PROYEK AKHIR – VS231743

PENERAPAN *MULTIMODEL DEEP LEARNING* DALAM PENDETEKSIAN BERITA HOAKS LAMAN "TURNBACKHOAX.ID" MENGGUNAKAN ARSITEKTUR CNN

AHMAD RIZAL BAYHAQI

NRP 2043201047

Dosen Pembimbing

MUKTI RATNA DEWI, S.Si., M.Sc.

NIP 1992201912084

Co-Dosen Pembimbing

Mochammad Reza Habibi, S.Si., M.Mat.

NIP 2022199611051

Program Studi Sarjana Terapan

Departemen Statistika Bisnis

Fakultas Vokasi

Institut Teknologi Sepuluh Nopember

Surabaya

2023/2024

PROYEK AKHIR – VS231743

PENERAPAN *MULTIMODEL DEEP LEARNING* DALAM PENDETEKSIAN BERITA HOAKS LAMAN "TURNBACKHOAX.ID" MENGGUNAKAN ARSITEKTUR CNN

Ahmad Rizal Bayhaqi

NRP 2043201047

Dosen Pembimbing

Mukti Ratna Dewi, S.Si., M.Sc.

NIP 1992201912084

Co-Dosen Pembimbing

Mochammad Reza Habibi, S.Si., M.Mat.

NIP 2022199611051

Program Studi Sarjana Terapan

Departemen Statistika Bisnis

Fakultas Vokasi

Institut Teknologi Sepuluh Nopember

Surabaya

2023/2024

FINAL PROJECT – VS231743

APPLICATION OF MULTIMODEL DEEP LEARNING IN DETECTING HOAX NEWS ON THE "TURNBACKHOAX.ID" PAGE USING THE CNN ALGORITHM

Ahmad Rizal Bayhaqi

NRP 2043201047

Advisor

Mukti Ratna Dewi, S.Si., M.Sc.

NIP 1992201912084

Co-Advisor

Mochammad Reza Habibi, S.Si., M.Mat.

NIP 2022199611051

Study Program Applied Bachelor

Department of Business Statistics

Faculty of Vocation

Institut Teknologi Sepuluh Nopember

Surabaya

2023/2024

Halaman ini sengaja dikosongkan

LEMBAR PENGESAHAN

Penerapan Multimodel Deep Learning Dalam Pendeteksian Berita Hoaks Laman "Turnbackhoax.id" Menggunakan Arsitektur CNN

PROYEK AKHIR

Diajukan untuk memenuhi salah satu syarat

memperoleh gelar Sarjana Terapan pada

Program Studi D4

Departemen Statistika Bisnis

Fakultas Vokasi

Institut Teknologi Sepuluh Nopember

Oleh: **Ahmad Rizal Bayhaqi**

NRP. 2043201047

Disetujui oleh Tim Penguji Proyek Akhir:

1.	Mukti Ratna Dewi, S.Si., M.Sc.	Pembimbing
2.	Mochammad Reza Habibi, S.Si., M.Mat.	Co-pembimbing
3.	Dra.Destri Susilaningrum, M.Si.	Penguji 1
4.	Zakiatul Wildani, S.Si., M.Sc.	Penguji 2

SURABAYA

25 April, 2024

Halaman ini sengaja dikosongkan

APPROVAL SHEET

Application of Multimodel Deep Learning in Detecting Hoax News on the "Turnbackhoax.id" Page Using the CNN Algorithm

FINAL PROJECT

Submitted to fulfill one of the requirements

for obtaining a degree Applied Bachelor of Business Statistics at

Undergraduate Study Program of D4

Department of Business Statistics

Faculty of Vocation

Institut Teknologi Sepuluh Nopember

By: **Ahmad Rizal Bayhaqi**

NRP. 2043201047

Approved by Final Project Examiner Team:

1. Mukti Ratna Dewi, S.Si., M.Sc. — Advisor
2. Mochammad Reza Habibi, S.Si., M.Mat. — Co-Advisor
3. Dra.Destri Susilaningrum, M.Si. — Examiner 1
4. Zakiatul Wildani, S.Si., M.Sc. — Examiner 2

SURABAYA

25 April, 2024

Halaman ini sengaja dikosongkan

PERNYATAAN ORISINALITAS

Yang bertanda tangan di bawah ini:

Nama mahasiswa / NRP	:	Ahmad Rizal Bayhaqi / 2043201047
Program studi	:	Statistika Bisnis
Dosen Pembimbing / NIP	:	Mukti Ratna Dewi, S.Si., M.Sc. / 1992201912084
Co.Dosen Pembimbing / NIP	:	Mochammad Reza Habibi, S.Si., M.Mat. / 2022199611051

dengan ini menyatakan bahwa Proyek Akhir dengan judul **"Penerapan Multimodel Deep Learning Dalam Pendeteksian Berita Hoaks Laman Turnbackhoax.id Menggunakan Arsitektur CNN"** adalah hasil karya sendiri, bersifat orisinal, dan ditulis dengan mengikuti kaidah penulisan ilmiah.

