o_c) = C_c \text{ jika } \max(o_c); c = 1, \dots, C' \quad (2.10)$$

Dalam konteks klasifikasi, C_c merupakan hasil prediksi untuk input x_i ditentukan berdasarkan nilai probabilitas tertinggi dari x_i yang terklasifikasi sebagai C_c .

3. Sigmoid

Fungsi sigmoid adalah fungsi matematika yang dicirikan oleh kurva berbentuk *S*. Fungsi ini mengubah nilai input apa pun dari negatif tak terhingga hingga positif tak terhingga menjadi output antara 0 dan 1 (Mulindwa et al. 2023). Fungsi ini digunakan pada *output layer* dari model *multilayer perceptron*. Berdasarkan hal tersebut didapatkan fungsi sigmoid yang ada di Persamaan (2.11).

$$\mathcal{A}_{sigmoid}(y) = \frac{1}{1 + e^{-y}}, \text{ dengan} \quad (2.11)$$
$$-\infty < y < \infty, 0 < \mathcal{A}_{sigmoid}(y) < 1$$

Kemampuan fungsi sigmoid untuk mengubah bilangan riil apa pun menjadi nilai antara 0 dan 1 membuatnya sangat berguna untuk memperkirakan probabilitas, terutama dalam tugas klasifikasi biner (Mulindwa and Du 2023).

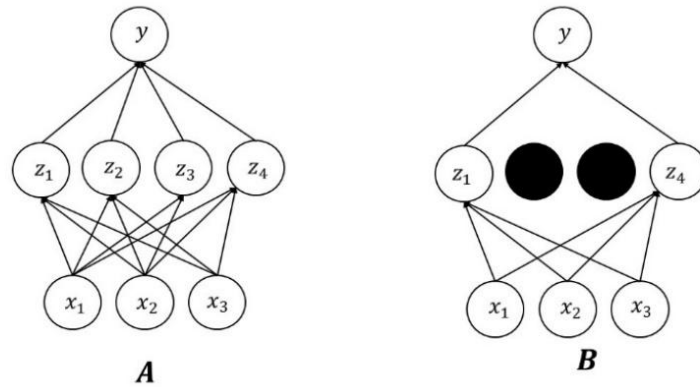
Pada penelitian ini, akan digunakan ketiga fungsi aktivasi tersebut, fungsi aktivasi ReLU digunakan sebagai Operasi yang digunakan oleh model *Visual Geometry Group 16 (VGG-16)*. Fungsi aktivasi *softmax* digunakan untuk mengklasifikasikan hasil prediksi yang digunakan pada kedua model yang akan diuji (*VGG-16* dan *Siamese Network*).

2.2.4 Dropout

Dropout adalah metode yang terinspirasi oleh random forest, idenya adalah untuk secara acak menghilangkan sebagian unit dalam suatu lapisan saat menyesuaikan model. Gambar 2.5

mengilustrasikan hal ini. Hal ini dilakukan secara terpisah setiap kali pengamatan pelatihan diproses. Unit yang bertahan menggantikan unit yang hilang, dan bobotnya ditingkatkan dengan faktor $\frac{1}{1-\theta}$ untuk mengimbangnya. Hal ini mencegah node menjadi terlalu terspesialisasi, dan dapat dilihat sebagai bentuk regularisasi. Dalam praktiknya, dropout dicapai dengan menetapkan aktivasi secara acak untuk unit yang "drop out" ke nol, sambil menjaga arsitektur tetap utuh.

Berdasarkan Ilustrasi Gambar 2.4, Dropout. **A** adalah jaringan yang terhubung sepenuhnya dan **B** adalah jaringan dengan dropout di lapisan input dan tersembunyi. Node berwarna hitam dipilih secara acak, dan diabaikan dalam contoh pelatihan



Gambar 2.4 Ilustrasi Dropout

2.2.5 Batch Normalization

Batch Normalization adalah teknik untuk melatih deep neural network membakukan input setiap lapisan mini-batch. Batch normalization memiliki efek menstabilkan proses training dan secara dramatis mengurangi jumlah periode pelatihan yang diperlukan untuk melatih *Neural Network*. Misalkan terdapat $x_i \in \mathbb{R}, i = 1, \dots, Q; Q, x_i$ menjadi Input gambar di ubah menjadi suatu matriks yang disebut matriks **X** yang memiliki dimensi $F \times G \times H$ dan parameter β dan γ , didapatkan persamaan statistik untuk *Bacth Nomralization* pada Persamaan (2.12) dan (2.12).

$$\mu_h = \frac{1}{mFG} \sum_{i=1}^Q \sum_{f=1}^F \sum_{g=1}^G x_{imfgh} \quad (2.12)$$

$$\sigma_c^2 = \frac{1}{mFG} \sum_{i=1}^K \sum_{f=1}^Q \sum_{g=1}^G (x_{imfgh} - \mu_h)^2 \quad (2.13)$$

Setelah menghitung statistik, kita melakukan normalisasi pada persamaan (2.13) untuk setiap piksel lalu menerapkan parameter yang dapat dipelajari γ (gamma) dan β (beta) untuk setiap channel (H) pada persamaan (2.14).

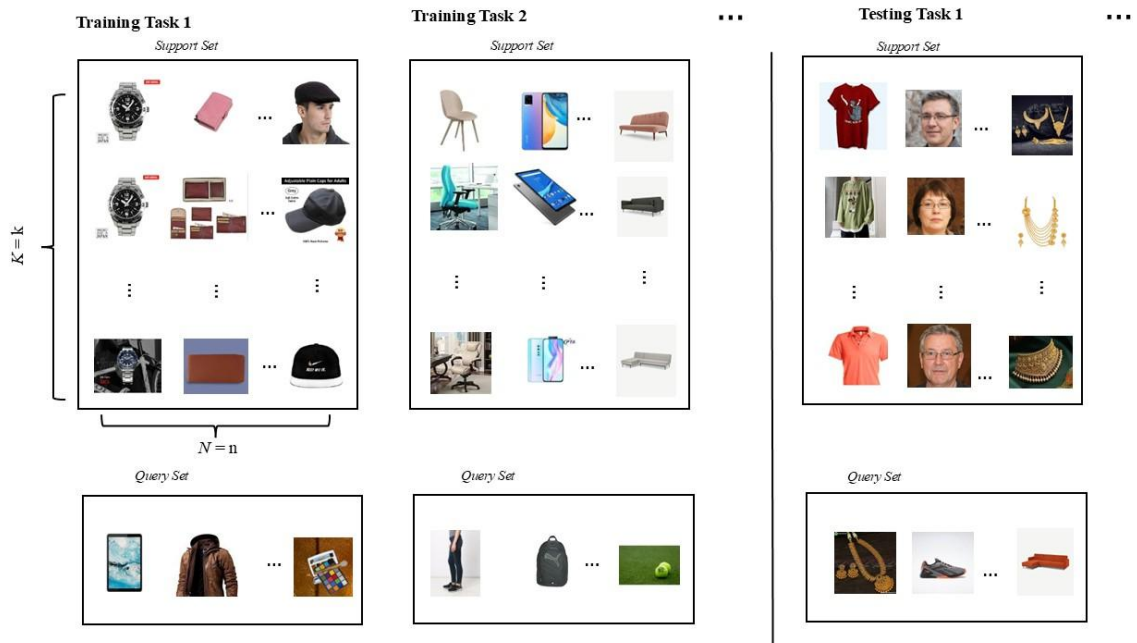
$$\hat{x}_{ifgh} = \frac{x_{ifgh} - \mu_h}{\sqrt{\sigma_c^2 + \epsilon}} \quad (2.13)$$

$$y_{ifgh} = \gamma_h \hat{x}_{ifgh} + \beta_h \quad (2.14)$$

(Furusho, Lee, and Suzuki 2019)

2.3 Few-Shot Learning

Few-Shot learning (FSL) sebagai konsep pertama kali diperkenalkan oleh Li Fei-Fei, Rob Fergus, dan Pietro Perona pada tahun 2006 dalam paper mereka yang berjudul "One-Shot Learning of Object Categories" (Parnami and Lee 2022). Dalam penelitian ini, mereka mengusulkan sebuah model komputasi yang dapat berhasil mempelajari kategori objek hanya dari satu contoh saja. FSL merupakan paradigma pembelajaran yang sedang berkembang yang bertujuan untuk mengatasi masalah kurangnya data pelatihan, yang memungkinkan model untuk mengidentifikasi kategori baru hanya dengan beberapa sampel data yang disediakan untuknya. Poin utamanya adalah bahwa sampel pelatihan perlu dipilih dengan cermat agar benar-benar sesuai dengan inferensi selama fase pengujian. Setiap langkah dirancang untuk mensimulasikan tugas pembelajaran sampel kecil dengan mengambil *sub-sampel* kelas dan titik data (misalnya, mengambil sampel lima kelas sekaligus, yang masing-masing memiliki lima sampel berlabel). Untuk menyelesaikan tugas FSL, *feature extraction* yang terlatih dengan baik harus dirancang, dan pengklasifikasi yang efektif sangat penting untuk menggali informasi yang kaya dari sejumlah kecil sampel berlabel.



Gambar 2.5 Ilustrasi Paradigma *Few-Shot Learning N-Way-K-Shot*

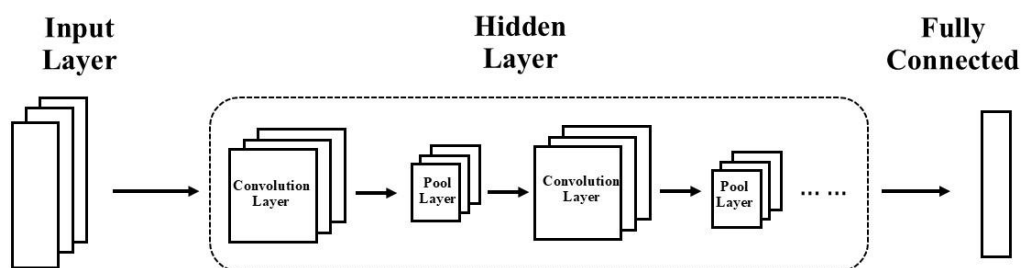
Berdasarkan Gambar 2.5, *Training Task* merupakan proses pembelajaran model FSL dengan berisi $N = n$ (Jumlah Label) dan $K = k$ (Jumlah Gambar) yang memiliki label yang berbeda dan bertujuan melatih model untuk mengenali perbedaan visual antar kelas. *Testing Task* merupakan tahap evaluasi dimana menerapkan model yang sudah dibuat sebelumnya, lalu

menerapkan di label yang berberda. Untuk Struktur pelatihan dan evaluasi FSL umumnya mengikuti kerangka kerja *N-Way-K-Shot*, di mana N mewakili jumlah label dan K mewakili jumlah Gambar (atau “percobaan”) yang disediakan untuk setiap kelas. Dalam klasifikasi *N-Way-K-Shot*, model mengalami beberapa episode pelatihan. Setiap episode pelatihan terdiri dari satu atau beberapa tugas pelatihan. Model dievaluasi melalui tugas-tugas tes, yang strukturnya mencerminkan tugas-tugas pelatihan. Setiap tugas pelatihan (dan tugas tes) terdiri dari dua kumpulan data (Maucher 2014).

Support Set berisi sampel pelatihan berlabel sebanyak N untuk masing-masing K Gambar. Model menggunakan sampel dukungan ini untuk mempelajari representasi umum untuk setiap kelas. Contohnya, kumpulan data untuk tugas klasifikasi *3-Way-2-Shot* berisi 3 kelas gambar dan menyediakan 3 contoh dari masing-masing kelas. Jika $K=1$, tugas ini merupakan pembelajaran satu contoh. Jika $K=0$, masalahnya adalah pembelajaran tanpa contoh—yang biasanya membutuhkan solusi unik, Lalu untuk *Query Set* berisi satu atau beberapa contoh baru untuk setiap kelas N yang disengaja tidak diberi label. Dengan menggunakan representasi yang dipelajari dari *Support Set*, model memprediksi klasifikasi untuk setiap contoh dalam *Query set*. Fungsi *Loss* mengukur perbedaan antara prediksi model dan prediksi yang “benar”, setelah setiap episode pelatihan, parameter model disesuaikan dan dioptimalkan, untuk meminimalkan Fungsi *Loss* (Xue et al. 2024).

2.4 Convolutional Neural Network

Convolutional Neural Network (CNN) adalah algoritma deep learning yang digunakan untuk memproses inputan data gambar, menentukan kepentingan (bobot dan bias yang dapat dipelajari) ke berbagai aspek dalam gambar dan berfungsi untuk membedakan objek satu dengan objek lainnya. CNN adalah jenis jaringan saraf tiruan yang dirancang khusus untuk memproses data visual, terutama gambar dan video. Sebenarnya CNN ini tidak jauh berbeda dengan neural network pada umumnya (Azmi et al. 2023). CNN bekerja dengan memecah gambar menjadi bagian-bagian kecil, kemudian memproses setiap bagian secara bertahap melalui lapisan-lapisan tersebut. Setiap lapisan belajar untuk mengenali fitur-fitur yang semakin kompleks, mulai dari tepi dan warna sederhana hingga bentuk dan objek yang lebih rumit.



Gambar 2.6 Arsitektur Convolutional Neural Network
Telah diolah Kembali dari (Ariyadi, et al. 2023)

CNN terdiri dari 2 lapisan arsitektur, yakni *featuring extraction* dan *classification layer* seperti terlihat Gambar 2.6. Lapisan *featuring extraction* diantaranya adalah *input layers*, *convolutional layer*, *pooling layers*, dan untuk *classification layer* diantaranya adalah *fully connected layers* dan *output layers*. Lapisan yang digunakan dalam CNN dapat dijelaskan sebagai pada Poin 1-5.

1. Input Layers

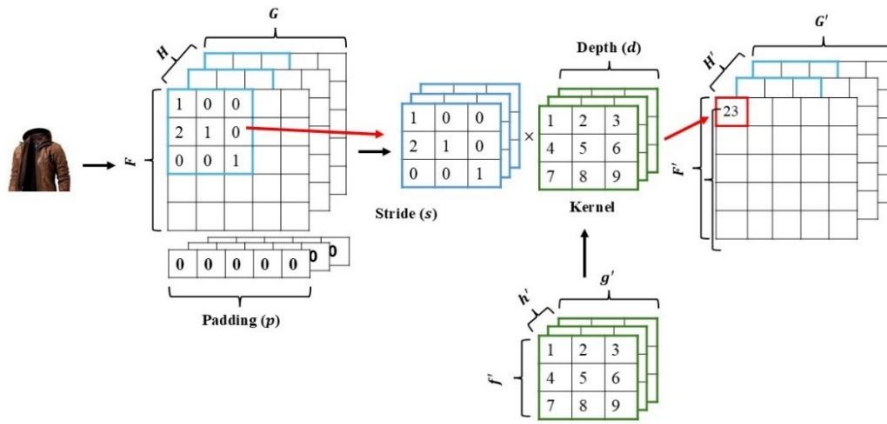
Lapisan ini berguna untuk menampung *pixel value* dari citra yang diinputkan. Citra batik tanah liat yang telah diinputkan memiliki 3 channel warna RGB (*Red, Green, Blue*).

2. Convolutional Layers

Convolutional Layers merupakan komponen fundamental dalam arsitektur (CNN) yang bertanggung jawab atas sebagian besar proses komputasi. Proses utamanya melibatkan operasi perkalian dot product antara matriks kernel dan bagian-bagian spesifik dari gambar input, yang kemudian menghasilkan peta fitur sebagai output. Operasi ini dilakukan secara berulang dengan menggeser kernel ke seluruh area gambar, yang dikenal sebagai proses konvolusi. Dalam implementasinya, *Convolutional Layers* memiliki beberapa parameter kunci yang mempengaruhi prosesnya. Parameter-parameter ini mencakup ukuran filter, *Stride*, dan *Padding*. Ukuran filter bisa disebut dengan *Depth* yang bisa dinyatakan d . *Stride* menentukan seberapa jauh kernel bergeser pada setiap langkah yang bisa dinyatakan s untuk menentukan banyak *Stride* yang dilakukan. *Padding* digunakan untuk menambahkan *pixel* bernilai nol untuk memanipulasi dimensi (Akhtar et al. 2022).

$$B_{(i',j',k')} = \sum_{c=1}^{h'} \sum_{a=1}^{g'} \sum_{b=1}^{f'} Z_{(a+i-1,b+j-1,c)} K_{er(a,b,c)} \quad (2.15)$$

Berdasarkan Gambar 2.7, Input gambar di ubah menjadi suatu matriks, di penelitian ini disebut matriks J yang memiliki dimensi $F \times G \times H$, dan output dari matriks disebut matriks B yang memiliki dimensi $F' \times G' \times H'$. Selanjutnya, di penelitian ini di cari matriks kernel yang cocok pada gambar input, berdasarkan Gambar, matriks kernel d memiliki dimensi $f' \times g' \times h'$. Berdasarkan parameter s , p , dan d , *pixel* dan *layers* yang didapatkan oleh matriks B adalah $F' = \frac{F+2p-f'}{s} + 1$ dan $G' = \frac{G+2p-g'}{s} + 1$. Misalkan terdapat elemen $Z_{(i,j,k)}$ dari matriks Z dan $B_{(i',j',k')}$ merupakan elemen dari B , dengan nilai $i = 1, 2, \dots, F$; $j = 1, 2, \dots, G$; $i' = 1, 2, \dots, F'$; $j' = 1, 2, \dots, G'$; $k = 1, 2, \dots, H$; dan $k' = 1, 2, \dots, H'$. Berdasarkan indeks yang sudah ditentukan nilai untuk elemen pada matriks Q dapat dirumuskan pada Persaman (2.15).

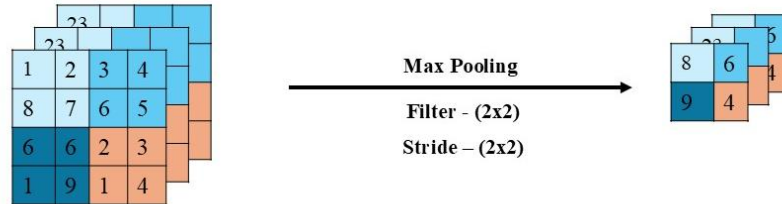


Gambar 2.7 Ilustrasi Proses Convolutional Layer
Telah diolah Kembali dari (Akhtar et al. 2022)

3. Pooling Layers

Pooling layer merupakan komponen esensial dalam arsitektur CNN yang berfungsi untuk melakukan downsampling pada feature map. Layer ini bekerja dengan mengambil nilai-nilai dari area lokal tertentu pada input dan menghasilkan satu nilai representatif, baik melalui pengambilan nilai maksimum (*max pooling*) atau perhitungan rata-rata (*average pooling*). Proses ini secara efektif mengurangi dimensi spasial dari data, sambil tetap mempertahankan informasi penting yang telah diekstrak oleh layer konvolusi sebelumnya (Edbert 2020).

Penempatan pooling layer yang strategis - biasanya setiap 2 hingga 3 layer konvolusi - memainkan peran krusial dalam arsitektur CNN. Pengurangan resolusi output ini tidak hanya menurunkan jumlah parameter yang diperlukan untuk pelatihan model, tetapi juga membantu mencegah overfitting dengan memberikan representasi data yang lebih kompak dan mudah dikelola (Edbert 2020). Dengan mengurangi dimensi data secara sistematis, pooling layer membantu model mencapai keseimbangan optimal antara kompleksitas komputasi dan kemampuan ekstraksi fitur, sambil tetap mempertahankan karakteristik penting dari data input.



Gambar 2.8 Max Pooling Layers dengan Ukuran 2x2 dan Stide 2
Telah diolah Kembali dari (Edbert 2020)

Berdasarkan Gambar 2.8, sebuah matriks input 4x4 diproses menggunakan operasi Max Pooling dengan filter berukuran 2x2 dan stride (2). Proses Max Pooling dengan persamaan (2.6) pada membagi matriks input menjadi empat region yang ditandai dengan warna berbeda. Setiap region warna memiliki nilai max yang dimiliki. Teknik ini efektif untuk mengurangi kompleksitas komputasi dan mencegah overfitting, sambil tetap mempertahankan fitur-fitur yang paling dominan dari data input. Penggunaan stride (2,2) memastikan bahwa setiap region diproses tanpa tumpang tindih, menghasilkan pengurangan dimensi yang konsisten. Misalkan $Z_{(i,j,k)}^v$ merupakan elemen matriks Z dari region v . Dan $Q_{(i',j',k')}^v$ elemen matriks Q dari region v . Dengan $i = 1, 2, \dots, F'$; $j = 1, 2, \dots, G'$; $u = 1, 2, \dots, H'$, $i' = 1, 2, \dots, F'$; $j' = 1, 2, \dots, G'$; $u' = 1, 2, \dots, H'$; dan f merupakan ukuran kernel sedangkan s adalah stride. Berdasarkan hal tersebut didapatkan perumusan *Max-Polling* pada Persamaan (2.16).

$$B_{(i',j',u')}^v = \max (Z_{((i.s):(i.s+f-1)(j.s):(j.s+f-1),u)}) \quad (2.16)$$

4. Flattening

Tahap flattening, vektor yang dihasilkan digunakan sebagai input untuk fully connected layer. Setiap lapisan pada fully connected layer dilengkapi dengan fungsi non linear. Fully connected layer adalah bagian dari jaringan saraf yang menghubungkan setiap neuron di lapisan tersebut dengan semua neuron di lapisan sebelumnya dan berikutnya seperti terlihat yang dijelaskan pada Poin 5. Oleh karena itu, sebelum dimasukkan ke fully connected layer, feature map dua dimensi tersebut diubah menjadi array satu dimensi atau vektor yang melalui proses flattening.

Misalkan terdapat L layer, $Z_{(i,j,u)}$ merupakan elemen output matriks Z , Dengan $i = 1, 2, \dots, F$; $j = 1, 2, \dots, G$; $k = 1, 2, \dots, H$. Persamaan yang digunakan dalam flatening terdapat pada Persamaan (2.17)

$$F_{flatten} = flatten(Z_{i,j,u}^{(L)}) \quad (2.17)$$

Fungsi flattening ini mengubah tensor multidimensi menjadi vektor satu dimensi dengan mempertahankan semua informasi namun mengubah strukturnya untuk dapat diproses oleh fully connected layer (Unzueta 2022).

5. Fully Connected Layers

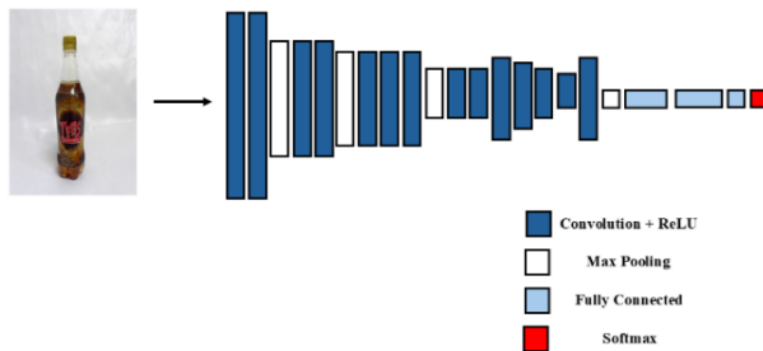
Fully Connected Layer (FC Layer) adalah komponen penting dalam arsitektur jaringan saraf tiruan, khususnya dalam Convolutional Neural Networks (CNN). Layer ini disebut "*fully connected*" karena setiap neuron pada layer tersebut terhubung dengan semua neuron di layer sebelumnya dan sesudahnya, menciptakan jaringan yang sangat terinterkoneksi (Unzueta 2022). Dalam struktur CNN, FC Layer ditempatkan di bagian akhir jaringan, setelah layer konvolusi dan pooling. Fungsi utamanya adalah untuk mengintegrasikan dan mengabstraksi fitur-fitur yang telah diekstraksi oleh layer-layer sebelumnya, serta mentransformasikannya menjadi bentuk yang sesuai untuk membuat prediksi akhir. FC Layer berperan penting dalam proses klasifikasi dan pengambilan keputusan akhir dari jaringan neural (Lakshmi et al. 2020).

Berdasarkan Gambar 2.6, Fully Connected Neural Network Layer, yang merupakan komponen fundamental dalam arsitektur jaringan saraf tiruan. Struktur ini melewati dua layer utama yang saling terhubung secara menyeluruh (*Input Layer* dan *Hidden Layer*). Dimana setiap layer memiliki jumlah neuron yang berbeda. Setelah melewati dua layer tersebut dilakukan transformasi menggunakan fungsi aktivasi. Berdasarkan hal tersebut, Misalkan, $Z_{(i)}$ merupakan elemen output matriks Z hasil *flattening*, Dengan $i = 1, 2, \dots, F$, dan $\mathcal{F}(\cdot)$ adalah fungsi aktivasi pada Persamaan (2.9) sampai (2.11) secara sistematis didapatkan hasil dari *fully connected layer* pada Persamaan (2.18).

$$y_i = \mathcal{F}(z_i) \quad (2.18)$$

Secara matematis, setiap neuron dalam FC Layer menerapkan transformasi linear terhadap *input layer* melalui matriks bobot, diikuti dengan transformasi non-linear menggunakan fungsi aktivasi yaitu *softmax* pada persamaan (2.9). Proses ini memungkinkan jaringan untuk mempelajari pola-pola kompleks dan hubungan non-linear dalam data. Meskipun FC Layer memiliki kemampuan yang kuat dalam mengkombinasikan fitur dan membuat prediksi, mereka juga memiliki jumlah parameter yang besar, yang dapat meningkatkan risiko overfitting dan membutuhkan komputasi yang lebih intensif (Unzueta 2022).

2.5 Transfer Learning Visual Geometry Group 16 (VGG-16)



Gambar 2.9 Arsitektur VGG-16

Telah diolah Kembali dari (Haile et al. 2024)

Transfer learning adalah teknik dalam machine learning di mana pengetahuan yang diperoleh dari satu tugas digunakan untuk meningkatkan kinerja pada tugas lain yang terkait. Dalam pendekatan ini, model yang telah dilatih sebelumnya pada dataset besar digunakan sebagai titik awal untuk menyelesaikan tugas baru, terutama ketika data untuk tugas baru tersebut terbatas. Proses ini melibatkan penggunaan fitur-fitur yang telah dipelajari dari tugas

awal dan mengadaptasinya untuk tugas target, sehingga mengurangi waktu pelatihan, kebutuhan data, dan sumber daya komputasi. Transfer learning sangat efektif dalam berbagai aplikasi, termasuk pengenalan gambar, pemrosesan bahasa alami, dan analisis data medis, memungkinkan pengembangan model AI yang lebih cepat dan efisien (FATURRAHMAN et al. 2023).

Visual Geometry Group 16 (VGG-16) merupakan salah satu arsitektur Convolutional Neural Network (CNN) yang populer dan sering digunakan untuk tugas-tugas computer vision, terutama klasifikasi citra. Dikembangkan oleh (Simonyan and Zisserman 2014), arsitektur VGG telah digunakan secara luas dalam berbagai aplikasi pengenalan gambar dan penambahan fitur. Ciri utama dari arsitektur VGG adalah penggunaan lapisan konvolusi kecil yang berurutan untuk mengekstraksi fitur yang semakin kompleks dari gambar. VGG-16 terkenal karena kesederhanaan dan efektivitasnya, serta kemampuannya untuk mencapai kinerja yang kuat pada berbagai tugas visi komputer, termasuk klasifikasi gambar dan pengenalan objek. Arsitektur model ini menampilkan tumpukan lapisan konvolusional diikuti oleh lapisan penyatuan maksimal, dengan kedalaman yang semakin meningkat.

Model VGG-16 yang diterapkan Gambar 2.9 memiliki akurasi pengujian 5 teratas sebesar 92,7% pada kumpulan data ImageNet yang berisi 14 juta gambar milik 1000 kelas. Desain ini memungkinkan model mempelajari representasi hierarki fitur visual yang rumit, sehingga menghasilkan prediksi yang kuat dan akurat. Meskipun sederhana dibandingkan dengan arsitektur yang lebih baru, VGG-16 tetap menjadi pilihan populer untuk banyak aplikasi pembelajaran mendalam karena fleksibilitas dan keunggulannya. Berdasarkan Gambar 2.4 dan Tabel 2.3, Perancangan model VGG-16 menggunakan input berukuran $224 \times 224 \times 3$, yang berarti memiliki panjang 224 piksel, lebar 224 piksel, dan depth 3, yang mewakili warna RGB. Selanjutnya, untuk jumlah filter, digunakan 64, 128, 256, dan 512 filter. Kernel yang digunakan berukuran 3×3 , dan lapisan max pooling memiliki ukuran 2×2 . Secara lengkap, perancangan algoritma VGG16 dalam penelitian ini dapat ditemukan dalam Tabel 2.3.

Menurut Puji kerangka kerja berbasis pembelajaran mendalam baru yang cocok untuk analisis elemen UI dalam situasi data yang langka, dengan fokus pada *few-Shot learning*. Kerangka kerja kami dimulai dengan beberapa modul *feature extraction* yang kuat yang menggunakan dan membandingkan model encoder canggih agar mahir dalam menangkap pola kompleks dari kumpulan data yang jarang (Puji et al. 2023). Pada Penelitian ini, *feature extraction* yaitu VGG-16. Sebagai contoh implementasinya pada Gambar 2.10.

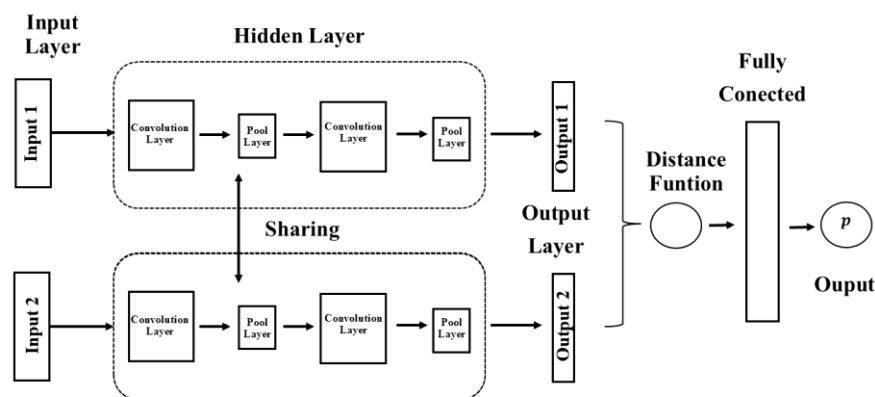
2.6 Siamese Network

Siamese Network pertama kali diperkenalkan pada awal tahun 1990-an oleh Bromley dan LeCun untuk menyelesaikan masalah verifikasi tanda tangan sebagai masalah pencocokan gambar (Boyle et al. 2022). *Siamese Network* adalah arsitektur neural network yang unik, dirancang untuk membandingkan dua input dan menentukan tingkat kemiripan di antara keduanya. Arsitektur ini terdiri dari dua *sub-network* identik yang memproses sepasang input secara paralel, berbagi bobot yang sama untuk memastikan konsistensi dalam pemrosesan. Cara kerjanya melibatkan pemasukan dua input ke masing-masing *sub-network*, ekstraksi fitur-fitur penting, dan penggabungan atau perbandingan fitur-fitur tersebut untuk menghasilkan nilai yang menunjukkan tingkat kemiripan. Kemiripan yang diukur bersifat kontekstual dan bergantung pada kasus penggunaan, seperti verifikasi wajah, analisis gaya berpakaian, pengenalan tanda tangan. Dalam proses pelatihan, model diberi pasangan data input dan menghasilkan output berupa nilai biner atau nilai *real* yang merepresentasikan tingkat kemiripan. *Siamese Network* merupakan alat yang ampuh untuk tugas-tugas yang melibatkan

pengukuran kesamaan atau verifikasi identitas. Kemampuannya untuk mempelajari fungsi kesamaan yang kuat membuatnya cocok untuk berbagai aplikasi.

2.6.1 Framework Siamese Network Secara Umum

Berdasarkan Gambar 2.10, *Siamese Network* memiliki beberapa karakteristik yang menjadikan pilihan populer untuk berbagai aplikasi pengenalan pola dan pembelajaran metrik, terutama dalam situasi dengan data pelatihan terbatas atau ketika perlu membandingkan entitas yang belum pernah dilihat sebelumnya yaitu.



Gambar 2.10 Ilustrasi Arsitektur *Siamese Network*
Telah diolah Kembali dari (Gao et al. 2021)

1. Twin Architecture

Kedua input (*Input 1* dan *Input 2*) diproses dengan subjaringan yang identik (*Hidden Layer*), yang memiliki arsitektur dan parameter yang sama untuk memastikan bahwa kedua input dilakukan transformasi dengan cara yang sama. Dalam *Siamese Network*, konsep *anchor* (*Input 1*) dan *positif* (*Input 2*) atau *negatif* (*Input 2*) digunakan untuk melatih model agar dapat membedakan kemiripan antara dua sampel. *Anchor* adalah sampel referensi yang menjadi titik acuan dalam proses pembelajaran. *Positif* adalah sampel lain yang memiliki kesamaan dengan *anchor*, misalnya gambar dari kategori yang sama dalam tugas klasifikasi gambar. Model dilatih untuk menghasilkan representasi vektor yang lebih dekat untuk pasangan (*anchor*, *positif*), sehingga dapat mengenali kemiripan antara objek yang serupa. Untuk meningkatkan kemampuan diskriminatif model, juga digunakan pasangan *anchor* dan *negatif*, di mana *negatif* adalah sampel yang berbeda dengan *anchor* agar model dapat memaksimalkan perbedaan dalam ruang vektor fitur. Gambar 2.10 merupakan arsitektur kembar yang digunakan pada *Siamese Network* (Verma 2024).

2. Sharing Weight

Prinsip *Sharing Weight* adalah berbagi bobot antar subjaringan, ketika kedua subjaringan menggunakan bobot yang sama, yang memastikan jaringan mempelajari fitur yang sama sehingga menjaga integritas perbandingan dan tahan dari *Overfitting* (Gao et al. 2021).

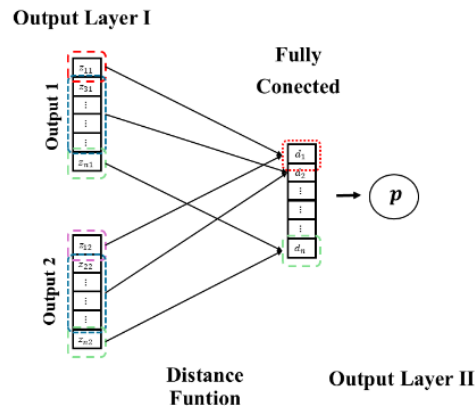
3. Similarity (Distance Function)

Karakteristik ini dirancang untuk mempelajari fungsi perbedaan antara pasangan yang “sama” dan “tidak sama”. Jaringan ini mengeluarkan vektor dari input yang sudah melewati *Hidden Layer* (*Output Layer*) dan kesamaan antara vektor ini dihitung menggunakan rumus *euclidean* Persamaan (2.19) (Verma 2024).

$$D_{12} = \sqrt{\sum_{m=1}^n (z_{1m} - z_{2m})^2} \quad (2.19)$$

Dimana, D_{ij} adalah nilai *euclidean (Similarity)* yang dihasilkan oleh *Input 1* dengan *Input 2*. Lalu, z_{im} merupakan *Output Layer 1* ke- m (Hasil vektor embbeding yang dilakukan oleh *Hidden Layer*) begitu juga dengan z_{2m} . m adalah jumlah piksel yang sudah ditentukan oleh penelitian ini pada *output layer*. Hasil perhitungan jarak ini menghasilkan nilai *similarity* antara 0 dan 1, di mana nilai yang lebih tinggi mengindikasikan kemiripan yang lebih besar antara kedua gambar input.

4. Fungsi Loss Siamese Network



Gambar 2.11 Ilustrasi Contrastive Loss

Siamese Network adalah arsitektur neural network yang unik, dirancang untuk membandingkan dua input dan menentukan tingkat kemiripan di antara keduanya. *Contrastive Loss* Karakteristik ini umum yang digunakan dalam pelatihan *Siamese Network*. Fungsi ini dirancang untuk meminimalkan jarak antara keluaran pasangan yang serupa dan memaksimalkan jarak antara keluaran pasangan yang tidak serupa. Keunggulan utamanya meliputi kemampuan menangani ketidakseimbangan kelas, pembelajaran kemiripan semantik, dan fleksibilitas dalam menangani kelas baru tanpa pelatihan ulang keseluruhan model (Rosebrock 2021). Terdapat satu pasang datum $\vec{x}_i$ dan $\vec{x}_j$ dengan $i = 1, 2, \dots, K; j = 1, 2, \dots, K; K$ merupakan jumlah data *training* jarak antara $\vec{x}_i$ dan $\vec{x}_j$ atau $D_{\vec{x}_i, \vec{x}_j}$ yang akan diminimalkan jika kelas sama, dan dimaksimalkan jika kodisinya sebaliknya. Oleh karena itu fungsi *loss* pada Persamaan (2.20) dan ilustrasi *contrastive loss* pada Gambar 2.11.

$$L_{SN}(Y, D_{\vec{x}_i, \vec{x}_j}) = (1 - Y) \frac{1}{2} D_{\vec{x}_i, \vec{x}_j}^2 + Y \frac{1}{2} \left\{ \max(0, m - D_{\vec{x}_i, \vec{x}_j}) \right\}^2 \quad (2.20)$$

Dimana, Y adalah label (bernilai 1 untuk pasangan yang sama, bernilai 0 untuk yang tidak sama), $D_{\vec{x}_i, \vec{x}_j}$ adalah nilai jarak *Euclidean* antara dan m adalah nilai batas margin yang ditentukan oleh peneliti.

5. Output

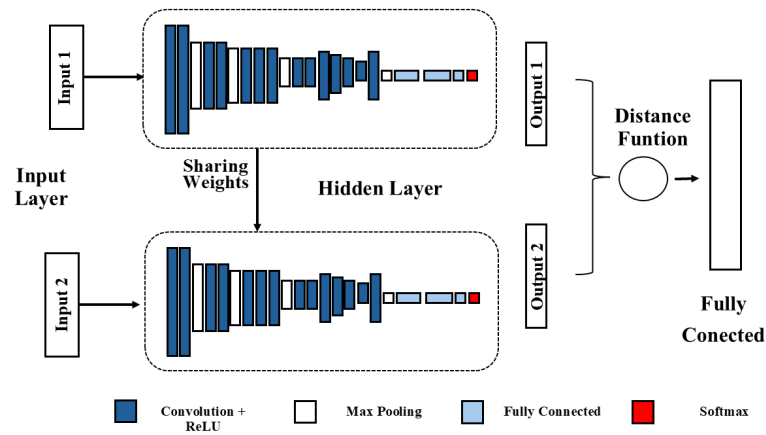
Berdasarkan gambar arsitektur *Siamese Network*, nilai p yang dihasilkan sebagai output merupakan nilai probabilitas yang diperoleh melalui fungsi aktivasi $\mathcal{F}(\cdot)$. Misalkan terdapat parameter $\beta_j, j = 1, 2, \dots, J$, output z_{1m} merupakan *Output Layer 1* dan z_{2m} merupakan *Output Layer 2*, secara matematis terdapat pada Persamaan (2.21).

$$p = \mathcal{F} \left(\sum_j^J \beta_j |z_{1m}^{(j)} - z_{2m}^{(2)}| \right) \quad (2.21)$$

Fungsi ini berperan penting karena mengubah hasil perhitungan jarak (distance) antara dua input menjadi nilai probabilitas yang dapat diinterpretasikan sebagai tingkat kemiripan antara kedua input tersebut.

2.6.2 Implementasi *Siamese Network* pada Model VGG-16

Pada Penelitian ini, proses identifikasi produk yang dilakukan menggunakan *Siamese Network* yang menggunakan *feature extraction* yang dimiliki oleh VGG-16 yang sudah dijelaskan pada sub-bab 2.6. Menurut Suyahman perbaruan arsitektur *Siamese Network* yang menggunakan berbagai kerangka kerja CNN, untuk membuat deteksi kesamaan citra merek dagang lebih akurat (Suyahman et al. 2024). Menggunakan ide *Siamese Network* dengan menggunakan *backbone* VGG-16 karena kemampuan pembelajaran transfernya yang kuat, dan strukturnya yang lugas untuk implementasi yang mudah, dan menggunakan arsitektur *Siamese Network* untuk melakukan perbandingan lebih lanjut (Li et al. 2019).



Gambar 2.12 Ilustrasi Arsitektur *Siamese Network* Menggunakan Backbone VGG-16
Telah diolah kembali dari (Li, Zhang, and Chen 2019)

Berdasarkan Gambar 2.12, menunjukkan keseluruhan struktur jaringan dari rancangan yang diusulkan pada penelitian ini dengan dua masukan (*Input 1* untuk gambar referensi dan *Input 2* untuk gambar contoh) dan satu keluaran untuk peta skor setelah menjalankan *Distance Function* antara dua masukan, yang dihasilkan berdasarkan hal tersebut adalah menunjukkan informasi kesamaan antara *Input 1* dan *Input 2*. Secara lengkap, perancangan algoritma *Siamese Network* yang menggunakan Arsitektur VGG16 dalam penelitian ini dapat ditemukan dalam Tabel 2.4.