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

Surabaya, 25 April 2024

Mengetahui Dosen Pembimbing	Dosen Co-Pembimbing	Mahasiswa
Mukti Ratna Dewi, S.Si., M.Sc.	Mochammad Reza Habibi, S.Si., M.Mat.	Ahmad Rizal Bayhaqi
NIP. 1992201912084	NIP. 2022199611051	NRP. 2043201047

Halaman ini sengaja dikosongkan

STATEMENT OF ORIGINALITY

The undersigned below:

Name of Student / NRP	:	Ahmad Rizal Bayhaqi / 2043201047
Department	:	Statistika Bisnis
Advisor / NIP	:	Mukti Ratna Dewi, S.Si., M.Sc. / 1992201912084
Co.Advisor / NIP	:	Mochammad Reza Habibi, S.Si., M.Mat. / 2022199611051

hereby declare that the Final Project with the title of **"Application of Multimodel Deep Learning in Detecting Hoax News on the Turnbackhoax.id Page Using the CNN Algorithm"** 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, 25 April 2024

Acknowledged Advisor	Co-Advisor	Student
Mukti Ratna Dewi, S.Si., M.Sc.	Mochammad Reza Habibi, S.Si., M.Mat.	Ahmad Rizal Bayhaqi
NIP. 1992201912084	NIP. 2022199611051	NRP. 2043201047

Halaman ini sengaja dikosongkan

ABSTRAK

Penerapan Multimodel Deep Learning Dalam Pendeteksian Berita Hoaks Laman "Turnbackhoax.id" Menggunakan Algoritma CNN

Nama Mahasiswa / NRP	**:**	**Ahmad Rizal Bayhaqi / 2043201047**
Departemen	**:**	**Statistika Bisnis FV – ITS**
Dosen Pembimbing	**:**	**Mukti Ratna Dewi, S.Si., M.Sc.**
Dosen Ko.Pembimbing	**:**	**Mochammad Reza Habibi, S.Si., M.Mat.**

Abstrak

Meningkatnya penyebaran berita hoaks memicu kebutuhan publik untuk mengatasi permasalahan tersebut. Masyarakat Anti Fitnah Indonesia (Mafindo), sebagai komunitas independen, berupaya memberikan edukasi dan identifikasi terkait berita hoaks. Meski demikian, proses verifikasi berita yang masih manual dan tidak cukup efektif mengingat jumlah berita hoaks yang tersebar cepat dan dalam jumlah besar. Oleh karena itu, penelitian dilakukan untuk membangun model *machine learning* menggunakan *deep learning*, khususnya *convolutional neural network* (CNN), dalam mengklasifikasikan berita hoaks secara cepat dan otomatis. Penggunaan CNN berbasis data teks dan gambar telah menunjukkan performa klasifikasi yang baik, terutama ketika kedua data digabungkan dalam *multimodel deep learning*. *Multimodel deep learning* atau model CNN gabungan, menggabungkan model CNN berbasis teks (CNN 1D) dan gambar (CNN 2D) yang menunjukkan kinerja lebih baik dibandingkan dengan model tunggal (*unimodel*). Model tersebut kemudian dilatih dengan 3103 data *training* dan 775 data *testing* dan diperoleh nilai akurasi 99,35% dan AUC 99,81%. Hasil evaluasi menunjukkan kinerja yang baik dalam mengklasifikasikan berita hoaks di dalam dan di luar model.

Kata kunci: CNN, *Hoax, Multimodel Deep Learning*.

Halaman ini sengaja dikosongkan

ABSTRACT

Application of Multimodel Deep Learning in Detecting Hoax News on the "Turnbackhoax.id" Page Using the CNN Algorithm

Student Name / NRP	**:**	**Ahmad Rizal Bayhaqi / 2043201047**
Department	**:**	**Statistika Bisnis FV – ITS**
Advisor	**:**	**Mukti Ratna Dewi, S.Si., M.Sc.**
Co.Advisor	**:**	**Mochammad Reza Habibi, S.Si., M.Mat.**

Abstract

The increasing spread of hoax news has prompted the public's need to address this issue. The Indonesian Anti-Slander Society (Mafindo), as an independent community, strives to provide education and identification regarding hoax news. However, the manual verification process proves ineffective given the rapid and large-scale dissemination of hoax news. Therefore, research is conducted to build a machine learning model using deep learning, specifically convolutional neural network (CNN), for the fast and automatic classification of hoax news. The use of CNN based on text and image data has shown good classification performance, especially when both data types are combined in multimodal deep learning. Multimodal deep learning, or the combined CNN model, merges text-based CNN (CNN 1D) and image-based CNN (CNN 2D), demonstrating superior performance compared to a single model (unimodal). The model is then trained with 3103 training data and 775 testing data, resulting in an accuracy of 99.35% and an AUC of 99.81%. Evaluation results indicate excellent performance in classifying hoax news both within and outside the model.

Keywords: CNN, *Hoax, Multimodal Deep Learning*.