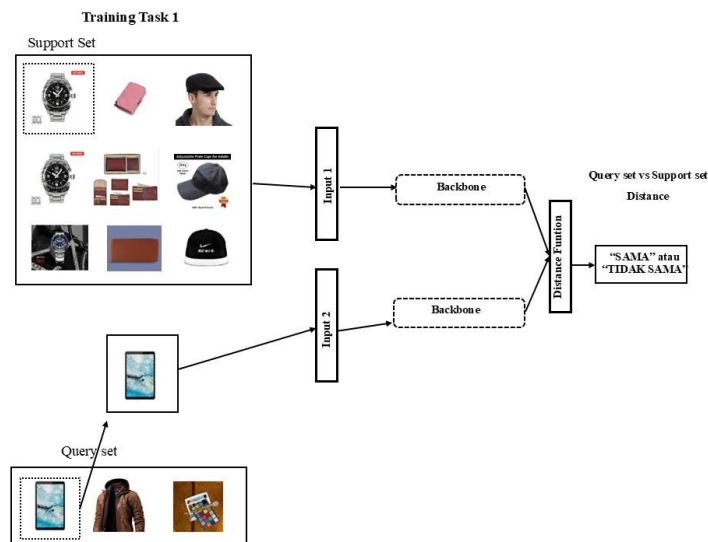
Tabel 2.3 Arsitektur model *Siamese Network* Menggunakan VGG-16

Layer		Operasi	Filter
Input 1	Input 2	-	-
Conv1	Conv1	Konvolusi + ReLU	3 x 3
Conv2	Conv2	Konvolusi + ReLU	3 x 3
Pool1	Pool1	Max Pooling	2 x 2
Conv3	Conv3	Konvolusi + ReLU	3 x 3
Conv4	Conv4	Konvolusi + ReLU	3 x 3
Pool2	Pool2	Max Pooling	2 x 2
Conv5	Conv5	Konvolusi + ReLU	3 x 3

Layer		Operasi	Filter
Conv6	Conv6	Konvolusi + ReLU	3 x 3
Conv7	Conv7	Konvolusi + ReLU	3 x 3
Pool3	Pool3	Max Pooling	2 x 2
Conv8	Conv8	Konvolusi + ReLU	3 x 3
Conv9	Conv9	Konvolusi + ReLU	3 x 3
Conv10	Conv10	Konvolusi + ReLU	3 x 3
Pool4	Pool4	Max Pooling	2 x 2
Conv11	Conv11	Konvolusi + ReLU	3 x 3
Conv12	Conv12	Konvolusi + ReLU	3 x 3
Conv13	Conv13	Konvolusi + ReLU	3 x 3
Pool5	Pool5	Max Pooling	2 x 2
Output 1	Output 2	-	-
FC1		Fully Connected + ReLU	-
FC2		Fully Connected + ReLU	-
FC3		Fully Connected	-
Distance		Euclidean Distance	-

2.6.3 Implementasi *Few-Shot Learning* Pada Arsitektur *Siamese Network*

Menurut Zhou, Secara *Siamese Network* yang berbasis CNN dibangun untuk mengatasi masalah *Few-Shot Learning*, sehingga kelas baru dapat diidentifikasi bahkan dengan hanya beberapa data sampel yang didukung (Zhou et al. 2021). Kerangka Pada penelitian ini menggunakan *backbone* VGG-16 yang sudah dijelaskan di Gambar 2.13. Berdasarkan Gambar 2.5 yang menggunakan pembelajaran *N-Way* dan *K-Shot* berdasarkan hitungan *N* dan *K*. Untuk membedakan antara kelas $N \leq 10$ dengan hanya $K \leq 10$ mungkin merupakan ukuran yang dipraktikkan secara luas (Faroz 2022). Sebagai contoh pada implementasi ini, dilakukan *3-Way-3-Shot*. Berdasarkan Gambar 2.13, Misalkan *Support Set* adalah matriks $S_{i,j}$, dengan $i = 1, 2, \dots, Q$, dan $j = 1, 2, \dots, N$, lalu *Query Set* adalah array $V_{i'}$ dengan $i' = 1, 2, \dots, K$. Setelah itu, $S_{i,j}$ masuk ke *Input 1* dan $V_{i'}$ masuk ke *Input 2*, dilakukan prediksi dengan menggunakan fungsi *Distance Euclidean* antara *Input 1* dan *Input 2*, lalu dilakukan fungsi *loss* agar untuk meminimalkan jarak antara keluaran pasangan yang serupa dan memaksimalkan jarak antara keluaran pasangan yang tidak serupa. Label dan jenis gambar yang masuk di *Traingin Task* tidak disertakan *Tesing Task*, sehingga *Support Set* dan *Query Set* tidak diekspos ke model *Siamese-VGG16*.



Gambar 2.13 3-Way-3-Shot Pada Arsitektur *Siamese Network* Pada *Training Task*

2.7 Evaluasi Model

Tabel 2.4 Confusion Matrix

	Predicted: 0 “Sama”	Predicted: 1 “Tidak Sama”
Actual: “Sama”	TN	FP
Actual: “Tidak Sama”	FN	TP

Untuk memastikan kinerja sistem identifikasi gambar di *e-commerce*, model harus dievaluasi menggunakan berbagai metrik evaluasi. Beberapa metrik yang sering digunakan dalam pengujian model identifikasi berbasis gambar meliputi akurasi, presisi, recall, F1-Score, dan AUC-ROC. Tabel 2.5 *Confussion Matrix* yang menunjukkan representasi yang lebih kompleks untuk *multiclass*. Pada Tabel 2.5, TN (True Negative) mewakili jumlah kelas negatif yang diklasifikasikan dengan benar, sementara TP (True Positive) mewakili jumlah kelas positif yang diklasifikasikan dengan benar. FP (False Positive) menunjukkan kelas negatif yang salah diklasifikasikan sebagai positif, dan FN (False Negative) adalah kelas positif yang salah diklasifikasikan sebagai negatif. Nilai yang berada pada diagonal utama menunjukkan hasil klasifikasi yang benar. Berdasarkan matriks ini, berbagai metrik evaluasi.

1. Akurasi

Akurasi merupakan metrik dasar yang mengukur proporsi prediksi yang benar terhadap total prediksi (Markoulidakis et al. 2021). Dalam konteks klasifikasi gambar, akurasi menunjukkan persentase gambar yang diklasifikasikan dengan benar oleh model. Akurasi dihitung menggunakan persamaan (2.21).

$$Akurasi = \frac{TP + TN}{TN + TP + FN + FP} \quad (2.21)$$

Meskipun akurasi memberikan gambaran umum performa model, metrik ini memiliki keterbatasan, terutama pada dataset yang tidak seimbang.

2. Presisi

Presisi mengukur keakuratan prediksi positif model. Metrik ini sangat penting ketika konsekuensi dari hasil positif palsu lebih besar daripada negatif palsu. Presisi dihitung dengan rumus pada Persamaan (2.22).

$$Presisi = \frac{TP}{TP + FP} \quad (2.22)$$

Presisi tinggi menunjukkan bahwa ketika model memprediksi kelas positif, prediksi tersebut cenderung benar (Markoulidakis et al. 2021). Metrik ini sangat penting ketika hasil positif palsu lebih merugikan daripada negatif palsu. Misalnya, dalam sistem rekomendasi musik atau video, hasil yang salah dapat menyebabkan ketidakpuasan dan kerugian pelanggan.

3. Recall

Recall, juga dikenal sebagai sensitivity, mengukur kemampuan model untuk mendeteksi semua sampel positif di antara semua sampel yang benar-benar positif (Markoulidakis et al. 2021).

$$Recall = \frac{TP}{TP + FN} \quad (2.23)$$

Recall dihitung menggunakan rumus pada Persamaan (2.23). Recall tinggi menunjukkan bahwa model mampu mengidentifikasi sebagian besar sampel positif yang ada.

4. F-1 Score

F1-score adalah metrik yang menggabungkan presisi dan recall menjadi satu nilai tunggal. Metrik ini memberikan gambaran yang lebih seimbang tentang performa model, terutama pada dataset yang tidak seimbang. F1-score dihitung menggunakan rumus Persamaan (2.24).

$$F1 - Score = 2 \cdot \frac{Presisi \cdot Recall}{Presisi + Recall} \quad (2.24)$$

F1-score berkisar antara 0 hingga 1, dengan 1 menunjukkan performa sempurna. Metrik ini sangat berguna ketika kita perlu menyeimbangkan presisi dan recall. *F1-Score* penting ketika positif palsu dan negatif palsu sama pentingnya. Ini menghitung rata-rata harmonik dari presisi dan perolehan. Keseimbangan ini memberikan wawasan yang lebih mendalam tentang performa model (Markoulidakis et al. 2021).

2.8 E-Commerce

E-commerce adalah kependekan dari *Electronic Commerce* atau perdagangan secara elektronik yang terjadi dalam jaringan elektronik. Sejak dikenalnya *e-commerce*, masyarakat dunia termasuk Indonesia merasakan kemudahan-kemudahan dalam berbagai hal yang sebelumnya tidak pernah mereka rasakan. *e-commerce* dapat dilihat dari berbagai perspektif, antara lain: (1) Dari perspektif komunitas, *e-commerce* merupakan pengiriman informasi, produk/layanan, atau pembayaran melalui lini telepon, jaringan komputer atau sarana elektronik lainnya. (2). Dari perspektif bisnis, *e-commerce* merupakan aplikasi teknologi menuju otomatisasi transaksi dan aliran kerja perusahaan. (3). Dari perspektif layanan, *e-commerce* merupakan satu alat yang memenuhi keinginan perusahaan, konsumen, dan manajemen dalam memangkas service cost ketika meningkatkan mutu barang dan ketepatan pelayanan. (4). Dari perspektif *online*, *e-commerce* berkaitan dengan kapasitas jual beli produk dan informasi di internet dan jasa on line lainnya (Annur 2022).

Shopee adalah sebuah *platform e-commerce* yang didirikan pada tahun 2015 di Singapura oleh Chris Feng dan Forrest Li, yang merupakan bagian dari SEA Group (sebelumnya dikenal sebagai Garena) Platform ini awalnya dirancang sebagai marketplace *consumer-to-consumer* (C2C) namun kemudian berkembang menjadi model hybrid yang mencakup *business-to-consumer* (B2C) melalui peluncuran Shopee Mall, yang menawarkan produk dari merek ternama. Shopee pertama kali diluncurkan di Singapura pada tahun 2015 dan segera memperluas jangkauannya ke negara-negara lain di Asia Tenggara, termasuk Malaysia, Thailand, Taiwan, Indonesia, Vietnam, dan Filipina. Pada tahun 2019, Shopee juga mulai beroperasi di Brasil, menjadikannya negara pertama di Amerika Selatan dan luar Asia yang dikunjungi Shopee (Agustina 2017).

Di sisi lain, Shopee mencatatkan peningkatan penjualan online hingga 200% pada tahun 2024, terutama dari produk lokal dan UMKM. Pertumbuhan ini didukung oleh program-program seperti Garansi Tepat Waktu dan kerja sama dengan Badan Penyelenggara Jaminan Produk Halal untuk mempermudah sertifikasi bagi UMKM. Selain itu, fitur baru Rangkuman Penilaian berbasis kecerdasan buatan (AI) memudahkan pembeli dalam membaca ulasan produk dengan lebih efisien (Leni 2024). Hal ini dapat disimpulkan bahwa *e-commerce* shopee sangat mendominasi di Indonesia dan Asia Tenggara.

BAB 3 METODOLOGI

3.1 Sumber Data

Sumber data yang digunakan pada penelitian ini yaitu data sekunder yang diperoleh dari *Kaggle* yang merupakan data terbuka (*opensource*). Data yang diambil adalah data yang sudah di *scrapping* Selenium Python dengan lebih dari 160K+ data produk yang bisa diakses <https://www.kaggle.com/datasets/adirizq/shopee-product-images>.

3.2 Data Penelitian

Pada penelitian ini, memanfaatkan data produk yang diperoleh dari platform e-commerce sebagai referensi. Data tersebut ditampilkan pada Tabel 3.2 dan terdiri dari beberapa atribut, antara lain `product_id` yang merupakan identitas unik untuk setiap produk, `image` yang berisi nama file atau representasi hash dari gambar produk, `shop_name` yang menunjukkan nama toko penjual, serta `shopid` sebagai identitas unik toko. Selanjutnya, terdapat atribut `main_category` yang menjelaskan kategori utama produk seperti “Elektronik”, dan `sub_category` yang memberikan klasifikasi lebih spesifik seperti “Konsol Game”. Informasi ini digunakan sebagai acuan dalam proses pencocokan antara postingan pengguna dengan data produk yang tersedia, guna mendukung sistem klasifikasi atau pencarian produk yang lebih akurat.

Tabel 3.1 Data Penelitian

<code>product_id</code>	<code>image</code>	<code>shop_name</code>	<code>shopid</code>	<code>main_category</code>	<code>sub_category</code>
12523520280	e6205d4f cb8a4852 35778ad2 10ac4d90	spirit elektronik	379815570	Elektronik	Konsol Game
17941082335	8563c9fc 31b880ed 105f91b8 d43e6d88	GamingÂ Console Store	268909000	Elektronik	Konsol Game
23517487660	id- 11134207 -7qul7- leyb85rn4 49s74	GAMOLBERKAH	102394348	Elektronik	Konsol Game
:	:	:	:	:	:
1335079260	f7081601 eea08bdd db24d4fd 20d3f08e	Full Cover Leather Case Fujifilm XT30 XT30 II XT20 XT10	6180218	Fotografi	Tas & Casing Kamera

Pada penelitian ini, diambil 15 label diantaranya pada Tabel 3.2, label 1-10 untuk training dan validasi, lalu label 11-15 digunakan untuk testing model.

Tabel 3.2 Label Data yang digunakan	
sub_category	Label Ke-
Album Foto	1
Buku Edukasi	2
Handphone	3
Kain	4
Laptop	5
Kotak P3K	6
Parfum	7
Printer Foto	8
Tas Kerja	9
Tote Bag	10
Billiard	11
Botol	12
Golf	13
Piringan Hitam	14
Voli	15

3.3 Struktur Data

Berdasarkan Tabel 3.2, Dalam penelitian ini, struktur data dapat dibagi menjadi dua bagian, yaitu struktur data secara umum dan struktur data yang lebih spesifik yang digunakan dalam pelatihan model. Struktur data umum diperlihatkan dalam Tabel 3.3, sementara struktur data yang lebih spesifik ditampilkan dalam Tabel 3.4. Berikut adalah deskripsi dari struktur data umum yang digunakan dalam penelitian ini.

Tabel 3.3 Sturtur Data Gambar		
No	Label Ke-	Gambar Produk Ke-
1	1	
500	1	
501	2	

No	Label Ke-	Gambar Produk Ke-
		
⋮	⋮	⋮
950	2	
951	3	
⋮	⋮	⋮
5000	10	

Setiap dari gambar diproses melalui lapisan convolutional dan pooling. Hasil akhirnya adalah array numerik berdimensi tetap yang disebut *feature vector*. Hasil yang merupakan struktur data dari setiap yang telah di ekstraksi pada Tabel 3.2.

		Tabel 3.4 Struktur Data Convolutional					
Gambar Ke-	Label Ke-	Struktur Variabel Convolutional					
		$X_{1,1}$	$X_{2,1}$	...	$X_{i,1}$	...	$X_{i,j}$
1	1	$x_{(1;1;1)R}$	$x_{(1;2;1)R}$	...	$x_{(1;i;1)R}$	...	$x_{(1;i;j)R}$
		$x_{(1;1;1)G}$	$x_{(1;2;1)G}$	...	$x_{(1;i;1)G}$	...	$x_{(1;i;j)G}$
		$x_{(1;1;1)B}$	$x_{(1;2;1)B}$	...	$x_{(1;i;1)B}$	...	$x_{(1;i;j)B}$
2	1	$x_{(2;1;1)R}$	$x_{(2;2;1)R}$	...	$x_{(2;i;1)R}$	...	$x_{(2;i;j)R}$
		$x_{(2;1;1)G}$	$x_{(2;2;1)G}$	...	$x_{(2;i;1)G}$	...	$x_{(2;i;j)G}$

		$x_{(2;1;1)B}$	$x_{(2;2;1)B}$	...	$x_{(2;i;1)B}$	...	$x_{(2;i;j)B}$
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\ddots$	$\vdots$	$\ddots$	$\vdots$
		$x_{(300;1;1)R}$	$x_{(300;2;1)R}$	...	$x_{(300;i;1)R}$	...	$x_{(300;i;j)R}$
300	2	$x_{(300;1;1)G}$	$x_{(300;2;1)G}$	...	$x_{(300;i;1)G}$	...	$x_{(300;i;j)G}$
		$x_{(300;1;1)B}$	$x_{(300;2;1)B}$	...	$x_{(300;i;1)B}$	...	$x_{(300;i;j)B}$
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\ddots$	$\vdots$	$\ddots$	$\vdots$
		$x_{(4500;1;1)R}$	$x_{(4500;2;1)R}$	...	$x_{(4500;i;1)R}$	...	$x_{(4500;i;j)R}$
5000	15	$x_{(4500;1;1)G}$	$x_{(4500;2;1)G}$	...	$x_{(4500;i;1)G}$	...	$x_{(4500;i;j)G}$
		$x_{(4500;1;1)B}$	$x_{(4500;2;1)B}$	...	$x_{(4500;i;1)B}$	...	$x_{(4500;i;j)B}$

3.4 Langkah Analisis

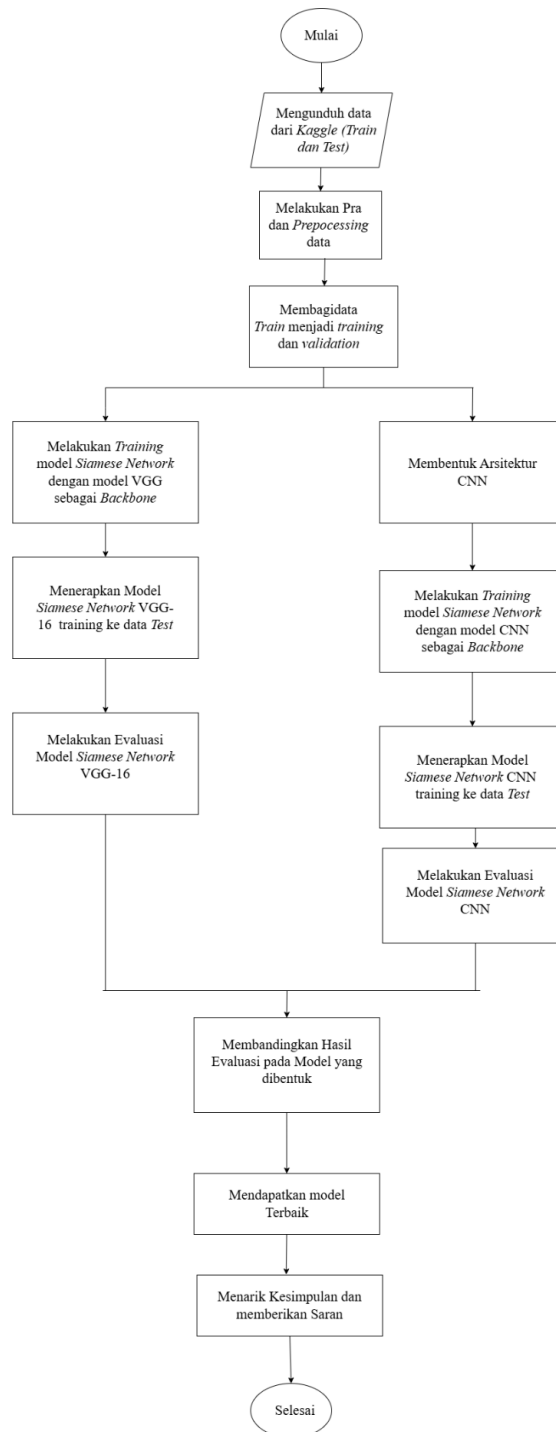
Langkah analisis yang akan dilakukan dalam penelitian ini adalah sebagai berikut.

1. Mendapatkan dataset dari *Kaggle* .
2. Melakukan *pre-processing* data.
3. Membuat *Training Set* dan *Testing Set*.
4. Menyiapkan *Support Set* dan *Query Set* di setiap *Training Set* dan *Testing Set* (N-Way-k-Shot), dan dijadikan sebagai *Training Task* dan *Testing Task*.
5. Membuatkan label pada gambar yang terpilih pada *Training Set* dan *Testing Set* “sama” dan “tidak sama”
6. Membangun arsitektur *Siamese Network* dengan *feature extraction* VGG-16.
 - a) Mengambil *Training Task*.
 - b) Mengambil sampel *Support Set* sebagai *Input 1* dan Mengsampil *Query Set* Sebagai *Input 2* dari *Training Task* yang akan diguanakn pada arsitektur *Siamese Network*.
 - c) Menghitung *Euclidean Distance* menggunakan Persamaan (2.19).
 - d) Menghitung *loss function* menggunakan persamaan (2.20).
 - e) Mendapatkan *Output* menggunakan persamaan.
 - f) Kembali ke-poin b) untuk mengambil sampel dari *Training Task* yang sama.
 - g) Kembali ke-poin a) untuk mengganti *Training Task*.
7. Mendapatkan Model *Siamese Network*, lalu menerapkan di *Testing Task*.
8. Membangun arsitektur *Siamese Network* dengan *feature extraction* CNN.
 - a) Mengambil *Training Task*.
 - b) Mengambil sampel *Support Set* sebagai *Input 1* dan Mengsampil *Query Set* Sebagai *Input 2* dari *Training Task* yang akan diguanakn pada arsitektur *Siamese Network*.
 - c) Menghitung *Euclidean Distance* menggunakan Persamaan (2.19).
 - d) Menghitung *loss function* menggunakan persamaan (2.20).
 - e) Mendapatkan *Output* menggunakan persamaan.
 - f) Kembali ke-poin b) untuk mengambil sampel dari *Training Task* yang sama.
 - g) Kembali ke-poin a) untuk mengganti *Training Task*
9. Mengatur konfigurasi hyperparameter seperti jumlah epoch, batch size, learning rate, dan strategi fine-tuning (freeze layer awal dan fine-tune layer atas) untuk arsitektur yang digunakan.

10. Melakukan save model untuk melakukan evaluasi pada data test.
11. Melakukan evaluasi model menggunakan Persamaan (2.26) pada Model *Siamese Network* VGG-16 dan CNN pada K-Shot.
12. Menentukan model terbaik berdasarkan performa tertinggi dan efisiensi komputasi.
13. Mendapatkan Hasil Terbaik.

3.5 Diagram Alir

Berdasarkan langkah penelitian pada subbab sebelumnya, dibuat diagram alir untuk penelitian ini yang ditunjukkan pada 3.1 sebagai berikut.



Gambar 3.1 Diagram Alir Penelitian

(Halaman ini sengaja dikosongkan)

BAB 4 HASIL DAN PEMBAHASAN

Bab ini membahas eksperimen dan analisis perbandingan performa dari implementasi arsitektur *Visual Geometry Group 16* (VGG-16) dan *Convolutional Neural Network* (CNN) sebagai *backbone* model *Siamese Network*. Subab 4.1 berisi spesifikasi *cloud service* penunjang eksperimen. Selanjutnya Subab 4.2 Membahas data penelitian Dilanjutkan oleh pembahasan alur penelitian secara umum pada Subbab 4.3. Adapun proses pra-pengolahan data dibahas pada Subbab 4.4. Selanjutnya penggunaan *hyperparameter* dibahas pada Subbab 4.5. Terakhir, analisis performa dari percobaan *backbone* VGG-16 dan CNN yang diimplementasikan pada arsitektur model *Siamese Network*, yang dibahas pada Subbab 4.6.

4.1 Spesifikasi Cloud Service Penunjang Eksperimen

Pada penelitian ini digunakan *cloud service* berbasis notebook, yaitu Kaggle, sebagai platform utama dalam pelaksanaan eksperimen. Kaggle menyediakan lingkungan pemrograman berbasis web yang menyerupai *Jupyter Notebook*, sehingga memudahkan pengguna untuk menulis dan menjalankan kode secara interaktif tanpa perlu melakukan instalasi lokal. Salah satu keunggulan utama Kaggle adalah dukungannya terhadap penggunaan GPU yang dapat mempercepat proses komputasi, terutama dalam pelatihan model berbasis *deep learning*. Spesifikasi lengkap dari sistem cloud dan lingkungan pemrograman yang digunakan pada penelitian ini ditampilkan pada Tabel 4.1.

Tabel 4.1 Spesifikasi Cloud Service

Cloud Hardware	
Processor	Intel® Xeon® CPU 2.30GHz
GPU	2× Nvidia T4 16GB VRAM
Memory (RAM)	25 GB
Disk	358 GB
Sistem Operasi	Ubuntu 18.04.3 LTS
Environment	
Bahasa Pemrograman	Python 3.7
Modul	
	1. PyTorch 1.4.0 CUDA 9.2
	2. Torchvision 0.5.0
	3. OpenCV-Python 4.40.46
	4. NumPy 1.17.2
	5. TensorBoard 2.4.0
	6. Matplotlib 3.3.3
	7. PyCocotools 2.0.2

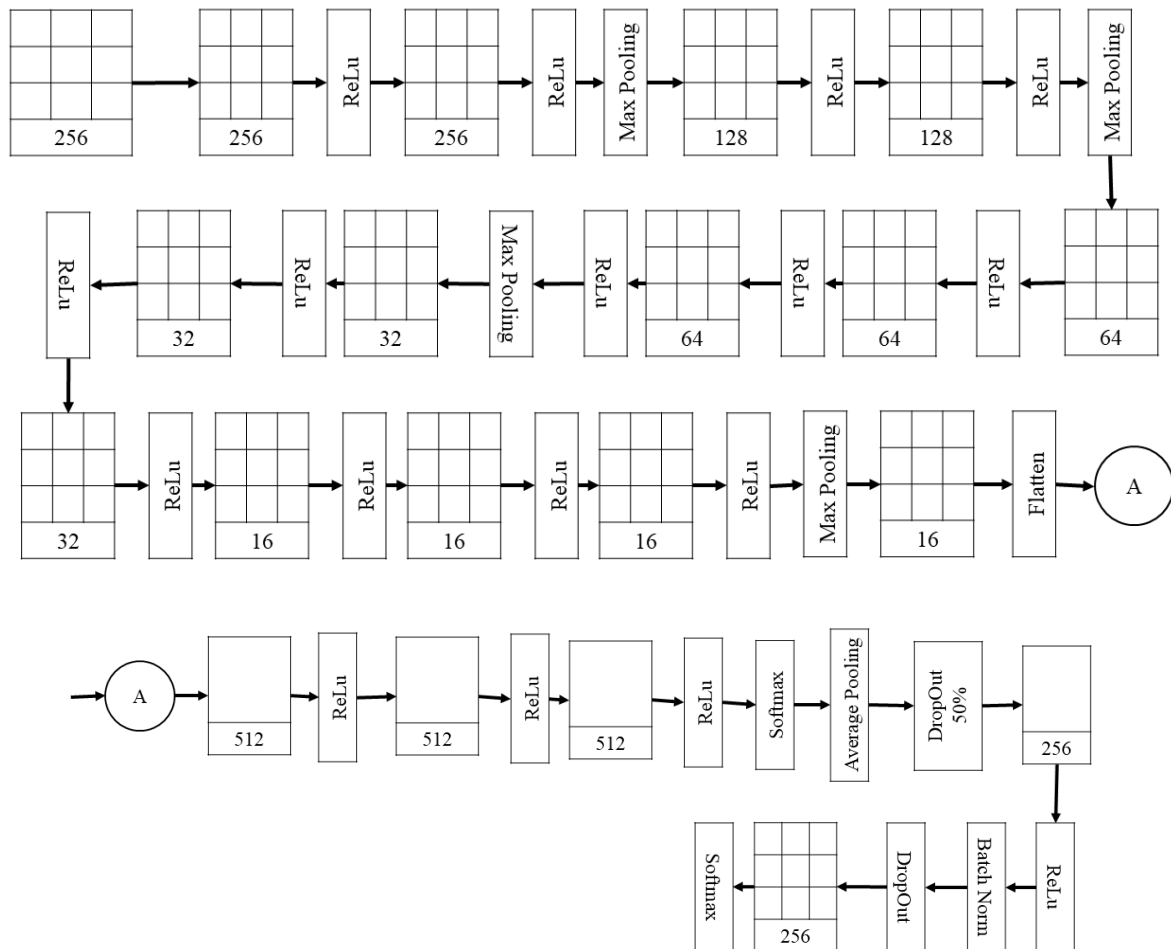
Platform Kaggle memungkinkan pengguna menjalankan program secara langsung pada lingkungan *cloud hardware* tanpa perlu konfigurasi tambahan. Lingkungan ini secara otomatis telah dilengkapi dengan spesifikasi sistem yang cukup tinggi, termasuk akses GPU secara gratis. GPU yang tersedia pada Kaggle biasanya berupa NVIDIA Tesla P100, T4, atau K80, yang masing-masing memiliki kapasitas komputasi dan memori yang memadai untuk pelatihan model skala menengah. Pada penelitian ini, model dilatih dengan dukungan GPU tipe T4x2

dengan kapasitas RAM sekitar 13 GB, yang sangat menunjang kebutuhan eksperimen *few-Shot* learning.

4.2 Arsitektur *Siamese Network* dengan VGG-16 dan CNN

Dalam penelitian ini, arsitektur *backbone* yang digunakan untuk membangun model *Siamese Network* terdiri dari dua jenis, yaitu model berbasis *transfer learning* VGG-16 dan model CNN rancangan sendiri (CNN). Kedua model tersebut berperan sebagai *feature extractor* yang bertugas menghasilkan representasi vektor dari citra input. Representasi vektor ini kemudian dibandingkan oleh komponen pembanding pada arsitektur *Siamese* untuk menentukan tingkat kemiripan antar dua gambar.

4.2.1 Backbone VGG-16



Gambar 4.1 Arsitektur VGG-16

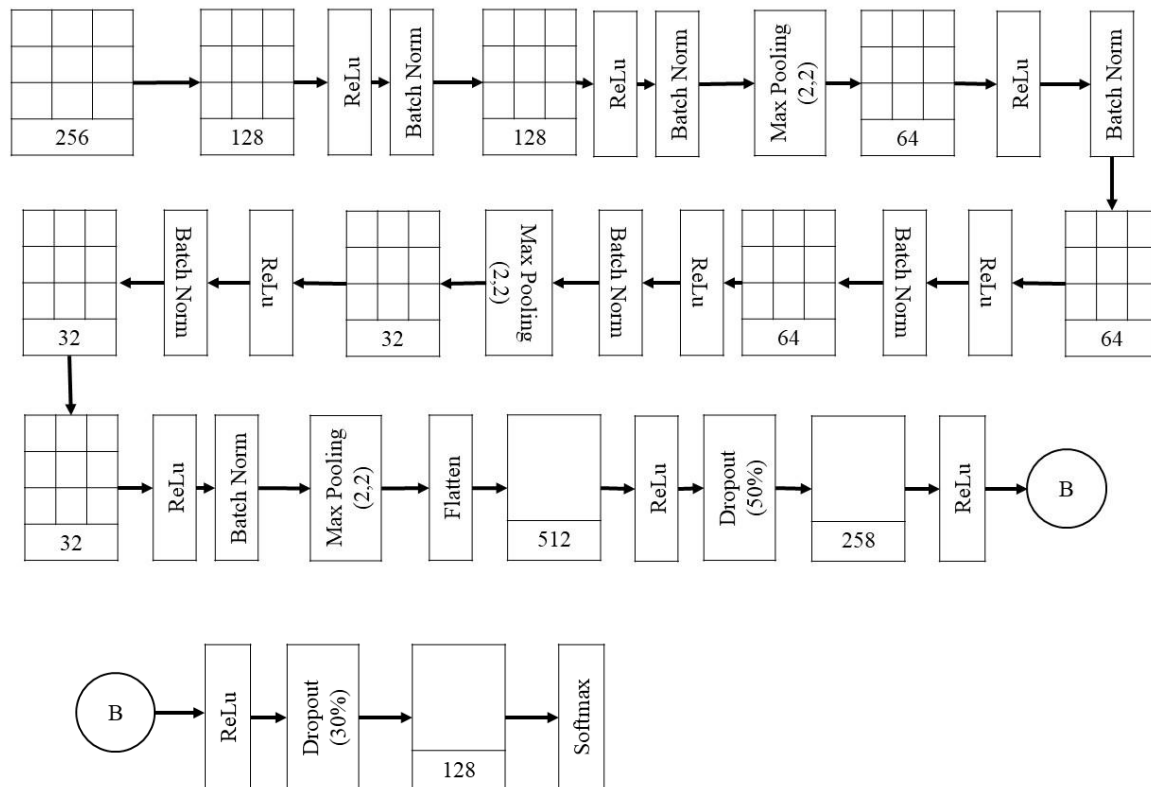
Transfer model CNN adalah arsitektur CNN pra-latih yang banyak digunakan sebagai *backbone* dalam berbagai tugas pengolahan citra karena bobotnya sudah teruji pada kumpulan data besar seperti ImageNet. Pada penelitian ini, VGG-16 (include_top=False) diimplementasikan sebagai *backbone* utama untuk model *Siamese Network*. Lapisan 0 hingga 9 dibekukan (*frozen*) untuk menjaga fitur dasar dari pretrained model, sedangkan lapisan ke-10 hingga lapisan atas dilakukan *fine-tuning* agar selaras dengan karakteristik citra e-commerce yang digunakan dalam eksperimen, parameter yang didapatkan dapat dilihat pada Lampiran 1, Summary dari Model VGG-16 yang digunakan pada penelitian ini.

Dengan pendekatan ini, model diharapkan mampu mempertahankan kemampuan generalisasi sambil tetap menyesuaikan diri dengan domain baru.

Implementasi backbone dilakukan menggunakan pustaka tensorflow Model dimuat menggunakan pretrained weights dari ImageNet dengan parameter `include_top=False`, sehingga bagian fully connected asli dari VGG-16 dihapus. Penyesuaian dilakukan dengan membekukan semua lapisan awal, kemudian mengaktifkan kembali lapisan ke-10 Gambar 4.1 menunjukkan rancangan arsitektur modifikasi dari VGG-16 yang digunakan sebagai cabang utama dalam jaringan Siamese. Arsitektur ini terdiri dari beberapa lapisan konvolusi yang diikuti dengan aktivasi ReLU dan *max pooling*, dilanjutkan dengan lapisan fully connected (FC) yang dioptimalkan untuk pembelajaran kesamaan citra.

VGG-16 dipilih sebagai *backbone* karena kemampuannya dalam mengekstrak fitur tingkat tinggi dari citra secara stabil, bahkan dengan data terbatas. Kombinasi kedalaman arsitektur dan pretrained weight dari ImageNet membuatnya sangat cocok untuk diterapkan pada pendekatan *few-Shot learning* melalui arsitektur *Siamese Network*.

4.2.2 Backbone CNN



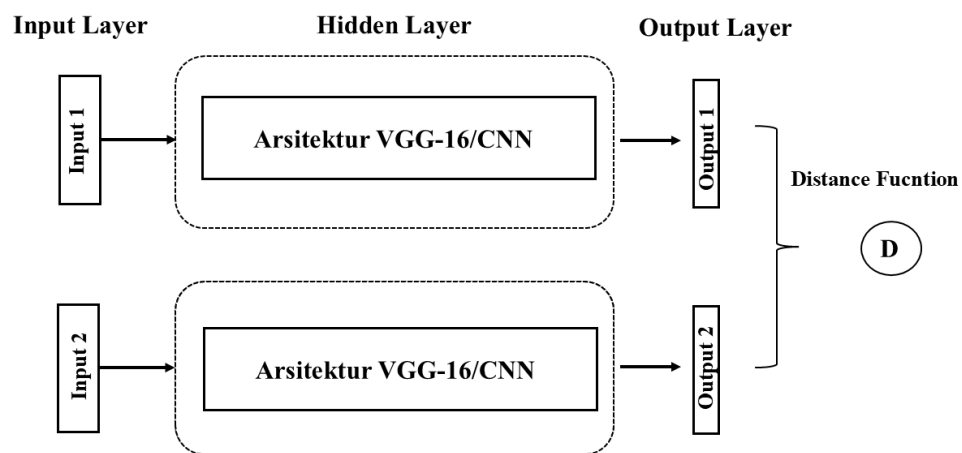
Gambar 4.2 Arsitektur CNN

Setelah melalui blok konvolusi awal dan tengah, fitur yang diperoleh diratakan (flatten) dan diteruskan ke lapisan fully connected dengan jumlah neuron berturut-turut sebanyak 512 dan 258 unit. Untuk menghindari overfitting, dropout dengan rasio 50% diterapkan di antara lapisan-lapisan tersebut. Arsitektur ini juga memanfaatkan Batch Normalization untuk mempercepat proses pelatihan dan menstabilkan distribusi aktivasi.

Penelitian ini juga mengembangkan arsitektur *Convolutional Neural Network* (CNN) buatan sendiri sebagai *backbone* alternatif dalam arsitektur *Siamese Network*. Arsitektur CNN ini dirancang secara modular dengan mempertimbangkan efisiensi komputasi, namun tetap mampu mengekstraksi fitur visual yang penting dari citra input. Seperti ditunjukkan pada Gambar 4.2, arsitektur dimulai dengan citra input berukuran $256 \times 256 \times 3$ yang diproses melalui beberapa blok konvolusi. Setiap blok konvolusi terdiri atas lapisan Conv2D dengan kernel 3×3 , diikuti dengan fungsi aktivasi ReLU, *Batch Normalization*, dan *Max Pooling* 2×2 untuk mengurangi dimensi spasial. Setelah melalui blok awal dan tengah, fitur yang diperoleh diratakan (flatten) dan diteruskan ke beberapa lapisan *fully connected* berukuran 512 dan 258 neuron. Untuk menghindari overfitting, digunakan teknik regularisasi *dropout* dengan rasio 50% di antara lapisan-lapisan tersebut. Kehadiran *Batch Normalization* juga membantu mempercepat proses pelatihan dan menstabilkan distribusi aktivasi pada jaringan.

Bagian akhir dari arsitektur ini menghasilkan representasi vektor fitur citra yang kemudian digunakan untuk proses perbandingan oleh cabang lain dalam struktur *Siamese Network*. Arsitektur CNN ini dipilih karena fleksibel untuk dilatih dari awal (*from scratch*), terutama ketika dataset yang digunakan memiliki karakteristik domain yang berbeda dari ImageNet, seperti citra produk e-commerce. Dengan kompleksitas yang lebih rendah dibandingkan VGG-16, model ini memberikan solusi yang lebih efisien dan ringan untuk skenario pembelajaran dengan data terbatas seperti pada *few-Shot learning*.

4.2.3 Konfigurasi *Siamese Network*



Gambar 4.3 Konfigurasi *Siamese Network* terhadap Arsitektur

Model yang digunakan dalam penelitian ini merupakan konfigurasi *Siamese Network* yang terdiri dari dua cabang arsitektur identik dengan parameter berbagi (shared weights), seperti diperlihatkan pada Gambar 4.3. Setiap cabang menerima satu citra input dan memprosesnya melalui arsitektur backbone usulan baik berupa VGG-16 yang telah dimodifikasi maupun CNN. Output dari masing-masing cabang berupa vektor embedding yang merepresentasikan fitur penting dari citra. Selanjutnya, kedua vektor embedding dibandingkan menggunakan fungsi jarak D , seperti Euclidean distance, untuk mengukur tingkat kemiripan antar citra. Output dari fungsi jarak ini digunakan sebagai dasar untuk menentukan apakah dua citra yang diberikan tergolong sebagai pasangan yang serupa atau tidak serupa. Arsitektur ini

dirancang untuk mengatasi keterbatasan data pelatihan melalui pendekatan *few-Shot* learning yang mengandalkan kemampuan model dalam mengenali kemiripan fitur meskipun data terbatas.

Tabel 4.2 Arsitektur *Siamese Network* VGG-16

Layer (type)	Output Shape	Param #	Connected to
input_layer (Input 1)	(256, 256, 3)	0	-
input_layer_1 (Input 2)	(256, 256, 3)	0	-
embedding_network (VGG)	(128)	14,879,936	input_layer, input_layer_1
lambda (Lambda)	(1)	0	embedding_network[0][0], embedding_network[1][0]
Total params: 14,879,936 (56.76 MB)			
Trainable params: 13,143,936 (50.14 MB)			
Non-trainable params: 1,736,000 (6.62 MB)			

Tabel 4.2 dan Tabel 4.3 menyajikan perbandingan arsitektur *Siamese Network* dengan backbone VGG-16 yang berkaitan dengan Gambar 4.1 dan CNN dan Gambar 4.2 secara detail. Model dengan backbone VGG-16 memiliki total parameter sebesar 14.879.936 dengan 13.143.936 parameter yang dilatih, sedangkan CNN memiliki jumlah parameter lebih besar, yaitu 17.282.218 yang hampir seluruhnya merupakan parameter yang dilatih dari awal. Perbedaan ini mencerminkan bahwa VGG-16 memanfaatkan bobot hasil pretraining, sehingga lebih efisien dalam waktu pelatihan dan stabil dalam performa, terutama pada skenario dengan jumlah data kecil atau besar (seperti *1-Shot* dan *5-Shot*). Di sisi lain, CNN menunjukkan performa yang kompetitif, bahkan unggul dalam konfigurasi data menengah (*2–4 Shot*), meskipun membutuhkan waktu pelatihan lebih lama dan cenderung kurang stabil pada jumlah data ekstrem.