Halaman ini sengaja dikosongkan

KATA PENGANTAR

Dengan rasa syukur dan penuh hormat, penulis ingin menyampaikan puji dan syukur kepada Allah SWT., yang telah melimpahkan segala rahmat, karunia, dan hidayah-Nya, sehingga Proyek Akhir berjudul **"Penerapan Multimodel Deep Learning Dalam Pendeteksian Berita Hoaks Laman Turnbackhoax.id Menggunakan Algoritma CNN"** dapat diselesaikan dengan baik. Ketercapaian ini tidak lepas dari doa, dukungan, dan bantuan yang diberikan oleh berbagai pihak, dan oleh karena itu, penulis ingin menyampaikan ucapan terima kasih yang sebesar-besarnya kepada.

1. Bapak Prof. Dr. Wahyu Wibowo, S.Si., M.Si. selaku Kepala Departemen Statistika Bisnis, Fakultas Vokasi, ITS yang telah memberikan dukungan dalam masa perkuliahan dan penyelesaian proyek akhir
2. Ibu Mukti Ratna Dewi, S.Si., M.Sc., dosen pembimbing utama, dan Bapak Mochammad Reza Habibi, S.Si., M.Mat., dosen pembimbing pendamping, atas semua bimbingan, pengetahuan, dan semangat yang luar biasa.
3. Ibu Dra.Destri Susilaningrum, M.Si., dan Zakiatul Wildani, S.Si., M.Sc., selaku dosen penguji, yang memberikan kritik dan saran membangun untuk menyempurnakan Proyek Akhir ini.
4. Ibu Iis Dewi Ratih, S.Si., M.Si. selaku dosen wali yang selalu memberikan semangat, dukungan, nasihat, serta arahan selama masa perkuliahan di Departemen Statistika Bisnis.
5. Bapak dan Ibu Dosen serta seluruh Tendik yang telah memberikan ilmu dan dukungan selama masa perkuliahan.
6. Ayah Aslahul Umam dan Ibu Dyah Asvihani, kedua orang tua yang selalu memberikan doa, dukungan, dan kasih sayang tanpa henti. Kehadiran mereka merupakan sumber inspirasi dan motivasi yang memandu langkah penulis dalam menyelesaikan perjalanan ini.
7. Semua pihak yang telah memberikan dukungan dalam penyelesaian Proyek Akhir ini, meskipun tidak dapat disebutkan satu persatu. Semua kontribusi, doa, dan semangat yang diberikan oleh teman-teman, rekan-rekan sejawat, dan semua pihak yang turut serta dalam perjalanan penulisan ini sangat dihargai.

Semua doa, dukungan, dan bimbingan yang diterima telah menjadi pendorong kesuksesan Proyek Akhir ini. Semoga hasilnya dapat memberikan manfaat dan kontribusi positif untuk perkembangan ilmu pengetahuan di masa depan. Terima kasih atas segala kontribusi dan kerjasama yang telah diberikan

Surabaya, 25 April 2024

Penulis

Halaman ini sengaja dikosongkan

DAFTAR ISI

Halaman ini sengaja dikosongkan

DAFTAR GAMBAR

Halaman ini sengaja dikosongkan

DAFTAR TABEL

Halaman ini sengaja dikosongkan

BAB I
PENDAHULUAN

1.1 Latar Belakang

Informasi atau berita yang banyak tersebar di internet tidak semua dapat dipercaya, sebagian informasi dibuat untuk tujuan menyesatkan pembaca oleh pihak tak bertanggung jawab, dengan kata lain disebut hoaks. Hoaks merujuk pada kabar, berita, informasi palsu atau bohong yang sengaja disebarkan untuk mencapai tujuan tertentu. Terdapat beberapa alasan atau faktor penyebab munculnya berita hoaks, yaitu sebagai hiburan, meramaikan perkembangan tren di internet, mencari sensasi, dan sebagai pekerjaan untuk menyudutkan pihak tertentu (Mazaya, 2019). Saat ini persebaran berita hoaks sangat luas dan mudah dijumpai dalam kehidupan sehari-hari. Mengutip publikasi Kementerian Komunikasi dan Informatika (Kominfo), sepanjang tahun 2018 setidaknya ada 800.000 situs internet penyebar hoaks dan triwulan pertama tahun 2023 Kominfo telah mengidentifikasi sebanyak 425 isu hoaks yang beredar melalui website di tanah air (Kominfo, 2023). Sementara hasil survei yang dilakukan Katadata Insight Center (KIC) bersama Kominfo menunjukkan bahwa 30%-60% masyarakat indonesia terpapar berita hoaks saat beraktivitas di dunia maya (Harnum, 2022).