Tabel 4.3 Arsitektur *Siamese Network* CNN

Layer (type)	Output Shape	Param #	Connected to
input_layer (Input 1)	(256, 256, 3)	0	-
input_layer_1 (Input 2)	(256, 256, 3)	0	-
embedding_network (CNN)	(128)	17,282,218	input_layer, input_layer_1
lambda (Lambda)	(1)	0	embedding_network[0][0], embedding_network[1][0]
Total params: 17,282,218 (65.93 MB)			
Trainable params: 17,280,614 (65.92 MB)			
Non-trainable params: 1,604 (6.27 KB)			

Secara keseluruhan, pemilihan arsitektur backbone dalam *Siamese Network* sangat bergantung pada kebutuhan konteks aplikasi. VGG-16 lebih disarankan pada kondisi data yang sangat terbatas atau sangat besar karena kestabilan dan efisiensinya, sementara CNN lebih

cocok untuk implementasi pada kondisi *few-Shot* menengah dengan kebutuhan fleksibilitas yang lebih tinggi terhadap variasi data.

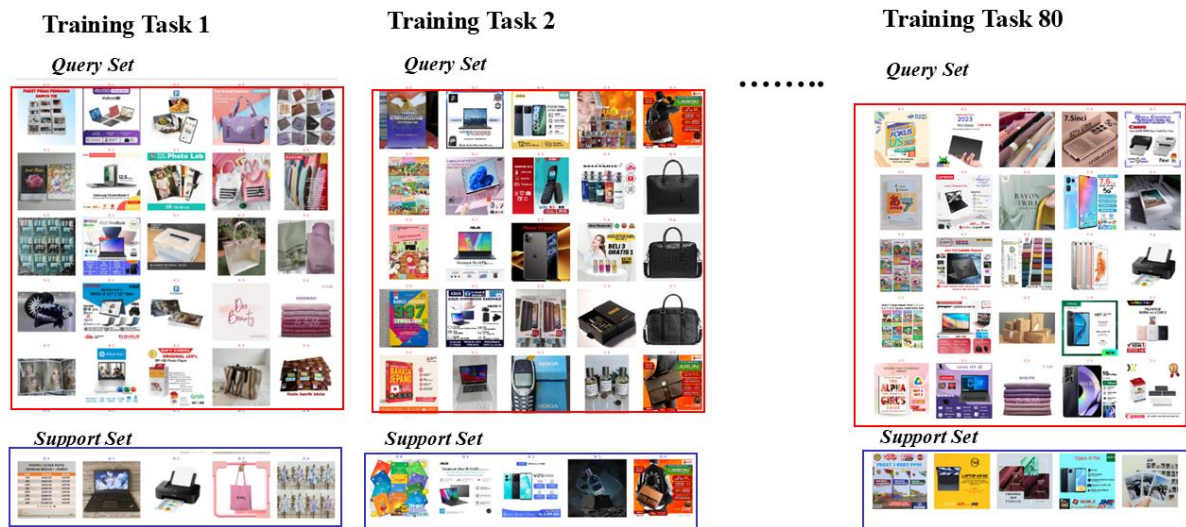
4.3 Pengolahan Data

Dataset utama yang digunakan dalam penelitian ini didapat dari Kaggle, yang memuat ribuan citra produk dari berbagai kategori e-commerce. Setiap gambar memiliki metadata kelas yang sudah tersedia; proses pengumpulan dilakukan dengan mendownload keseluruhan arsip dan mengekstrak citra ke dalam struktur penyimpanan berbasis folder. Karena penamaan folder dan label kelas telah dibahas pada bab sebelumnya, pada subbab ini fokus diarahkan pada mekanisme pairing dan pelabelan otomatis untuk keperluan *Siamese Network*.

4.3.1 Pra-Preprocessing Data

Langkah pra-preprocessing dimulai dengan implementasi skrip Python untuk memindai metadata kelas dari setiap file gambar. Dataset (Label ke-1 sampai ke-10) kemudian dibagi menjadi dua subset utama: 80 % untuk training dan 20 % untuk validasi. Untuk proses training, data diorganisir ke dalam episode sesuai paradigma *few-Shot learning*. Dalam setiap episode, secara acak dipilih 5 kelas berbeda (*5-Way*), dengan masing-masing kelas menyumbangkan K contoh citra support (K -Shot) serta satu citra query. Nilai K divariasikan antara 1 hingga 5, untuk mengevaluasi pengaruh jumlah contoh support terhadap performa model, seperti pada Gambar 4.4, Lalu untuk Ilustrasi *5-Way* 1 sampai 4 Shot dapat dilihat pada Lampiran 2.

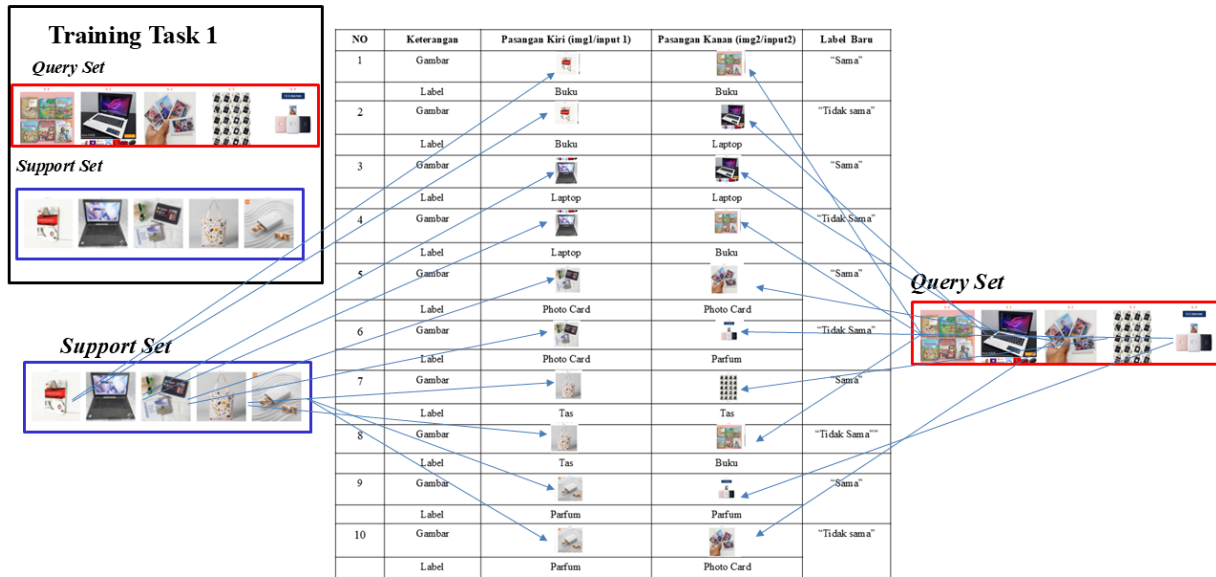
80% Data Train (Ilustrasi 5-Way 5 Shot)



Gambar 4.4 Alur Pra-Preprocessing Data (5 Way 5 Shot)

Setiap task membentuk suatu kumpulan yang terdiri atas $5 \times K$ pasangan support (dengan keseimbangan positif-negatif) dan 5 pasangan query. Format ini memberikan struktur input yang konsisten untuk melatih arsitektur *Siamese Network*. Berdasarkan label ini, sistem secara otomatis membentuk pasangan citra ($img1$, $img2$) secara acak dimana $img1$ berasal dari query set yang sebagai citra yang ingin dilakukan test an $img2$ menjadi support set yang menjadi pembanding kemiripan untuk query set. Jika kedua gambar memiliki label kelas yang sama, maka pasangan tersebut diberi label “sama”; jika berbeda, diberi label “tidak sama”. Pembagian dilakukan secara stratified untuk memastikan proporsi pasangan “sama” dan “tidak sama” tetap seimbang di kedua subset. Selanjutnya Gambar 4.5 menggambarkan alur pembentukan

pasangan citra dan proses pra-preprocessing yang diterapkan dalam penelitian ini. Proses ini dilakukan baik untuk subset training maupun testing.



Gambar 4.5 Proses pembuatan pasangan untuk Arsitektur *Siamese Network*

Pada Tabel 4.4, Pada setiap task *5-Way K-Shot*, sistem menghasilkan total pasangan citra yang terdiri dari pasangan positif (berlabel “sama”) dan pasangan negatif (berlabel “tidak sama”). Jumlah ini diperoleh dari 5 citra query, masing-masing dipasangkan dengan 5 support image dari kelas yang sama sebagai pasangan positif, serta 5 support image dari kelas lain sebagai pasangan negatif. Strategi ini memastikan bahwa jumlah pasangan positif dan negatif tetap seimbang, terlepas dari nilai K yang divariasikan dari 1 hingga 5. Meski jumlah citra support per kelas meningkat seiring bertambahnya nilai K, proses pemilihan pasangan dilakukan secara acak dan terkontrol, sehingga jumlah pasangan tetap konstan. Pendekatan ini menjaga konsistensi ukuran batch input dan kestabilan proses pembelajaran dalam arsitektur *Siamese Network*. Dengan struktur episode yang terstandarisasi, model dapat dilatih untuk menggeneralisasi kemiripan antar kelas secara efektif meskipun hanya menggunakan sedikit contoh.

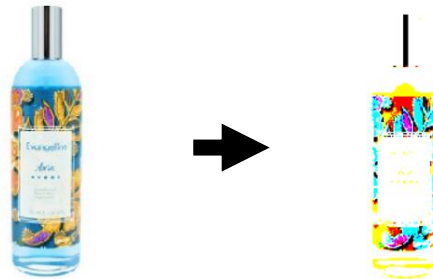
Tabel 4.4 Jumlah pairs yang terbentuk pada kombinasi setiap *task*.

<i>Shot</i>	Jumlah Pairs (Per Task)	Total Pairs
1	10	800
2	20	1600
3	30	2400
4	40	3200
5	50	4000

4.3.2 Preprocessing Data

Pada tahap preprocessing, setiap citra produk terlebih dahulu diubah ukurannya menjadi resolusi tetap sebesar 256×256 piksel agar sesuai dengan format input yang dibutuhkan oleh arsitektur backbone VGG-16. Proses ini dimulai dengan membaca file gambar dari penyimpanan, kemudian mengonversinya ke dalam representasi array numerik bertipe float32

dengan tiga kanal warna (RGB). Selanjutnya, citra tersebut diproses menggunakan fungsi `preprocess_input()` dari Keras Applications, yang merupakan prosedur standarisasi input khusus untuk model VGG-16 pretrained. Fungsi ini secara deterministik melakukan konversi channel warna dari RGB ke BGR, serta mengurangi nilai rata-rata pixel pada setiap channel berdasarkan distribusi dataset ImageNet (yakni 103.939 untuk kanal biru, 116.779 untuk kanal hijau, dan 123.68 untuk kanal merah).



Gambar 4.6 Process Input Menggunakan VGG-16

Berbeda dengan augmentasi data yang bersifat acak dan bertujuan untuk memperkaya variasi data pelatihan, proses ini bersifat tetap dan konsisten, serta ditujukan untuk menyamakan distribusi input dengan kondisi pelatihan awal model. Oleh karena itu, tidak dilakukan augmentasi seperti rotasi, flipping, atau penyesuaian brightness dalam penelitian ini. Preprocessing ini diterapkan secara identik terhadap seluruh data, baik pada subset training maupun validasi, serta untuk pasangan gambar support dan query dalam setiap episode pelatihan. Visualisasi hasil transformasi ini dapat dilihat pada Gambar 4.6, di mana tampak bahwa nilai visual pada citra berubah secara signifikan akibat transformasi numerik, meskipun bentuk dan struktur utama dari objek tetap terjaga.

4.4 Analisis Perbandingan Performa *Backbone* VGG-16 dan CNN

Penelitian ini mengevaluasi dua arsitektur *backbone* yang digunakan dalam model *Siamese Network*, yaitu VGG-16 dan CNN yang dirancang dari awal. Kedua model diuji dalam skema *few-Shot learning* dengan pendekatan *5-Way K-Shot*, di mana nilai K divariasikan dari 1 hingga 5. Tujuan utama dari eksperimen ini adalah untuk menilai sejauh mana masing-masing backbone mampu mengekstrak fitur representatif dan melakukan generalisasi terhadap kelas-kelas baru dalam kondisi data terbatas. Data yang digunakan pada Perbandingan ini terdapat pada Tabel 4.5.

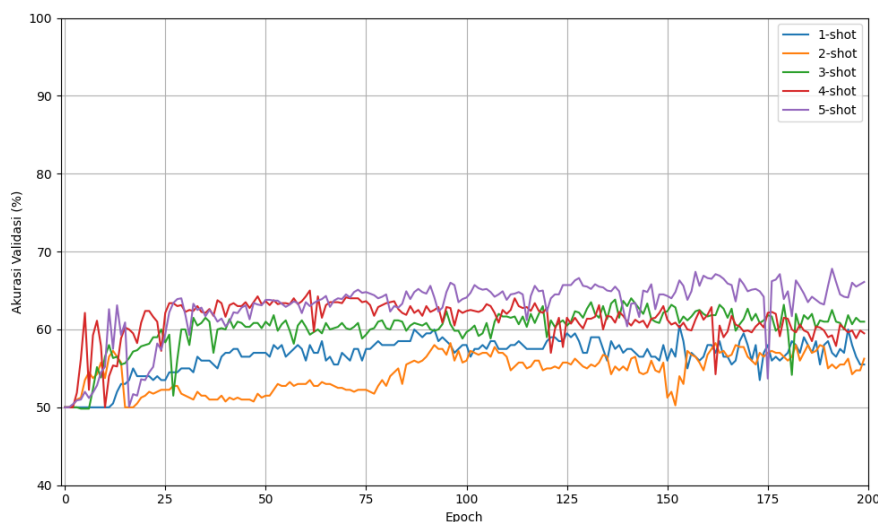
Tabel 4.5 Data Training dan Validasi

sub_category	Label Ke-
Album Foto	1
Buku Edukasi	2
Handphone	3
Kain	4
Laptop	5
Kotak P3K	6
Parfum	7
Printer Foto	8
Tas Kerja	9
Tote Bag	10

Evaluasi performa dilakukan dengan menggunakan dua metrik utama, yaitu akurasi dan *contrastive loss*, baik pada data pelatihan maupun data validasi. Akurasi mencerminkan kemampuan model dalam membedakan pasangan gambar yang “sama” dan “tidak sama” secara tepat, sedangkan *contrastive loss* menunjukkan efektivitas pembelajaran representasi fitur pada ruang vektor yang digunakan untuk perbandingan kemiripan. Dengan membandingkan kurva akurasi dan loss dari kedua backbone untuk setiap variasi K , analisis ini bertujuan mengidentifikasi arsitektur yang paling optimal dalam konteks *few-Shot similarity learning*. Hasil dari eksperimen ini dijabarkan lebih lanjut pada subbab 4.4.1 dan 4.4.2, yang masing-masing membahas performa VGG-16 dan CNN secara terpisah, sebelum dilakukan komparasi akhir pada subbab 4.4.3.

4.4.1 Few-Shot Learning Siamese Network Backbone VGG-16

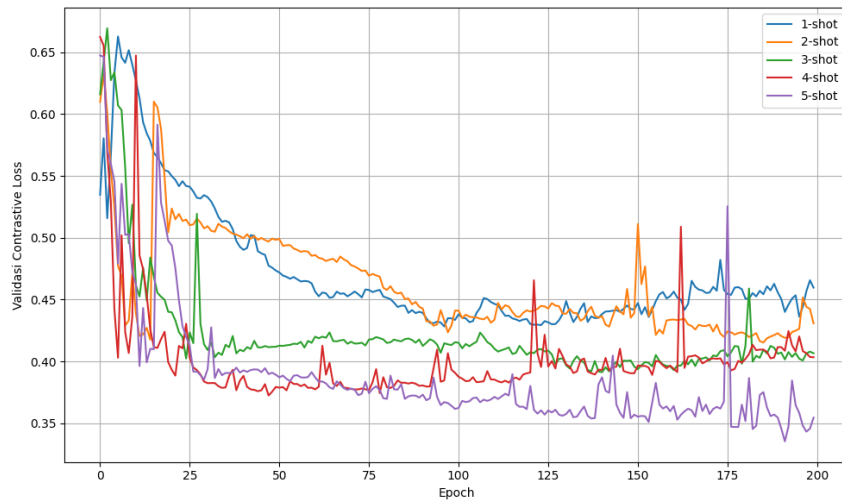
Evaluasi performa model *Siamese Network* dengan backbone VGG-16 dilakukan dengan pendekatan *5-Way K-Shot*, di mana nilai K divariasikan dari 1 hingga 5. Tujuan utama adalah mengamati pengaruh jumlah contoh support terhadap kemampuan model dalam mengidentifikasi kemiripan citra, khususnya dalam skenario data terbatas. Hasil pengujian disajikan melalui metrik akurasi dan *contrastive loss* pada data validasi, seperti diperlihatkan pada Gambar 4.7 dan Gambar 4.8. Sementara itu, ringkasan performa akurasi rata-rata disajikan pada Tabel 4.6.



Gambar 4.7 Akurasi Validasi pada Model Siamese VGG-16 (5-Way K-Shot)

Gambar 4.7 menunjukkan bahwa akurasi validasi meningkat seiring bertambahnya nilai K . Model dengan 1-Shot menghasilkan akurasi paling rendah dan fluktuatif, berkisar antara 50% hingga 58%. Hal ini dapat dimaklumi karena hanya tersedia satu contoh per kelas dalam support set, sehingga representasi fitur yang diperoleh masih terbatas dan kurang stabil. Mulai dari 3-Shot hingga 5-Shot, model menunjukkan tren akurasi yang lebih tinggi dan konsisten, dengan kisaran akurasi yang lebih stabil mendekati atau melebihi 60%.

Hal tersebut diperkuat oleh Gambar 4.8 yang menunjukkan tren penurunan *contrastive loss* seiring bertambahnya nilai K . Model dengan 1-Shot dan 2-Shot memiliki nilai loss yang lebih tinggi dan tidak stabil, mencerminkan kesulitan model dalam membentuk ruang embedding yang terstruktur hanya dengan sedikit data. Sebaliknya, konfigurasi 3-Shot hingga 5-Shot memperlihatkan nilai loss yang lebih rendah dan namun belum konvergen, yang menandakan bahwa proses pembelajaran representasi visual berjalan lebih efektif dan efisien.



Gambar 4.8 Contrastive LossValidasi pada Model Siamese VGG-16 (5-Way K-Shot)

Hasil kuantitatif pada Tabel 4.6 berikut mendukung hasil visual tersebut. Rata-rata akurasi validasi tertinggi diperoleh pada konfigurasi 5-Way 5-Shot sebesar 55,98%, diikuti oleh 4-Shot sebesar 54,48%, dan 2-Shot sebesar 53,57%. Akurasi terendah tercatat pada 1-Shot, yakni 50,90%. Namun demikian, semakin besar nilai K , standar error juga cenderung meningkat. Ini menunjukkan adanya variabilitas antar episode yang lebih tinggi, yang mungkin disebabkan oleh peningkatan jumlah kombinasi pasangan dalam support dan query.

Tabel 4.6 Hasil Akurasi Model VGG-16 Sebagai Backbone pada 5-Way K-Shot

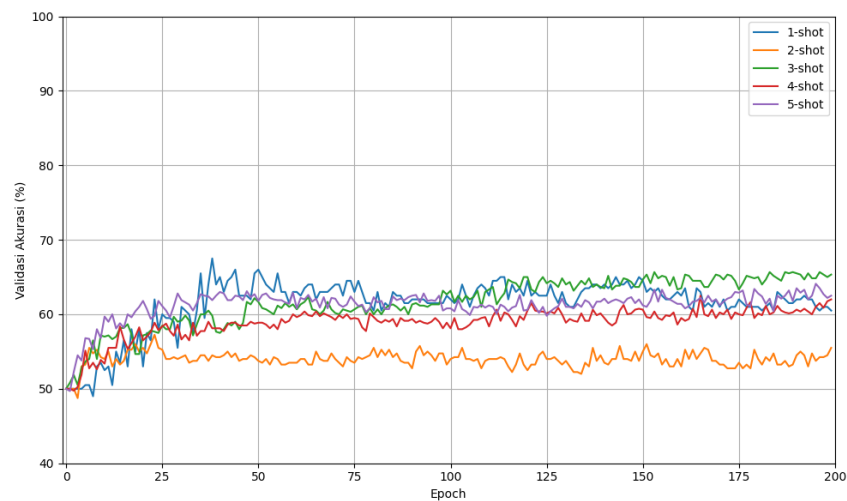
N-Way K-Shot	Akurasi (%)	Standar Error
5-Way 1-Shot	50,90	1,74
5-Way 2-Shot	53,57	5,23
5-Way 3-Shot	51,26	3,46
5-Way 4-Shot	54,48	5,32
5-Way 5-Shot	55,98	6,49

4.4.2 Few-Shot Learning Siamese Network Backbone CNN

Eksperimen pada bagian ini bertujuan untuk mengevaluasi performa model *Siamese Network* dengan backbone CNN dalam skema 5-Way K-Shot. Nilai K divariasikan dari 1 hingga 5, dengan pengukuran performa dilakukan melalui akurasi validasi dan contrastive loss, sebagaimana ditampilkan pada Gambar 4.9 dan Gambar 4.10. Ringkasan hasil numerik akurasi ditunjukkan pada Tabel 4.7.

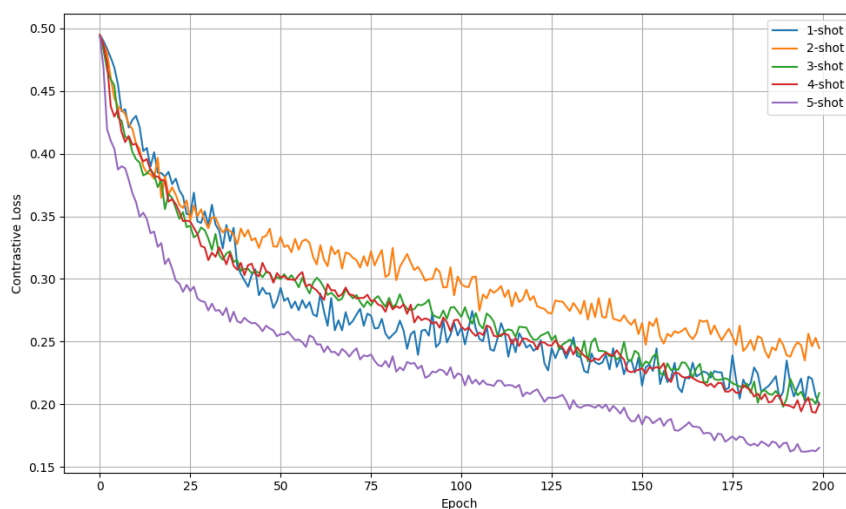
Dari Gambar 4.9, memperlihatkan tren akurasi validasi dari model CNN untuk masing-masing nilai K . Tidak seperti model VGG-16, performa CNN cenderung lebih fluktuatif dan cenderung berada pada kisaran akurasi yang lebih rendah. Akurasi awal model berkisar di bawah 60% dan meningkat seiring bertambahnya nilai K , namun peningkatan tersebut tidak terlalu signifikan. Peningkatan akurasi tampak paling konsisten pada konfigurasi 5-Shot, dengan kisaran stabil antara 60–66%. Sebaliknya, 1-Shot dan 2-Shot menunjukkan fluktuasi yang besar dan akurasi yang lebih rendah. Gambar 4.10 menampilkan tren contrastive loss yang cenderung menurun secara bertahap untuk semua konfigurasi, yang menandakan proses pembelajaran tetap berlangsung. Namun, nilai loss keseluruhan lebih tinggi dibandingkan VGG-16 dan konvergensinya terjadi lebih lambat. Hal ini menunjukkan bahwa representasi

fitur yang dipelajari oleh CNN memerlukan lebih banyak data dan waktu pelatihan untuk mencapai stabilitas yang memadai.



Gambar 4.9 Akurasi Validasi pada Model Siamese CNN (5-Way K-Shot)

Hasil kuantitatif pada Tabel 4.7 memperkuat tren visual yang ditampilkan pada Gambar 4.9 dan Gambar 4.10. Akurasi validasi tertinggi dicapai oleh model pada konfigurasi 5-Way 5-Shot, yaitu sebesar 60,59%, disusul oleh 3-Shot (60,31%) dan 4-Shot (59,68%). Hal ini mengindikasikan bahwa penambahan jumlah contoh (*Shot*) dalam support set berdampak positif terhadap akurasi klasifikasi model CNN. Tidak terjadi peningkatan akurasi yang bersifat monoton seiring naiknya nilai *K*. Misalnya, akurasi pada 3-Shot justru lebih tinggi dibanding 4-Shot. Namun secara umum, semua konfigurasi di atas 2-Shot memberikan performa yang cukup stabil dan lebih baik dibanding 1-Shot (55,25%). Nilai standar error untuk semua konfigurasi relatif kecil, berkisar antara 2,35 hingga 4,09, yang menandakan bahwa model CNN menghasilkan akurasi yang cukup konsisten antar episode. Variabilitas paling tinggi terjadi pada 1-Shot, yang sesuai dengan ekspektasi karena terbatasnya informasi representatif dari satu contoh per kelas. Di sisi lain, nilai standar error cenderung stabil dan rendah pada konfigurasi 2 hingga 5-Shot, menunjukkan peningkatan konsistensi prediksi saat support set semakin lengkap.



Gambar 4. 10 Contrastive Loss Validasi pada Model Siamese CNN (5-Way K-Shot)

Secara keseluruhan, backbone CNN buatan sendiri menunjukkan performa yang kompetitif dan efisien, terutama pada $K \geq 3$. Stabilitas kurva, penurunan loss yang konsisten, serta nilai akurasi tinggi menunjukkan bahwa meskipun dilatih dari awal, arsitektur CNN dapat menghasilkan representasi fitur yang kuat dan efektif untuk pembelajaran kemiripan dalam skenario *few-Shot learning*.

Tabel 4.7 Hasil Akurasi Model CNN Sebagai Backbone pada *5-Way K-Shot*

N-Way K-Shot	Akurasi (%)	Standar Error
5-Way 1-Shot	55,25	4,09
5-Way 2-Shot	56,00	2,35
5-Way 3-Shot	60,31	2,42
5-Way 4-Shot	59,68	2,78
5-Way 5-Shot	60,59	2,72

4.4.3 Model Terbaik

Setelah dilakukan evaluasi performa terhadap kedua arsitektur backbone VGG-16 dan CNN dalam skenario *5-Way K-Shot*, langkah selanjutnya adalah menentukan model terbaik berdasarkan kombinasi metrik akurasi dan contrastive loss pada data validasi. Setiap konfigurasi K (dari 1 hingga 5) akan dibandingkan secara langsung antara kedua backbone untuk melihat mana yang memberikan hasil lebih optimal pada jumlah support tertentu. Analisis ini dilakukan pada data yang berbeda pada Tabel 4.4 dapat dilihat pada Tabel 4.8 untuk mendapatkan analisis yang *fair*, analisis ini dilakukan karena efektivitas backbone tidak hanya bergantung pada kompleksitas model, tetapi juga pada kapasitasnya dalam memahami pola dari jumlah contoh yang terbatas dalam setting *few-Shot learning*. Oleh karena itu, pemilihan model terbaik akan dilakukan pada lima skenario terpisah: *5-Way 1-Shot*, *2-Shot*, *3-Shot*, *4-Shot*, dan *5-Shot*.

Tabel 4.8 Data Testing

sub_category	Label Ke-
Billiard	11
Botol	12
Golf	13
Piringan Hitam	14
Voli	15

Pada konfigurasi *5-Way K-Shot*, masing-masing model backbone diuji untuk mengukur kemampuannya dalam mengidentifikasi kemiripan citra dengan hanya satu contoh per kelas yang tersedia dalam episode pelatihan. Hasil evaluasi uji pada data baru disajikan dalam Tabel 4.8, yang mencakup akurasi prediksi, standar error, dan waktu pelatihan dari model VGG-16 dan CNN.

Hasil evaluasi terhadap data test yang ditampilkan pada Tabel 4.9 memperlihatkan performa model *Siamese Network* dengan backbone VGG-16 dan CNN dalam berbagai konfigurasi *K-Shot*. Secara umum, hasil ini mengonfirmasi pola yang telah diamati pada data validasi, namun juga menunjukkan dinamika baru yang penting untuk dianalisis. Pada konfigurasi *1-Shot*, model VGG-16 kembali menunjukkan performa yang lebih baik dengan akurasi sebesar 50,17%, sedikit mengungguli CNN yang memperoleh 47,70%. Meskipun selisihnya tidak terlalu besar, hasil ini memperkuat keunggulan awal dari model pretrained dalam skenario dengan keterbatasan data ekstrem. Namun, pada konfigurasi *2-Shot* dan *4-Shot*,

CNN mulai menunjukkan performa yang kompetitif. Pada konfigurasi 4-*Shot*, CNN mencatat akurasi tertinggi sebesar 49,90%, sedikit di atas VGG-16 yang berada di angka 49,94%. Sebaliknya, VGG-16 justru kembali mendominasi pada konfigurasi 5-*Shot*, dengan akurasi sebesar 50,09%, unggul tipis dari CNN yang hanya mencapai 49,95%.

Tabel 4.9 Hasil Perbandingan Model Siamese VGG-16 dan Siamese CNN pada K-*Shot*

K- <i>Shot</i>	Model	Akurasi (%)	Standar Error	Waktu Training (Detik (Jam))
1- <i>Shot</i>	VGG-16	50,17	0,02	4170,80 ($\approx 1,16$)
	CNN	47,70	0,02	6036,80 ($\approx 1,68$)
2- <i>Shot</i>	VGG-16	50,37	0,03	8423,30 ($\approx 2,34$)
	CNN	48,84	0,09	12433,00 ($\approx 3,45$)
3- <i>Shot</i>	VGG-16	48,75	0,02	12565,70 ($\approx 3,49$)
	CNN	48,80	0,01	18313,40 ($\approx 5,09$)
4- <i>Shot</i>	VGG-16	49,94	0,02	16654,30 ($\approx 4,63$)
	CNN	49,90	0,02	25592,80 ($\approx 7,11$)
5- <i>Shot</i>	VGG-16	50,09	0,02	20509,20 ($\approx 5,70$)
	CNN	49,95	0,01	29370,60 ($\approx 8,16$)

Dari sisi waktu pelatihan, CNN consistently membutuhkan durasi training yang jauh lebih lama dibandingkan VGG-16 di semua konfigurasi. Perbedaan ini paling mencolok pada konfigurasi 5-*Shot*, di mana waktu pelatihan CNN mencapai 8,16 jam, sedangkan VGG-16 hanya membutuhkan 5,70 jam. Perbedaan durasi ini mengindikasikan bahwa VGG-16 memiliki keunggulan efisiensi yang cukup signifikan, karena bobot awal dari model sudah diinisialisasi melalui proses pretraining. Hal ini sangat menguntungkan terutama dalam penerapan praktis di lingkungan dengan keterbatasan sumber daya komputasi.

Secara umum, meskipun CNN mampu mencapai akurasi yang kompetitif, performanya cenderung lebih fluktuatif dibandingkan VGG-16. Hal ini terlihat dari nilai standar error yang lebih besar pada beberapa konfigurasi, seperti pada 2-*Shot* dan 3-*Shot*. Di sisi lain, VGG-16 menunjukkan performa yang lebih stabil, dengan standar error yang relatif konstan dan kecil di seluruh konfigurasi. Oleh karena itu, meskipun CNN memiliki potensi untuk mencapai akurasi lebih tinggi dalam kondisi tertentu, VGG-16 tetap menjadi model yang lebih andal, stabil, dan efisien untuk implementasi sistem identifikasi produk berbasis citra dengan pendekatan few-*Shot* learning, khususnya di sektor e-commerce.

Tabel 4.10 Hasil Perbandingan Presisi Pada Model Siamese VGG-16 dan Siamese CNN pada K-*Shot*

K- <i>Shot</i>	Model Siamese	Presisi "Sama"	Presisi "Tidak Sama"	Jumlah Pasangan
1- <i>Shot</i>	VGG-16	51,06	51,16	90
	CNN	47,37	49,28	
2- <i>Shot</i>	VGG-16	42,25	44,95	180
	CNN	53,03	51,72	
3- <i>Shot</i>	VGG-16	50,85	50,66	270
	CNN	54,84	52,57	
4- <i>Shot</i>	VGG-16	44,51	45,41	360
	CNN	61,54	52,54	
5- <i>Shot</i>	VGG-16	50,88	50,54	450
	CNN	48,38	49,88	

Tabel 4.10 menyajikan nilai presisi untuk masing-masing kelas, yaitu kelas "sama" (label 0) dan kelas "tidak sama" (label 1), pada model VGG-16 dan CNN dalam berbagai skenario *few-Shot learning* mulai dari 1-*Shot* hingga 5-*Shot*. Evaluasi ini bertujuan untuk menilai ketepatan model dalam memprediksi masing-masing kelas secara terpisah, yang sangat penting dalam konteks identifikasi citra berbasis kemiripan visual. Pada konfigurasi 1-*Shot*, model VGG-16 menunjukkan performa yang seimbang antara dua kelas dengan presisi sebesar 51,06% untuk kelas "sama" dan 51,16% untuk kelas "tidak sama", meskipun hanya memiliki satu contoh per kelas dalam *support set*. Hal ini menunjukkan bahwa fitur yang ditransfer dari pretrained model VGG-16 mampu memberikan representasi awal yang cukup stabil. Sebaliknya, CNN mencatat presisi yang lebih rendah dan tidak seimbang, yaitu 47,37% (kelas sama) dan 49,28% (kelas tidak sama), mengindikasikan bahwa arsitektur yang dilatih dari awal membutuhkan lebih banyak data untuk menghasilkan prediksi yang konsisten.

Ketika jumlah *Shot* meningkat menjadi 2, performa VGG-16 justru menurun signifikan dengan presisi di bawah 45% untuk kedua kelas. Hal ini menunjukkan bahwa penambahan jumlah pasangan citra tidak otomatis meningkatkan ketepatan model jika arsitektur belum cukup adaptif terhadap variasi data yang terbatas. Sebaliknya, CNN menunjukkan peningkatan performa yang signifikan dengan presisi sebesar 53,03% (kelas "sama") dan 51,72% (kelas "tidak sama"), menandakan bahwa CNN mulai berhasil membentuk representasi visual yang lebih diskriminatif dengan tambahan data. Pada konfigurasi 3-*Shot*, VGG-16 kembali menunjukkan kestabilannya dengan presisi di atas 50% untuk kedua kelas, sementara CNN mempertahankan keunggulan dengan presisi tertinggi hingga saat itu, yaitu 54,84% dan 52,57% masing-masing untuk kelas "sama" dan "tidak sama". Keunggulan CNN mencapai puncaknya pada 4-*Shot* dengan presisi tertinggi 61,54% pada kelas "sama", jauh melampaui VGG-16 yang hanya mencapai 44,51%. Hal ini menunjukkan fleksibilitas CNN dalam menyerap informasi baru secara lebih cepat pada kondisi data menengah.

Namun, pada 5-*Shot*, presisi CNN justru mengalami penurunan terutama pada kelas "sama" (48,38%), sedangkan VGG-16 kembali menunjukkan kestabilannya dengan presisi seimbang sekitar 50% pada kedua kelas. Kondisi ini memperkuat temuan bahwa VGG-16 unggul dalam situasi dengan *support set* yang lebih besar karena pretrained weight-nya mampu menjaga konsistensi prediksi antar kelas.

Secara keseluruhan, analisis ini menunjukkan bahwa pemilihan backbone sangat berpengaruh terhadap distribusi prediksi per kelas. VGG-16 unggul dalam stabilitas dan keseimbangan prediksi, sedangkan CNN memiliki keunggulan pada range data menengah (2–4 *Shot*), terutama ketika memprioritaskan ketepatan tinggi pada satu kelas. Oleh karena itu, pemilihan model ideal harus mempertimbangkan konteks penggunaan—baik dari sisi stabilitas prediksi antar kelas, jumlah data latih yang tersedia, maupun kebutuhan efisiensi pelatihan.

Bab ini telah merangkum hasil eksperimen dan analisis terhadap model *Few-Shot Learning* berbasis *Siamese Network* menggunakan dua jenis backbone, yakni VGG-16 dan CNN. Evaluasi dilakukan pada skenario 5-*Way* dengan variasi *K-Shot* dari 1 hingga 5, menggunakan metrik akurasi, contrastive loss, dan presisi per kelas. Hasil eksperimen mengindikasikan bahwa VGG-16 unggul dari sisi stabilitas dan efisiensi, terutama pada konfigurasi 1-*Shot*, 3-*Shot*, dan 5-*Shot*. Keunggulan ini didorong oleh transfer learning dari ImageNet yang memungkinkan pembelajaran cepat dan konsisten.

Sebaliknya, CNN menunjukkan potensi tinggi pada 2-*Shot* hingga 4-*Shot*, dengan presisi lebih tinggi pada kelas "sama". Ini menandakan bahwa meskipun dilatih dari awal, CNN dapat beradaptasi secara fleksibel terhadap penambahan support terbatas, tetapi rentan terhadap overfitting saat support terlalu banyak. Oleh karena itu, pemilihan backbone terbaik sangat tergantung pada konteks implementasi. VGG-16 disarankan untuk sistem yang membutuhkan akurasi seimbang dan waktu pelatihan cepat, sedangkan CNN cocok untuk kondisi low-data yang memerlukan efisiensi memori dan performa tinggi pada satu kategori prediksi dominan.

Selain itu, dapat dilihat bahwa akurasi yang dihasilkan oleh kedua model memiliki tidak memiliki perbedaan, dapat dilihat dari standar deviasi yang dihasilkan yang dapat disimpulkan bahwa 1 sampai 5 *Shot* tidak memiliki efek pada model. Hal ini dikarenakan, model belum melakukan progress yang baik, dapat dilihat pada *epoch* yang dilakukan pada penelitian ini kurang banyak. Terlihat bahwa nilai standar error pada beberapa konfigurasi K-*Shot* masih tergolong tinggi, khususnya pada model dengan backbone VGG-16. Hal ini mengindikasikan bahwa model belum menunjukkan performa yang stabil dan konsisten. Sebagai contoh, standar error pada konfigurasi 5-Way 5-*Shot* untuk VGG-16 mencapai 6,49, yang menunjukkan adanya variabilitas hasil prediksi yang cukup besar antar episode pelatihan. Tingginya nilai standar error ini mengindikasikan bahwa model belum mengalami konvergensi yang memadai, yang kemungkinan besar disebabkan oleh jumlah epoch pelatihan yang masih terbatas. Dengan kata lain, model belum mengalami progres yang signifikan dalam memahami pola representasi antar gambar, sehingga penambahan jumlah support (*shot*) belum memberikan peningkatan performa yang nyata. Sementara itu, meskipun model CNN menunjukkan standar error yang lebih rendah dan cenderung stabil, tren akurasi antar K-*Shot* juga masih relatif datar. Hal ini memperkuat dugaan bahwa kedua model belum menjalani proses pelatihan yang cukup lama untuk mencapai generalisasi yang optimal. Oleh karena itu, diperlukan pelatihan tambahan dengan jumlah epoch yang lebih besar agar model dapat menunjukkan peningkatan akurasi yang signifikan dan konsistensi prediksi yang lebih baik antar episode.

(Halaman ini sengaja dikosongkan)

BAB 5 PENUTUP

5.1 KESIMPULAN

Penelitian ini bertujuan untuk mengevaluasi efektivitas metode *Few-Shot Learning* menggunakan arsitektur *Siamese Network* dalam tugas klasifikasi kemiripan gambar produk pada platform e-commerce. Pendekatan ini diuji dalam skenario *5-Way K-Shot* ($K = 1$ hingga 5) dengan membandingkan dua jenis backbone: VGG-16 (pretrained dari ImageNet) dan CNN (dibangun dari awal). Dataset yang digunakan diambil dari platform Kaggle dengan berbagai kategori produk yang umum ditemukan di aplikasi e-commerce seperti Shopee.

Dari hasil eksperimen yang telah dibahas, dapat disimpulkan beberapa poin utama sebagai berikut:

1. Stabilitas dan Konsistensi VGG-16: Backbone VGG-16 menunjukkan performa paling stabil dan konsisten di seluruh konfigurasi *K-Shot*. Akurasi validasi tertinggi dicapai pada konfigurasi *5-Way 5-Shot* dengan nilai 55,98%, disertai presisi per kelas yang seimbang dan error yang relatif rendah. Keunggulan ini berasal dari pemanfaatan fitur pretrained yang membantu model belajar representasi visual lebih efisien pada kondisi data terbatas.
2. Keunggulan CNN pada *Shot* Menengah: CNN menunjukkan performa superior pada konfigurasi *Shot* menengah (2–4 *Shot*), dengan presisi tertinggi pada kelas “sama” mencapai 61,54% pada konfigurasi *4-Shot*. Namun, performa tersebut cenderung fluktuatif dan menurun pada konfigurasi *5-Shot*, diduga akibat overfitting terhadap kombinasi pasangan support-query yang kompleks.
3. Distribusi Presisi Antar Kelas: Evaluasi presisi per kelas menunjukkan bahwa VGG-16 memiliki distribusi prediksi yang lebih seimbang antara kelas “sama” dan “tidak sama”, sedangkan CNN cenderung menunjukkan presisi tinggi pada salah satu kelas, namun kurang stabil secara keseluruhan.
4. Efisiensi Komputasi: Dari sisi efisiensi pelatihan, VGG-16 membutuhkan waktu pelatihan yang lebih singkat dibandingkan CNN di seluruh konfigurasi. Perbedaan paling mencolok terjadi pada konfigurasi *5-Shot*, di mana CNN membutuhkan hingga 8,16 jam, sedangkan VGG-16 hanya memerlukan 5,70 jam.
5. Rekomendasi Model Terbaik: Dengan mempertimbangkan aspek akurasi, kestabilan prediksi antar kelas, serta efisiensi waktu pelatihan, model terbaik dalam penelitian ini adalah *Siamese Network* dengan backbone VGG-16 pada konfigurasi *5-Way 5-Shot*. Model ini dinilai paling andal untuk implementasi sistem identifikasi produk berbasis gambar pada lingkungan e-commerce yang menghadapi keterbatasan data latih.