Berita hoaks menyebar dengan sangat cepat dan merupakan permasalahan serius di Indonesia. Hoaks merugikan berbagai sektor kehidupan serta mengganggu kenyamanan dan keamanan masyarakat. (Pranesti, 2019). Dampak negatif berita hoaks juga dapat memecah belah publik dan memicu ujaran kebencian terhadap individu hingga kelompok atau lembaga tertentu (Huroh, 2022). Seperti yang terjadi pada kasus rasisme terhadap mahasiswa Papua di Surabaya yang dipicu oleh berita hoaks yang disebar melalui media sosial. Berita tersebut dibagikan oleh Tri Susanti melalui *whatsapps group* dengan narasi bahwa mahasiswa Papua merobek dan mematahkan tiang bendera merah putih, hingga pada akhirnya memicu ketegangan antar kelompok ormas setempat dan mahasiswa Papua di Surabaya (Siahaan & Tampubolon, 2021). Tragedi rasisme terhadap mahasiswa papua menunjukkan penyebaran berita sangat cepat melalui media sosial dan memberikan dampak yang signifikan terhadap masyarakat. Berbeda dengan berita konvensional dan berita untuk melawan hoaks, dimana penerbitan keduanya harus melewati proses kaidah jurnalistik, sementara hoaks bisa dilakukan oleh siapa saja dengan konten apa saja tanpa melalui proses kaidah jurnalistik (Dulkiah, 2020). Keterbatasan tersebut yang membuat berita hoaks dapat tersebar secara masif mengungguli berita konvensional saat ini, sangat berbahaya, dan sulit untuk diverifikasi (Simarmata, 2019).

Upaya menekan persebaran berita hoaks dilakukan oleh pemerintah melalui Kominfo dan kelompok-kelompok masyarakat salah satunya Masyarakat Anti Fitnah Indonesia (Mafindo). Mafindo merupakan gerakan komunitas independen yang bergerak di bidang literasi dan edukasi masyarakat mengenai hoaks, Mafindo didirikan untuk meminimalisasi penyebaran informasi bohong (hoaks) di media sosial (Dilla, 2019). Dalam upaya menekan persebaran hoaks akibat perkembangan teknologi media sosial Mafindo membangun komunitas media sosial dan membangun website *turnbackhoax.id.* Melalui website *turnbackhoax.id*, Mafindo berfokus membagikan informasi hoaks yang sudah diverifikasi. Namun proses verifikasi suatu informasi atau berita saat ini masih dilakukan secara manual, dimana kecepatan prosesnya tidak sebanding dengan jumlah persebaran berita hoaks yang masif beredar di masyarakat. Jika tidak segara diatasi, penyebaran berita hoaks akan semakin luas serta pihak pemerintah dan Mafindo akan tertinggal, serta tidak mampu memberi penanganan secara maksimal.

Mengatasi permasalahan berita hoaks yang menyebar cepat di media sosial dan jumlahnya yang masif dapat diselesaikan dengan bantuan mesin. Penggunaan mesin akan membuat proses klasifikasi dapat berjalan secara otomatis, tidak lagi manual, dan tidak memakan banyak waktu (Asriyar, 2019). Mesin klasifikasi berita hoaks dan fakta dapat dibangun menggunakan berbagai jenis model, contohnya model *deep learning*. Model *deep learning* (DL) merupakan salah satu jenis model yang banyak digunakan dalam tugas mendeksi teks, citra digital, pengenalan suara, dan lain-lain (Yunanto, 2021). Kebanyakan model DL dirancang untuk fokus pada pengolahan satu jenis data (*unimodel*) semisal mengolah teks saja atau gambar saja. Dalam kasus model klasifikasi berita penggunaan model berbasis teks paling banyak digunakan dalam klasifikasi berita hoaks dan fakta berbahasa indonesia.

Penggunaan model klasifikasi tunggal (*unimodel*) dapat ditingkatkan dengan menggunakan gabungan model berbasis teks dan model berbasis gambar (*multimodel*). Menurut Triasa dan Herlina bahwa penggunaan *news photho or caption* (foto berita) pada berita hoaks mampu memengaruhi emosi pembaca. Foto berita mempengaruhi emosi pembaca melalui dua hal yaitu, foto berita sebagai bukti atas terjadinya sebuah peristiwa serta foto sebagai pembangkit beragam emosi untuk membuat orang lebih mudah memercayai hoaks (Hawari, 2019). Berdasarkan penelitian tersebut menambahkan foto berita memiliki kemungkinan untuk meningkatkan model dalam mengklasifikasi berita hoaks dan fakta. Sementara hasil gabungan dua model berbasis teks dan gambar tersebut disebut *multimodel deep learning*. Penelitian yang dilakukan Li dkk. menjelaskan penggunaan *multimodel deep learning* pada berita satire di Amerika Serikat menunjukkan performa *multimodel* lebih baik dibanding model dengan foto berita saja atau text saja (Li, 2020). Penelitian-penelitan di atas menjadi dasar untuk mengembangkan *multimodel deep learning* pada foto berita dan text berbahasa indonesia untuk mengidentifikasi berita hoaks dan fakta. Adapun penggunaan *multimodel deep learning* dengan arsitektur *convolution Neural network* (CNN) memiliki performa unggul dibanding arsitektur lain dalam mengklasifikasikan berita hoaks berbahasa Indonesia (Kurniawan, 2021). Diharapkan penggunaan *multimodel deep learning* dengan arsitektur CNN menjadi solusi mengimbangi persebaran berita hoaks yang masif.

Berdasarkan permasalahan dan penelitian-penelitian sebelumnya, penelitian ini bertujuan untuk membangun model klasifikasi berita hoaks dan fakta berbahasa indonesia menggunakan *multimodel deep learning* yang berbasis pada berita di laman *turnbackhoax.id*. *Multimodel deep learning* menggunakan teks dan foto berita dari berita sebagai input dalam bentuk vektor. Model diharapkan dapat dimanfaatkan untuk melakukan prediksi apabila terdapat berita atau laporan informasi baru yang masuk. Klasifikasi berita dengan model akan berjalan secara otomatis dan terstruktur untuk mengidentifikasi kebenaran suatu berita, sehingga mempercepat proses kerja deteksi dan verifikasi berita yang tersebar di masyarakat dan media sosial.

1.2 Rumusan Masalah

Berita hoaks, yang merupakan informasi palsu dengan dampak menyesatkan pembaca, saat ini ditangani secara manual oleh Mafindo. Namun, penanganan manual ini tidak sebanding dengan tingkat penyebaran berita hoaks yang masif di masyarakat. Untuk mengatasi tantangan ini, dilakukan penelitian untuk mengembangkan mesin dengan *model deep learning* yang dapat mengklasifikasikan berita hoaks secara otomatis. Latar belakang ini mendorong pertanyaan tentang bagaimana *multimodel deep learning* dengan arsitektur CNN dapat efektif mengklasifikasikan berita hoaks berbahasa Indonesia. Selain itu, model perlu dievaluasi dan dibandingkan nilai kinerja antara *multimodel* dengan *unimodel* lainnya, dalam mengklasifikasikan berita hoaks dan fakta berbahasa Indonesia. Model klasifikasi yang

dibandingkan mencakup model dengan teks saja, model dengan gambar saja, dan model gabungan teks dan gambar (*multimodel*), dengan setiap model menggunakan arsitektur CNN sebagai dasar pengklasifikasian.

1.3 Tujuan

Berdasarkan Rumusan permasalahan yang telah dijelaskan sebelumnya, penelitian ini bertujuan sebagai berikut.

1. Membangun *multimodel deep learning* dengan arsitektur CNN (Model CNN Gabungan) dan mengklasifikasikan berita hoaks dan fakta berbahasa Indonesia.
2. Melakukan perbandingan evaluasi model pada *multimodel* dan *unimodel* dalam mengklasifikasikan berita hoaks dan fakta berbahasa Indonesia.

1.4 Batasan Masalah

Batasan masalah pada penelitian ini menggunakan database website *turnbackhoax.id* milik Mafindo dan database website *republika.co.id* dengan periode publikasi berita yang digunakan pada tanggal 1 Januari 2023 hingga 13 Oktober 2023. Penelitian berfokus pada foto berita dan konten berita.

1.5 Manfaat

Penelitian ini secara umum diharapkan dapat memberikan kontribusi dalam mengatasi masalah penyebaran hoaks dan bias dalam informasi. Pengembangan lebih lanjut diharapkan dapat membantu Mafindo dalam melakukan klasifikasi berita hoaks secara lebih cepat. Selain itu, penelitian ini juga diharapkan dapat memberikan wawasan dan pemahaman yang lebih baik mengenai teknik-teknik klasifikasi dan deteksi hoaks menggunakan pendekatan *multimodel deep learning*.

Halaman ini sengaja dikosongkan

BAB II
TINJAUAN PUSTAKA

2.1 Penelitian Terdahulu

Penelitian terkait klasifikasi berita hoaks sudah pernah dilakukan menggunakan berbagai pendekatan dan metode *machine learning*. Berikut ringkasan penelitian terdahulu terkait klasifikasi berita hoaks menggunakan model *machine learning* dan *deep learning*.

1. Penelitian yang dilakukan oleh Yuliani dan Sahib, yaitu membandingkan performa klasifikasi berita hoaks dari enam algoritma ML menunjukkan model *neural network* (NN) memiliki hasil akurasi terbaik sebesar 93,33%. Sedangkan model *support vektor machine* (SVM), *naive bayes, decision tree, logistic regression nilai* akurasi tidak lebih besar dari 91% (Yuliani, 2019).
2. Penelitian terhadap penggunaan teks dan gambar untuk meningkatkan kemampuan model klasifikasi tanggap bencana oleh Ferda Olfi, model menunjukkan bahwa arsitektur *multimodel* yang diusulkan memberikan kinerja yang lebih baik daripada model yang dilatih menggunakan satu model saja (misalnya, teks atau gambar saja) (Ofli, 2020)
3. Penelitian yang dilakukan oleh Antonius dan Metty dalam klasifikasi berita hoaks berbahasa Indonesia. Hasil penelitian menunjukkan akurasi model CNN sebesar 88%, sedangkan model LSTM 4% lebih rendah dibawah CNN (Kurniawan, 2021).
4. Penelitian terkait *multimodel deep learning* menggunakan data gambar dan teks oleh Edward Kim, menunjukkan model multimodel mampu melebihi kinerja setiap model individu dan kemampuan annotator manusia rata-rata (Kim, 2018).
5. Selanjutnya penelitian yang dilakukan Hussein, Rizk, dan Award . Hasil pengujian model tunggal memiliki hasil akurasi dibawah 90%, sementara *multimodel deep learning* memiliki akurasi di atas 90% (Mouzannar, 2018).