5.2 SARAN

Berdasarkan hasil dan keterbatasan penelitian ini, beberapa saran yang dapat diberikan untuk pengembangan ke depan adalah sebagai berikut:

1. Menambahkan backbone lain yang lebih ringan dan modern seperti ResNet, MobileNet, atau EfficientNet agar dapat dibandingkan dari segi efisiensi dan akurasi pada perangkat dengan sumber daya terbatas.
2. Melakukan evaluasi tambahan dengan metrik seperti F1-score, recall, dan ROC-AUC, terutama pada dataset yang tidak seimbang atau mengandung ambiguitas visual tinggi, agar kualitas klasifikasi lebih komprehensif.

3. Menguji ketahanan model terhadap citra berkualitas rendah atau noise tinggi, seperti blur, pencahayaan buruk, atau sudut pandang ekstrem, untuk mensimulasikan kondisi dunia nyata yang lebih realistis.
4. Mengimplementasikan metode fine-tuning bertahap (progressive learning) untuk backbone pretrained, agar model dapat menyesuaikan fitur global dengan domain lokal produk e-commerce secara lebih fleksibel.
5. Mengintegrasikan sistem ini ke dalam pipeline pencarian visual aktual di e-commerce, misalnya untuk deteksi duplikat produk, validasi retur, atau rekomendasi produk serupa, guna mengevaluasi performa dalam lingkungan produksi nyata.
6. Mengembangkan metode *prototype-based few-Shot* classification (seperti Prototypical Networks atau Relation Networks) sebagai pembandingan, agar diperoleh gambaran lebih luas mengenai pendekatan optimal untuk klasifikasi gambar dengan data terbatas.

DAFTAR PUSTAKA

- Agustina, Diah, 2017, “Fitur Social Commerce Dalam Website E-Commerce Di Indonesia,” *Informatika Mulawarman : Jurnal Ilmiah Ilmu Komputer* 12 (1): 25–29, [Https://E-Journals,Unmul,Ac,Id/Index,Php/JIM/Article/View/219](https://E-Journals,Unmul,Ac,Id/Index,Php/JIM/Article/View/219),
- Akhtar, Malik Javed, Rabbia Mahum, Faisal Shafique Butt, Rashid Amin, Ahmed M, El-Sherbeeney, Seongkwan Mark Lee, And Sarang Shaikh, 2022, “A Robust Framework For Object Detection In A Traffic Surveillance System,” *Electronics (Switzerland)* 11 (21), [Https://Doi,Org/10,3390/ELECTRONICS11213425](https://doi.org/10.3390/ELECTRONICS11213425),
- Alqasir, Hiba, Damien Muselet, And Christophe Ducottet, 2024, “Introducing Shape Priors In Siamese Networks For Image Classification,” *Neurocomputing* 568, [Https://Doi,Org/10,1016/J,Neucom,2023,127034](https://doi.org/10.1016/j.neucom.2023.127034),
- Annur, C, 2022, “Google Prediksi E-Commerce Indonesia Terus Menguat Sampai 2025,” 2022, [Https://Databoks,Katadata,Co,Id/Ekonomi-Makro/Statistik/3f8eb7de9656bc5/Google-Prediksi-E-Commerce-Indonesia-Terus-Menguat-Sampai-2025](https://databoks.katadata.co.id/ekonomi-makro/statistik/3f8eb7de9656bc5/google-prediksi-e-commerce-indonesia-terus-menguat-sampai-2025),
- Argüeso, David, Artzai Picon, Unai Irusta, Alfonso Medela, Miguel G, San-Emeterio, Arantza Bereciartua, And Aitor Alvarez-Gila, 2020, “Few-Shot Learning Approach For Plant Disease Classification Using Images Taken In The Field,” *Computers And Electronics In Agriculture* 175 (August), [Https://Doi,Org/10,1016/J,Compag,2020,105542](https://doi.org/10.1016/j.compag.2020.105542),
- Ariyadi, M, Rama Nopan, Muhammad Rizky Pribadi, And Eka Puji Widiyanto, 2023, “Unmanned Aerial Vehicle For Remote Sensing Detection Of Oil Palm Trees Using You Only Look Once And Convolutional Neural Network,” *International Conference On Electrical Engineering, Computer Science And Informatics (EECSI)*, 226–30, [Https://Doi,Org/10,1109/EECSI59885,2023,10295670](https://doi.org/10.1109/EECSI59885.2023.10295670),
- Azmi, Khairul, Sarjon Defit, And Universitas Putra Indonesia YPTK Padang Jl Raya Lubuk Begalung-Padang-Sumatera Barat, 2023, “Implementasi Convolutional Neural Network (CNN) Untuk Klasifikasi Batik Tanah Liat Sumatera Barat” 16 (1): 2023,
- Basak, Hritam, Rohit Kundu, Sukanta Chakraborty, And Nibaran Das, 2021, “Cervical Cytology Classification Using PCA And GWO Enhanced Deep Features Selection,” *SN Computer Science* 2 (5), [Https://Doi,Org/10,1007/S42979-021-00741-2](https://doi.org/10.1007/S42979-021-00741-2),
- Boyle, Jonathan, Phd Thesis, James Ferryman, And Hong Wei, 2022, “Siamese Networks For Surveillance And Security,”
- Devi, Citra, 2016, “Mesin Pencari Berbasis Gambar Untuk Mendeteksi Produk Smartphone Menggunakan Fitur Morfologi Gambar” 6 (2): 96–102,
- Edbert, I, 2020, “POOLING LAYER – School Of Computer Science,” 2020, [Https://Socs,Binus,Ac,Id/2021/10/07/Pooling-Layer/](https://socs.binus.ac.id/2021/10/07/Pooling-Layer/),
- Faroz, Salman, 2022, “One Shot Learning (N Way K Shot): Siamese Network With Contrastive Loss For Pokémon Classification | By Salman Faroz | Becoming Human: Artificial

- Intelligence Magazine,” September 19, 2022, <https://Becominghuman,Ai/N-Way-K-Shot-Siamese-Network-With-Contrastive-Loss-For-Pokemon-Classification-B1bc6c2ab766>,
- FATURRAHMAN, RAIHAN, YULI SUN HARIYANI, And SUGONDO HADIYOSO, 2023, “Klasifikasi Jajanan Tradisional Indonesia Berbasis Deep Learning Dan Metode Transfer Learning,” *ELKOMIKA: Jurnal Teknik Energi Elektrik, Teknik Telekomunikasi, & Teknik Elektronika* 11 (4): 945, <https://doi.org/10.26760/Elkomika,V11i4,945>,
- Fedele, Andrea, Riccardo Guidotti, And Dino Pedreschi, 2024, “Explaining *Siamese Networks* In Few-Shot Learning,” *Machine Learning*, October, <https://doi.org/10.1007/S10994-024-06529-8>,
- Firmansyah, Ichsan, And Dan B Herawan Hayadi, 2022, “Komparasi Fungsi Aktivasi Relu Dan Tanh Pada Multilayer Perceptron,” *JIKO (Jurnal Informatika Dan Komputer)* 6 (2): 200–206, <https://doi.org/10.26798/JIKO,V6i2,600>,
- Gao, Hongmin, Junpeng Zhang, Xueying Cao, Zhonghao Chen, Yiyan Zhang, And Chenming Li, 2021, “Dynamic Data Augmentation Method For Hyperspectral Image Classification Based On Siamese Structure,” *IEEE Journal Of Selected Topics In Applied Earth Observations And Remote Sensing* 14:8063–76, <https://doi.org/10.1109/JSTARS.2021.3102610>,
- Guntara, Rangga Gelar, 2023, “Pemanfaatan Komputer Vision Pada E-Commerce,” *Rangga Gelar Guntara) Madani: Jurnal Ilmiah Multidisiplin* 1 (3): 2302–6219, <https://doi.org/10.5281/Zenodo.7881002>,
- Haile, Melaku Bitew, Yelkal Mulualem Walle, And Abebech Jenber Belay, 2024, “Enhanced Image-Based Malware Multiclass Classification Method With The Ensemble Model And SVM,” *Open Information Science* 8 (1), <https://doi.org/10.1515/OPIS-2024-0003>,
- Heidari, Mohsen, And Kazim Fouladi-Ghaleh, 2020, “Using *Siamese Networks* With Transfer Learning For Face Recognition On Small-Samples Datasets,” *Iranian Conference On Machine Vision And Image Processing, MVIP 2020-February* (February), <https://doi.org/10.1109/MVIP49855.2020.9116915>,
- Howard, Andrew G., Menglong Zhu, Bo Chen, Dmitry Kalenichenko, Weijun Wang, Tobias Weyand, Marco Andreetto, And Hartwig Adam, 2017, “Mobilenets: Efficient Convolutional Neural Networks For Mobile Vision Applications,” April, <http://arxiv.org/abs/1704.04861>,
- Isabona, Joseph, Agbotiname Lucky Imoize, Stephen Ojo, Olukayode Karunwi, Yongsung Kim, Cheng Chi Lee, And Chun Ta Li, 2022, “Development Of A Multilayer Perceptron Neural Network For Optimal Predictive Modeling In Urban Microcellular Radio Environments,” *Applied Sciences* 2022, Vol, 12, Page 5713 12 (11): 5713, <https://doi.org/10.3390/AP12115713>,
- James, Gareth, Daniela Witten, Trevor Hastie, Robert Tibshirani, And Jonathan Taylor, 2023, “An Introduction To Statistical Learning,” Springer Texts In Statistics, , <https://doi.org/10.1007/978-3-031-38747-0>,
- Kingma, Diederik P, And Jimmy Lei Ba, 2015, “ADAM: A METHOD FOR STOCHASTIC OPTIMIZATION,”

- Lakshmi, Moturiyamini, Murikipudi Swethasupriya, Jonnala Nandini, And S Sai, 2020, “CNN Based Image Identification With Python,” *International Advanced Research Journal In Science, Engineering And Technology* 7 (2), <https://doi.org/10.17148/IARJSET,2020,7212>,
- Leni, Wandira, 2024, “Shopee Catatkan Penjualan Online Naik Hingga 200% Di 2024,” 2024, <https://pressrelease.kontan.co.id/news/shopee-catatkan-penjualan-online-naik-hingga-200-di-2024>,
- Li, Yuhong, Xiaofan Zhang, And Deming Chen, 2019, “Siamvgg: Visual Tracking Using Deeper Siamese Networks,” February, <http://arxiv.org/abs/1902.02804>,
- Mao, Anqi, Mehryar Mohri, And Yutao Zhong, 2023, “Cross-Entropy Loss Functions: Theoretical Analysis And Applications,”
- Markoulidakis, Ioannis, Ioannis Rallis, Ioannis Georgoulas, George Kopsiaftis, Anastasios Doulamis, And Nikolaos Doulamis, 2021, “Multiclass Confusion Matrix Reduction Method And Its Application On Net Promoter Score Classification Problem,” *Technologies* 9 (4), <https://doi.org/10.3390/TECHNOLOGIES9040081>,
- Ma’Rufah, Laila, Tita Karlita, Umi Sa’Adah, And Willy Achmat Fauzi, 2023, “A Novel Approach To Visual Search In E-Commerce Fashion Using Siamese Neural Network And Multi-Scale CNN,” In *IES 2023 - International Electronics Symposium: Unlocking The Potential Of Immersive Technology To Live A Better Life, Proceeding*, <https://doi.org/10.1109/IES59143.2023.10242507>,
- Masnadi-Shirazi, Hamed, And Nuno Vasconcelos, N,D, “On The Design Of Loss Functions For Classification: Theory, Robustness To Outliers, And Savageboost,” Accessed January 24, 2025,
- Maucher, J, 2014, “Machine Learning, Deep Learning And The Data Problem — Machine Learning Lecture,” October 22, 2014, <https://maucher.pages.mi.hdm-stuttgart.de/mlbook/00mlanddl.html>,
- Mishra, Ranjan Kumar, G, Y,Sandesh Reddy, And Himanshu Pathak, 2021, “The Understanding Of Deep Learning: A Comprehensive Review,” *Mathematical Problems In Engineering* 2021, <https://doi.org/10.1155/2021/5548884>,
- Mulindwa, Desire Burume, And Shengzhi Du, 2023, “An N-Sigmoid Activation Function To Improve The Squeeze-And-Excitation For 2D And 3D Deep Networks,” *Electronics* 2023, Vol, 12, Page 911 12 (4): 911, <https://doi.org/10.3390/ELECTRONICS12040911>,
- Nasteski, Vladimir, 2017, “An Overview Of The Supervised Machine Learning Methods,” *HORIZONS,B* 4 (December):51–62, <https://doi.org/10.20544/HORIZONS,B,04,1,17,P05>,
- Nurhalizah, Ria Suci, Rian Ardianto, And Purwono Purwono, 2024, “Analisis Supervised Dan Unsupervised Learning Pada Machine Learning: Systematic Literature Review,” *Jurnal Ilmu Komputer Dan Informatika* 4 (1): 61–72, <https://doi.org/10.54082/JIKI,168>,
- Ouali, Yassine, Céline Hudelot, And Myriam Tami, 2020, “An Overview Of Deep Semi-Supervised Learning,” June, <http://arxiv.org/abs/2006.05278>,

- Panggabean, Trisatin, Salsabila Yusra, Sri Ratna Dewi, Jl V William Iskandar Ps, Medan Estate, Kec Percut Sei Tuan, And Kabupaten Deli Serdang, 2025, “Penerapan Komputer Dalam Identifikasi Barang Di E-Commerce Berbasis AI,” 107–14, <https://doi.org/10.59581/jusiik-widyakarya.v3i1.4455>,
- Parnami, Archit, And Minwoo Lee, 2022, “Learning From Few Examples: A Summary Of Approaches To Few-Shot Learning,” March, <http://arxiv.org/abs/2203.04291>,
- Pedregosa FABIANPEDREGOSA, Fabian, Vincent Michel, Olivier Grisel OLIVIERGRISEL, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Jake Vanderplas, Et Al, 2011, “Scikit-Learn: Machine Learning In Python Gaël Varoquaux Bertrand Thirion Vincent Dubourg Alexandre Passos PEDREGOSA, VAROQUAUX, GRAMFORT ET AL, Matthieu Perrot,” *Journal Of Machine Learning Research* 12:2825–30, <http://scikit-learn.sourceforge.net>,
- Priya, Bala, 2023, “Softmax Activation Function: Everything You Need To Know | Pinecone,” June 30, 2023, <https://www.pinecone.io/learn/softmax-activation/>,
- Puji Widodo, Aris, Adi Wibowo, And Kabul Kurniawan, 2023, “Enhancing Software User Interface Testing Through Few Shot Deep Learning: A Novel Approach For Automated Accuracy And Usability Evaluation,” *IJACSA International Journal Of Advanced Computer Science And Applications*, Vol, 14, www.ijacsa.thesai.org,
- Retnoningsih, Endang, And Rully Pramudita, 2020, “Mengenal Machine Learning Dengan Teknik Supervised Dan Unsupervised Learning Menggunakan Python,” *BINA INSANI ICT JOURNAL* 7 (2): 156–65, <https://www.python.org/>,
- Rosebrock, A, 2021, “Contrastive Loss For Siamese Networks With Keras And Tensorflow - Pyimagesearch,” January 18, 2021, <https://pyimagesearch.com/2021/01/18/contrastive-loss-for-siamese-networks-with-keras-and-tensorflow/>,
- Santoso, Puput, Heri Abijono, And Novita Lestari Anggreini, 2021, “ALGORITMA SUPERVISED LEARNING DAN UNSUPERVISED LEARNING DALAM PENGOLAHAN DATA,” *Unira Malang* | 4 (2),
- Sarker, Iqbal H, 2021, “Deep Learning: A Comprehensive Overview On Techniques, Taxonomy, Applications And Research Directions,” *SN Computer Science* 2 (6): 1–20, <https://doi.org/10.1007/s42979-021-00815-1/figures/11>,
- Sharma, Siddharth, Simone Sharma, And Anidhya Athaiya, 2020, “ACTIVATION FUNCTIONS IN NEURAL NETWORKS,” *International Journal Of Engineering Applied Sciences And Technology* 4:310–16, <http://www.ijeast.com>,
- Shim, Jae Wan, 2024, “Enhancing Cross Entropy With A Linearly Adaptive Loss Function For Optimized Classification Performance,” *Scientific Reports* 2024 14:1 14 (1): 1–6, <https://doi.org/10.1038/s41598-024-78858-6>,
- “Shopee - Price Match Guarantee | Kaggle,” 2024, 2024, <https://www.kaggle.com/competitions/shopee-product-matching>,
- Simonyan, Karen, And Andrew Zisserman, 2014, “Very Deep Convolutional Networks For Large-Scale Image Recognition,” *3rd International Conference On Learning*

- Representations, ICLR 2015 - Conference Track Proceedings*, September, <https://arxiv.org/abs/1409.1556v6>,
- Smetanin, Sergey, And Mikhail Komarov, 2019, “Sentiment Analysis Of Product Reviews In Russian Using Convolutional Neural Networks,” *Conference On Business Informatics* 1 (July):482–86, <https://doi.org/10.1109/CBI.2019.00062>,
- Staudemeyer, R, 2019, “(PDF) Understanding LSTM -- A Tutorial Into Long Short-Term Memory Recurrent Neural Networks,” September 2019, https://www.researchgate.net/publication/335975993_Understanding_LSTM_-_A_Tutorial_Into_Long_Short-Term_Memory_Recurrent_Neural_Networks,
- Suyahman, Sunardi, And Murinto, 2024, “Comparative Analysis Of CNN Architectures In Siamese Networks With Test-Time Augmentation For Trademark Image Similarity Detection,” *Scientific Journal Of Informatics* 11 (4): 949–58, <https://doi.org/10.15294/Sji.V11i4.13811>,
- Togashi, Riku, Hideyuki Maeda, Vibhor Kanojia, Kousuke Morimoto, And Sumio Fujita, 2017, “Euclidean Image Embedding In View Of Similarity Ranking In Auction Search By Image,” In *26th International World Wide Web Conference 2017, WWW 2017 Companion*, <https://doi.org/10.1145/3041021.3054262>,
- Unzueta, D, 2022, “Fully Connected Layer Vs Convolutional Layer: Explained | Built In,” October 19, 2022, <https://builtin.com/machine-learning/fully-connected-layer>,
- Verma, A, 2024, “Siamese Neural Network In Deep Learning,” September 16, 2024, <https://www.pickl.ai/blog/siamese-neural-network-in-deep-learning-features-and-architecture/>,
- Villalba-Diez, Javier, Daniel Schmidt, Roman Gevers, Joaquín Ordieres-Meré, Martin Buchwitz, And Wanja Wellbrock, 2019, “Deep Learning For Industrial Computer Vision Quality Control In The Printing Industry 4.0,” *Sensors* 2019, Vol, 19, Page 3987 19 (18): 3987, <https://doi.org/10.3390/S19183987>,
- Xue, Hui, Yuexuan An, Yongchun Qin, Wenqian Li, Yixin Wu, Yongjuan Che, Pengfei Fang, And Minling Zhang, 2024, “Towards Few-Shot Learning In The Open World: A Review And Beyond,” August, <http://arxiv.org/abs/2408.09722>,
- Yu, Yaodong, Maziar Sanjabi, Yi Ma, Kamalika Chaudhuri, And Chuan Guo, 2024, “Vip: A Differentially Private Foundation Model For Computer Vision,” <https://github.com/>,
- Zhou, Xiaokang, Wei Liang, Shohei Shimizu, Jianhua Ma, And Qun Jin, 2021, “Siamese Neural Network Based Few-Shot Learning For Anomaly Detection In Industrial Cyber-Physical Systems,” *IEEE Transactions On Industrial Informatics* 17 (8): 5790–98, <https://doi.org/10.1109/TII.2020.3047675>,

(Halaman ini sengaja dikosongkan)