2.2 Web scraping

Web scraping adalah proses pengambilan dokumen semi-terstruktur dari halaman web untuk mendapatkan data tertentu dari halaman tersebut. Teknik ini digunakan untuk mengekstraksi informasi dari situs web dan menyimpannya dalam file atau database untuk keperluan analisis data. *Web scraping* memungkinkan pengumpulan *big data* secara efisien dan kuat yang tidak dapat dilakukan oleh mesin pencari tradisional seperti Google Search. (Djufri, 2020).

2.3 Data Labeling

Data labeling adalah metode pemberian label otomatis pada data berdasarkan input kata kunci khas yang dimiliki oleh masing-masing data. Proses ini memanfaatkan algoritma atau aturan untuk mengenali kata kunci spesifik dalam dataset, dan ketika kata kunci tersebut terdeteksi, label atau kategori secara otomatis diberikan kepada data. Misalnya, dalam konteks berita, kata kunci seperti "politik" atau "ekonomi" dapat digunakan untuk memberikan label kategori. Keuntungan utama dari metode ini adalah efisiensi dan konsistensi dalam memberikan label pada dataset besar, dengan penghematan waktu dan sumber daya. Namun, penting untuk memantau tingkat akurasi dan hasil secara berkala guna memastikan bahwa metode ini memberikan label yang sesuai dengan konteks dan tujuan analisis data.

2.4 Data Pre-processing

Data pre-processing adalah proses persiapan dan pembersihan data sebelum diolah lebih lanjut oleh algoritma atau model *machine learning*. Tujuan dari *pre-processing* adalah untuk menghilangkan informasi yang tidak relevan, mengurangi informasi yang mengganggu, dan menghasilkan data teks yang lebih terstruktur dan mudah diolah oleh model (Fahmi, 2021). *Data Preprocessing* pada penelitian ini dibagi kedalam tiga tahapan yaitu *data exploration, data cleaning,* dan *data vectorization* yang dijelaskan sebagai berikut.

2.4.1 Data Exploration

Tahap ini, tujuan utama adalah untuk memahami struktur dan karakteristik dasar dari data yang akan diolah. Eksplorasi data membantu dalam mengidentifikasi potensi masalah dan memahami distribusi variabel. Pada data teks, hasil eksplorasi dilakukan dengan mengetahui beberapa hal seperti jumlah dokumen dalam *corpus* (kumpulan dokumen), total token (menghitung semua token dalam korpus, dan token unik (total token yang hanya ada dalam korpus bukan token duplikat) (Nurtikasari, 2022). Sebaran token selanjutnya dapat divisualisasikan untuk mengetahui bagaimana hubungan frekuensi antar token dalam satu *corpus* atau kategori dokumen. Sementara pada data gambar proses eksplorasi dilakukan untuk mengetahui bagaimana ukuran resolusi gambar. Ukuran dari resolusi gambar dinyatakan dalam satuan *pixel*, dimana setiap pixel tersusun dalam matriks. Resolusi gambar memiliki dimensi tinggi (*height*) dan lebar (width) yang dapat mempengaruhi ukuran dari matriks yang terbentuk.

2.4.2 Data Cleaning

Setelah memahami struktur data, langkah selanjutnya adalah membersihkan data dari kesalahan atau nilai yang tidak valid. Proses ini meningkatkan kualitas data dan mengurangi kemungkinan kesalahan dalam model. Data yang sudah dibersihkan akan dilakukan eksplorasi kembali seperti pada tahap data exploration. Berikut adalah langkah-langkah untuk tahap data cleaning:

A. Case Folding

Case folding merupakan proses menyamaratakan semua huruf sehingga tidak ada perbedaan antar huruf kapital dan huruf bukan kapital, seperti yang ditunjukkan oleh Tabel 2.1 berikut:

Tabel 2.1 Contoh *Casefolding*

Index	Dokumen	Hasil *Case Folding*
1	1Mafindo Perlu Ada Kode Etik Bagi Influencer Selamat Pagi Indonesia 09 Agustus 2020 095455 Beberapa waktu lalu ramai menjadi pembicaraan soal konten video yang dibuat musisi Anji bersama Hadi Pranoto	1mafindo perlu ada kode etik bagi influencer selamat pagi indonesia 09 agustus 2020 095455 beberapa waktu lalu ramai menjadi pembicaraan soal konten video yang dibuat musisi anji bersama hadi pranoto

B. Menghapus Format Penanggalan

Proses menghapus format tanggal dan waktu yang dilakukan memiliki pada pola umum seperti "YYYY-MM-DD, HH:MM:SS.", format dihapus dari teks menggunakan fungsi *regular expression.* Hasil dari proses ini ditampilkan pada Tabel 2.2 berikut:

Tabel 2.2 Contoh Menghapus Format Penanggalan

Index	Dokumen	Hasil Penghapusan
1	1mafindo perlu ada kode etik bagi influencer selamat pagi indonesia 09 agustus 2020 095455 beberapa waktu lalu ramai menjadi pembicaraan soal konten video yang dibuat musisi anji bersama hadi pranoto	1mafindo perlu ada kode etik bagi influencer selamat pagi indonesia beberapa waktu lalu ramai menjadi pembicaraan soal konten video yang dibuat musisi anji bersama hadi pranoto

C. Menghapus Satuan

Menghapus satuan merupakan proses bertujuan untuk menghilangkan satuan waktu seperti menit, jam, hari, minggu, dll. Selain itu, menghapus satuan byte, satuan berat, satuan uang dan menghilangkan pecahan dari teks. Contoh proses menghapus satuan ditampilkan pada Tabel 2.3 berikut:

Tabel 2.3 Contoh Menghapus Satuan

Index	Dokumen	Hasil Penghapusan
1	Saat ini harga paket data 1GB melambung naik menjadi 52 000 rupiah dari sebelumnya Rp 40 000	Saat ini harga paket data 1 melambung naik menjadi 52 000 dari sebelumnya 40 000

D. Menghapus Link

Menghapus link merupakan proses menghilangkan kode aktivasi, URL, dan referensi lainnya. Hasil dari proses ini ditampilkan seperti pada Tabel 2.4 berikut:

Tabel 2.4 Contoh Menghapus Link

Index	Dokumen	Hasil Penghapusan
1	Organisasi lingkungan Green Earth juga menyatakan keprihatinan dan mengajak masyarakat untuk berkontribusi melalui donasi yang dapat dilakukan dengan memasukkan Kode Aktivasi "SEA2023" saat berdonasi melalui situs mereka. Referensi: (URL publikasi: www.jurnalilmiah.com/climateimpact | Green Earth: www.greenearth.org/donate)	Organisasi lingkungan Green Earth juga menyatakan keprihatinan dan mengajak masyarakat untuk berkontribusi melalui donasi yang dapat dilakukan dengan memasukkan Kode Aktivasi "" saat berdonasi melalui situs mereka. Referensi: ()

E. Menghapus Emoticon, Angka, dan karakter khusus lainnya.

Proses menghilangkan emoticon, angka, dan karakter khusus lainnya ditampilkan pada Tabel 2.5 berikut:

Tabel 2.5 Contoh Mengahapus Karakter Khusus

Index	Dokumen	Hasil Penghapusan
1	Selamat ulang tahun ke-25! Hari ini adalah momen spesial untukmu, semoga semua harapan dan impianmu terwujud. Have a blast!	Selamat ulang tahun ke Hari ini adalah momen spesial untukmu semoga semua harapan dan impianmu terwujud Have a blast

F. Punctuation

Punctuation merupakan proses menghilangkan tanda baca, kumpulan simbol dan tanda, yang digunakan dalam bahasa tertulis untuk memberikan struktur dan arti pada teks. Contoh penerapan *punctuation* ditampilkan pada Tabel 2.6 berikut:

Tabel 2.6 Contoh *Punctuation*

Index	Dokumen	Hasil Penghapusan
1	Mafindo: “Perlu Ada Kode Etik Bagi Influence...”	Mafindo Perlu Ada Kode Etik Bagi Influence...

G. Lemmatizing

Lemmatizing adalah proses dalam pemrosesan bahasa alami (natural language processing/NLP) yang bertujuan untuk mengubah kata-kata dalam bentuk *infleksional* (kata dengan bentuk berubah karena waktu, aspek, jenis kata, dll.) menjadi kata dasar atau bentuk dasarnya yang disebut *lema*. *Lema* adalah kata dasar dari suatu kata yang menggambarkan makna inti atau bentuk standar dari kata tersebut. Contoh hasil penerapan *lemmatizing* ditunjukkan pada Tabel 2.7 sebagai berikut:

Tabel 2.7 Contoh *Lemmatizing*

Index	Dokumen	Hasil *Lemmatizing*
1	Pada suatu pagi di Indonesia, Mafindo, seorang influencer ternama, menyadari perlunya kode etik bagi para influencer. Kontroversi konten video beberapa musisi beberapa waktu lalu menguatkan kebutuhan untuk bertanggung jawab dan memberikan contoh yang baik dalam platform digital. Mafindo berkomitmen untuk menyajikan konten yang positif dan mengedepankan nilai-nilai baik.	Pada satu pagi di Indonesia, Mafindo, seorang influencer ternama, sadar perlunya kode etik bagi para influencer. Kontroversi konten video beberapa musisi beberapa waktu lalu kuat kebutuhan untuk bertanggung jawab dan beri contoh baik dalam platform digital. Mafindo komitmen untuk saji konten yang positif dan edepankan nilai baik.