LAMPIRAN

Lampiran 1 Arsitektur VGG-16

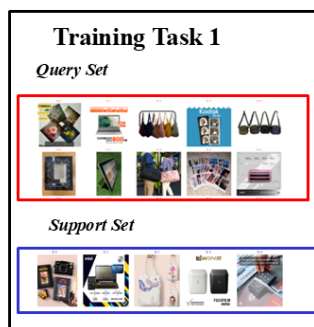
No.	Nama Layer (Type)	Output Shape	Jumlah Parameter
1	Input Layer	(256, 256, 3)	0
2	Block1_Conv1 (Conv2D)	(256, 256, 64)	1,792
3	Block1_Conv2 (Conv2D)	(256, 256, 64)	36,928
4	Block1_Pool (MaxPooling2D)	(128, 128, 64)	0
5	Block2_Conv1 (Conv2D)	(128, 128, 128)	73,856
6	Block2_Conv2 (Conv2D)	(128, 128, 128)	147,584
7	Block2_Pool (MaxPooling2D)	(64, 64, 128)	0
8	Block3_Conv1 (Conv2D)	(64, 64, 256)	295,168
9	Block3_Conv2 (Conv2D)	(64, 64, 256)	590,080
10	Block3_Conv3 (Conv2D)	(64, 64, 256)	590,080
11	Block3_Pool (MaxPooling2D)	(32, 32, 256)	0
12	Block4_Conv1 (Conv2D)	(32, 32, 512)	1,180,160
13	Block4_Conv2 (Conv2D)	(32, 32, 512)	2,359,808
14	Block4_Conv3 (Conv2D)	(32, 32, 512)	2,359,808
15	Block4_Pool (MaxPooling2D)	(16, 16, 512)	0
16	Block5_Conv1 (Conv2D)	(16, 16, 512)	2,359,808
17	Block5_Conv2 (Conv2D)	(16, 16, 512)	2,359,808
18	Block5_Conv3 (Conv2D)	(16, 16, 512)	2,359,808
19	Block5_Pool (MaxPooling2D)	(8, 8, 512)	0
20	GlobalAveragePooling2D	(512)	0
21	Dropout	(512)	0
22	Dense	(256)	131,328
23	BatchNormalization	(256)	1,024
24	Dropout	(256)	0
25	Dense	(128)	32,896
Total Parameter			14,879,936

No.	Nama Layer (Type)	Output Shape	Jumlah Parameter
	Trainable Parameters		13,143,936
	Non-trainable Parameters		1,736,000

Lampiran 2 Ilustrasi Setiap Task dari 1-Shot sampai 5-Shot



Lampiran 3 Ilustrasi 2 Shot



NO	Keterangan	Pasangan Kiri (img1/input 1)	Pasangan Kanan (img2/input2)	Label Baru
1	Gambar			"Sama"
	Label	Kamera	Kamera	
2	Gambar			"Tidak sama"
	Label	Kamera	Laptop	
3	Gambar			"Sama"
	Label	Kamera	Kamera	
4	Gambar			"Tidak Sama"
	Label	Tas	Buku	
5	Gambar			"Sama"
	Label	Tas	Photo Card	
6	Gambar			"Tidak Sama"
	Label	Tas	Parfum	
7	Gambar			"Sama"
	Label	Tas	Tas	
8	Gambar			"Tidak Sama"
	Label	Tas	Buku	
9	Gambar			"Sama"
	Label	Laptop	Buku	
10	Gambar			"Tidak sama"
	Label	Laptop	Laptop	

Lampiran 4 Lanjutan Ilustrasi 2 Shot

Training Task 1

Query Set



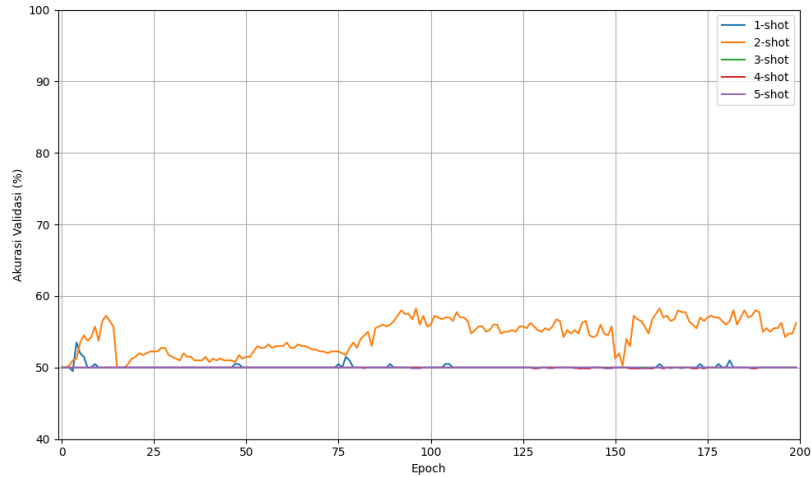
Support Set



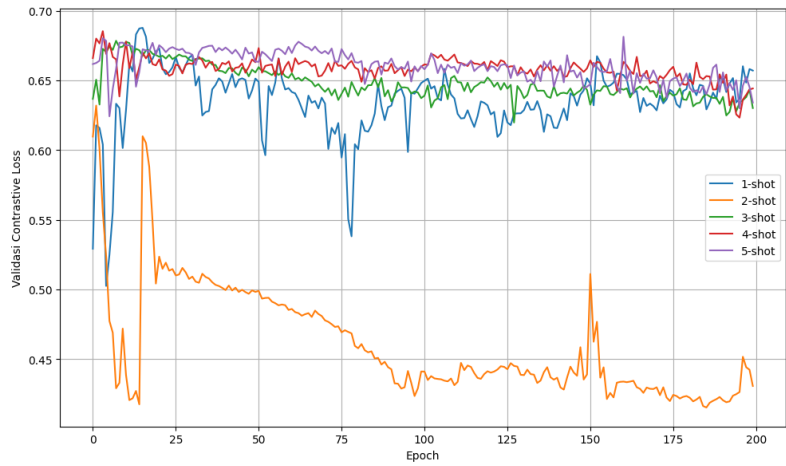
→

NO	Keterangan	Pasangan Kiri (img1/input 1)	Pasangan Kanan (img2/input2)	Label Baru
	Label	Laptop	Laptop	
11	Gambar			"Sama"
	Label	Laptop	Laptop	
12	Gambar			"Tidak Sama"
	Label	Laptop	Buku	
13	Gambar			"Sama"
	Label	Photo Card	Photo Card	
14	Gambar			"Tidak Sama"
	Label	Photo Card	Parfum	
15	Gambar			"Sama"
	Label	Photo Card	Tas	
16	Gambar			"Tidak Sama"
	Label	Photo Card	Buku	
17	Gambar			"Sama"
	Label	Tote Bag	Buku	
18	Gambar			"Tidak sama"
	Label	Tote Bag	Laptop	
19	Gambar			"Sama"
	Label	Tote Bag	Laptop	
20	Gambar			"Tidak Sama"
	Label	Tote Bag	Buku	

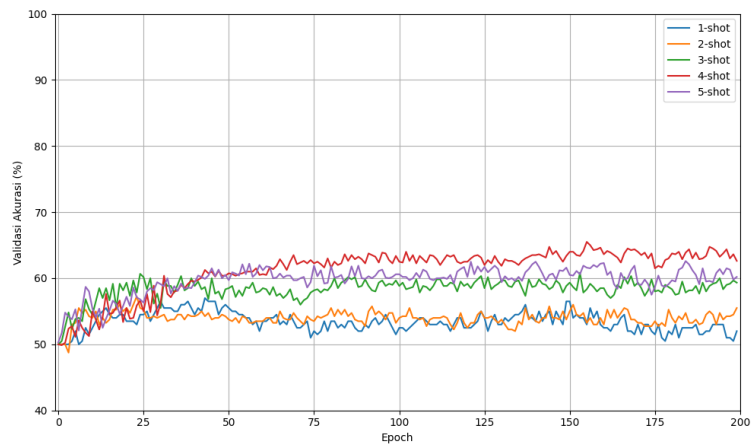
Lampiran 5 Akurasi Validasi pada Model Siamese VGG-16 (5-Way K-Shot) Replikasi Ke-1



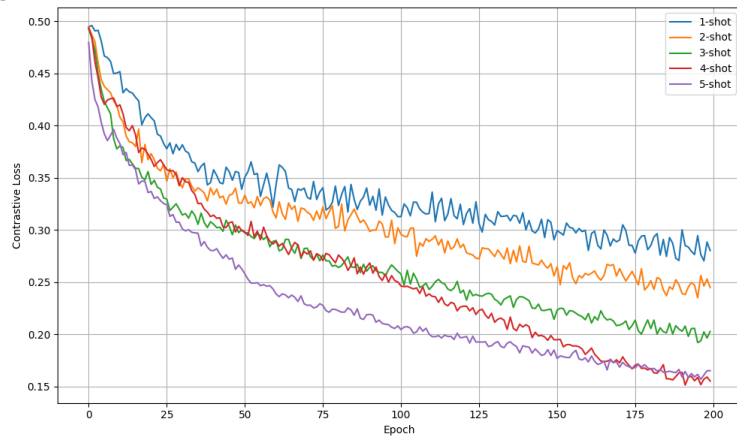
Lampiran 6 Contrastive Loss Validasi pada Model Siamese VGG-16 (5-Way K-Shot) Replikasi Ke-1



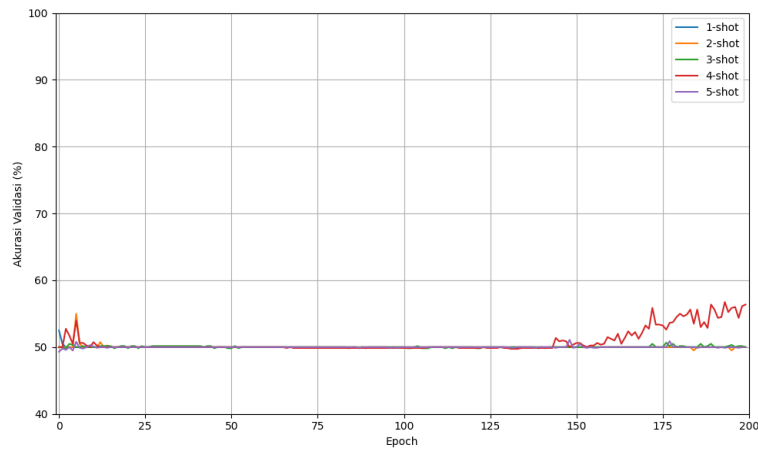
Lampiran 7 Akurasi Validasi pada Model Siamese CNN (5-Way K-Shot) Replikasi Ke-1



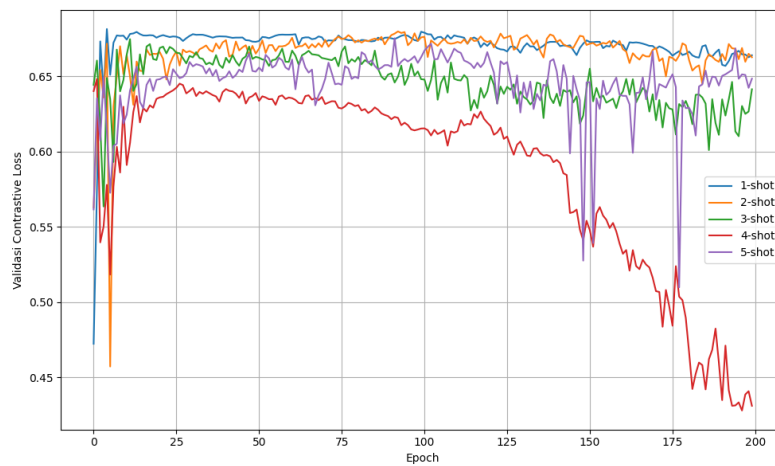
Lampiran 8 Contrastive Loss Validasi pada Model Siamese CNN (5-Way K-Shot) Replikasi Ke-1



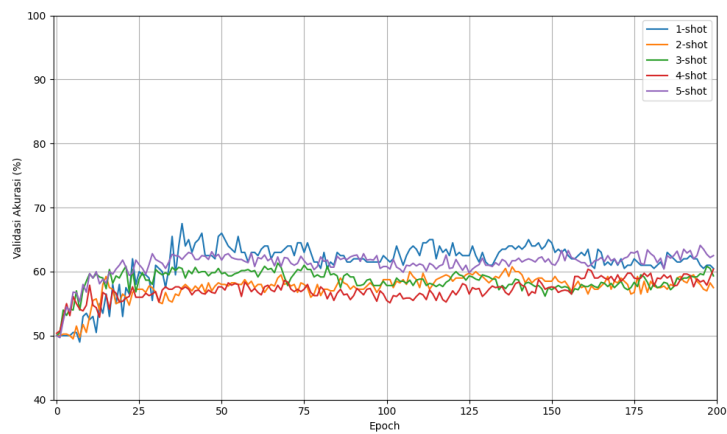
Lampiran 9 Akurasi Validasi pada Model Siamese VGG-16 (5-Way K-Shot) Replikasi Ke-10



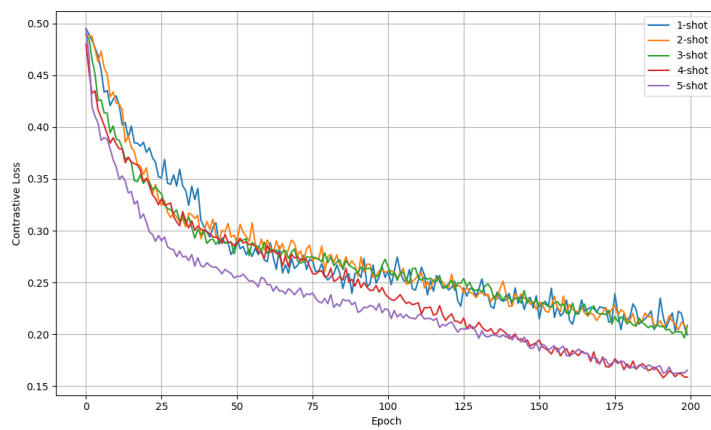
Lampiran 10 Contrastive Loss Validasi pada Model Siamese VGG-16 (*5-Way K-Shot*) Replikasi Ke-10



Lampiran 11 Akurasi Validasi pada Model Siamese CNN (*5-Way K-Shot*) Replikasi Ke-10



Lampiran 12 Contrastive Loss Validasi pada Model Siamese CNN (*5-Way K-Shot*) Replikasi Ke-10



Lampiran 13 Akurasi Validasi pada Model VGG-16

Replikasi Ke-	Akurasi				
	<i>5 Way 1 Shot</i>	<i>5 Way 2 Shot</i>	<i>5 Way 3 Shot</i>	<i>5 Way 4 Shot</i>	<i>5 Way 5 Shot</i>
1	50	50	50	50	50
2	50	58	50	55.5	51.4
3	51	50	61	50	63.9
4	50	50	50	50	50.2
5	50.5	50	51.67	50	62.5
6	52	62.25	50.17	50.13	66.1
7	50	62	50	58.38	57.6
8	55.5	50.25	50	65	58.3
9	50	50	49.83	59.5	49.8
10	50	56.25	50	56.38	50

Lampiran 14 Waktu Training Validasi pada Model VGG-16

Replikasi Ke-	Waktu				
	<i>5 Way 1 Shot</i>	<i>5 Way 2 Shot</i>	<i>5 Way 3 Shot</i>	<i>5 Way 4 Shot</i>	<i>5 Way 5 Shot</i>
1	1h 8m 24s	2h 13m 59s	3h 11m 48s	4h 24m 57s	5h 43m 0s
2	1h 10m 24s	2h 25m 15s	3h 37m 57s	4h 32m 39s	5h 46m 38s
3	1h 13m 53s	2h 24m 8s	3h 37m 41s	4h 30m 40s	5h 30m 29s
4	1h 4m 55s	2h 31m 17s	3h 18m 40s	4h 51m 12s	5h 41m 35s
5	1h 6m 17s	2h 25m 21s	3h 13m 11s	4h 46m 20s	5h 44m 47s
6	1h 9m 30s	2h 23m 35s	3h 48m 39s	4h 32m 6s	5h 49m 15s
7	1h 6m 40s	2h 20m 42s	3h 36m 54s	4h 32m 19s	5h 12m 20s
8	1h 13m 23s	2h 13m 15s	3h 50m 25s	4h 50m 51s	6h 6m 33s
9	1h 8m 21s	2h 20m 13s	3h 20m 1s	4h 34m 5s	5h 45m 53s
10	1h 13m 21s	2h 6m 8s	3h 19m 1s	4h 40m 34s	5h 37m 42s

Lampiran 15 Akurasi Validasi pada Model CNN

REPLIKASI KE-	Akurasi				
	<i>5 Way 1 Shot</i>	<i>5 Way 2 Shot</i>	<i>5 Way 3 Shot</i>	<i>5 Way 4 Shot</i>	<i>5 Way 5 Shot</i>
1	52	57.5	59.33	62.63	60.2
2	53	58	60.17	62.12	60.4
3	51	55.25	62	61.75	55.7
4	57.5	55	60.33	55.25	59.1
5	50.5	57	56.83	57.5	60.5
6	58	54	57.5	58.38	61
7	51	58.5	62	55.62	65.7
8	60	55.75	60.33	61.2	62.7
9	59	55.5	65.33	62	58.1
10	60.5	53.5	59.33	60.37	62.5

Lampiran 16 Waktu Validasi pada Model CNN

REPLIKASI KE-	Waktu				
	<i>5 Way 1 Shot</i>	<i>5 Way 2 Shot</i>	<i>5 Way 3 Shot</i>	<i>5 Way 4 Shot</i>	<i>5 Way 5 Shot</i>
1	1h 41m 19s	3h 34m 52s	5h 25m 51s	6h 59m 51s	8h 33m 11s
2	1h 37m 16s	3h 29m 46s	4h 54m 52s	7h 14m 19s	9h 26m 15s
3	1h 36m 51s	3h 38m 60s	4h 56m 13s	7h 33m 57s	7h 52m 54s
4	1h 34m 21s	3h 28m 57s	4h 52m 21s	6h 54m 57s	8h 9m 31s
5	1h 38m 59s	3h 19m 18s	4h 37m 13s	6h 12m 9s	8h 8m 48s
6	1h 51m 21s	3h 4m 32s	5h 37m 14s	7h 8m 0s	7h 50m 26s
7	1h 52m 38s	3h 16m 51s	5h 25m 9s	7h 40m 21s	8h 10m 57s
8	1h 37m 9s	3h 32m 16s	5h 3m 2s	6h 46m 45s	7h 42m 29s
9	1h 41m 58s	3h 30m 53s	4h 53m 38s	7h 2s	7h 41m 18s
10	1h 34m 16s	3h 35m 45s	5h 6m 41s	7h 35m 7s	7h 59m 17s

Lampiran 17 Code Analisis VGG-16 dan CNN

<https://www.kaggle.com/code/hanifdidin/few-Shot-learning-siamese-network-vgg-16>

<https://www.kaggle.com/code/hanifdidin/few-Shot-learning-siamese-network-cnn>

SURAT PERNYATAAN

Saya yang bertanda tangan di bawah ini, mahasiswa Departemen Statistika FSAD ITS:

Nama : Hanif Choiruddin

NRP : 5003211063

menyatakan bahwa data yang digunakan dalam Tugas Akhir/~~Thesis~~ ini merupakan data sekunder yang diambil dari ~~penelitian/buku/ Tugas-Akhir/Thesis/ publikasi lainnya~~ yaitu:

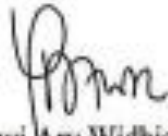
Sumber : Data *website*

<https://www.kaggle.com/datasets/adirizq/shopee-product-images>

Keterangan : Kumpulan data produk e-commerce (Shopee Indonesia) yang di-scraped dan digunakan untuk tugas kategori klasifikasi produk

Surat Pernyataan ini dibuat dengan sebenarnya. Apabila terdapat pemalsuan data maka saya siap menerima sanksi sesuai aturan yang berlaku

Mengetahui
Ko-Pembimbing Tugas Akhir


Tintrim Dwi Ary Widhianingsih,
S.Si., M.Stat., Ph.D.
NIP 19950520 202406 2 003

Surabaya, 9 - 8-2025
Mahasiswa


Hanif Choiruddin
NRP. 5003211063

Pembimbing Tugas Akhir


Dr. Dra. Kartika Fithriasari, M.Si.
NIP 19691212 199303 2 002

*(coret yang tidak perlu)

BIODATA PENULIS



Penulis dengan nama lengkap Hanif Choiruddin yang biasa dipanggil Hanif dilahirkan di Nganjuk, 07 Maret 2023, merupakan anak pertama dari 2 bersaudara. Penulis telah menempuh pendidikan formal yaitu SD Aisyiyah' 1 Nganjuk, MTs Aisyiyah' 1 Nganjuk, dan SMAN 1 Nganjuk. Setelah lulus dari SMAN tahun 2021 Penulis mengikuti seleksi SBMPTN dan diterima di Departemen Statistika FSAD – ITS pada tahun 2021 dan terdaftar dengan NRP 5003211063.

Di Departemen Statistika sempat menjadi sebagai asisten dosen di Departemen Statistika pada mata kuliah Statistical Machine Learning Jika terdapat hal yang ingin didiskusikan dengan Penulis terkait metode maupun topik yang ada di dalam penelitian ini, harap menghubungi via e-mail: realhanifchoiruddin@gmail.com.

(Halaman ini sengaja dikosongkan)