H. Tokenisasi

Dalam analisis teks dan pemrosesan bahasa alami (*Natural Language Processing*/NLP), token merujuk pada unit-unit terkecil yang dihasilkan setelah suatu teks yang dibagi-bagi, biasanya dalam bentuk kata atau frasa. Token unik dalam konteks ini merujuk pada himpunan kata-kata unik yang muncul dalam suatu teks setelah proses tokenisasi. Tokenisasi merupakan proses untuk memisahkan kata/*word* pada masing-masing dokumen kedalam sebuah daftar kata berindeks. Contoh hasil penerapan tokenisasi ditunjukkan pada Tabel 2.8 sebagai berikut:

Tabel 2.8 Contoh Tokenisasi

Index	Dokumen	Hasil Tokenisasi
1	"mafindo perlu ada kode etik bagi influencer selamat pagi indonesia beberapa waktu lalu ramai jadi bicara soal konten video yang buat musisi ... indonesia"	['mafindo', 'perlu', 'ada', 'kode', 'etik', 'bagi', 'influencer', 'selamat', 'pagi', 'indonesia', 'beberapa', 'waktu', 'lalu', 'ramai', 'jadi', 'bicara', 'soal', 'konten', 'video', 'yang', 'buat', 'musisi', ..., 'indonesia']

2.4.3 Vektorisasi Data

Vektorisasi data adalah proses mengubah data dari bentuk aslinya ke dalam bentuk vektor numerik. Tujuan utama vektorisasi adalah membuat data dapat dimengerti dan diproses oleh model atau algoritma *machine learning*. Proses ini diterapkan tergantung pada jenis data yang dihadapi, dan tujuan akhirnya adalah mengonversi data menjadi representasi numerik yang dapat diolah oleh algoritma *machine learning* atau analisis data. (Ramayasa, 2023)

Vektorisasi dilakukan pada data teks dan foto berita dengan tujuan mendapatkan data dengan dimensi vektor atau matriks. Vektorisasi data teks menggunakan *Term Frequency-Inverse Document Frequency* (TF-IDF) adalah salah satu metode yang umum digunakan. Proses ini melibatkan konversi dokumen teks yang sudah dalam bentuk token menjadi vektor

numerik dengan panjang mengikuti jumlah token unik dalam korpus. Sementara vektorisasi data gambar melibatkan konversi gambar ke dalam bentuk kumpulan vektor numerik atau disebut matriks. Vektorisasi dapat dilakukan dengan menggunakan format array melalui library *Numpy*. Vektor atau matriks memiliki kemungkinan berbeda ukuran dimensinya atau penggunaan seluruh elemen pada vektor dan matriks akan membebani proses komputasi. Untuk itu, perlu memastikan bahwa matriks gambar foto berita dan jumlah vektor token sesuai dengan kebutuhan model.

Penentuan panjang vektor dan matriks dilakukan berdasarkan persyaratan pada tiap model dalam hal ini model CNN. Pada model CNN berbasis teks, model menerima vektor TF-IDF dengan panjang vektor yang sama. Kedua, dilakukan pemotongan panjang vektor TF-IDF atau mengambil sebagian untuk mengurangi beban komputasi. Sama seperti model CNN teks pada model CNN berbasis gambar dibutuhkan foto berita dengan resolusi yang sama untuk menghasilkan matriks yang seragam. Kedua, dilakukan pengecilan resolusi dari ukuran semula untuk mengurangi beban komputasi. Sementara dalam menentukan nilai panjang vektor atau ukuran resolusi yang digunakan dapat diperoleh melalui penelitian sebelumnya terhadap model terkait.

2.5 TF-IDF

TF-IDF adalah singkatan dari *Term Frequency-Inverse Document Frequency* yang merupakan sebuah metode algoritma yang berguna untuk menghitung bobot dari setiap kata yang umum digunakan. Metode ini juga dikenal sebagai metode yang efisien, mudah, dan memberikan hasil yang akurat. Nilai TF-IDF untuk setiap token (kata) dalam setiap dokumen dalam korpus akan dihitung dengan menerapkan interaksi antara TF dan IDF (Lubis, 2021). Nilai *Term Frequency* (TF) mengukur seberapa sering sebuah kata muncul dalam sebuah dokumen. Jumlah kemunculan kata tersebut dibagi dengan total kata dalam dokumen untuk menghasilkan nilai normalisasi. Tujuannya adalah untuk memberikan bobot yang lebih tinggi pada kata-kata yang muncul lebih sering dalam dokumen tersebut, karena kata-kata tersebut dianggap lebih relevan dengan konten dokumen tersebut, menghitung nilai TF dapat dilakukan menggunakan persamaan (2.1) berikut ini (Qaiser, 2018):

$$TF_{k,d} = \begin{Bmatrix} 1+\log_{10}(f_{k,d}),\ f_{k,d} > 0 \\ 0,\ f_{k,d} = 0 \end{Bmatrix} \tag{2.1}$$

Persamaan (2.1) menunjukkan $f_{k,d}$ merupakan frekuensi kata *k* yang muncul pada dokumen *d*. Sementara jika kata tidak ditemukan dalam dokumen *d* maka, $TF_{k,d}$ akan bernilai nol. Kemudian Inverse Document Frequency (IDF), pendekatan ini mengukur seberapa umum atau jarang sebuah kata dalam sebuah korpus (kumpulan dokumen). IDF membantu memberikan bobot yang lebih tinggi pada kata-kata yang relatif jarang muncul dalam keseluruhan dokumen (korpus), karena kata-kata tersebut dianggap lebih informatif atau memuat informasi yang lebih khusus. Persamaan IDF dapat diketahui dari persamaan (2.2) berikut ini (Qaiser, 2018):

$$IDF_k = \begin{Bmatrix} \log\left(\frac{N}{df_k}\right), N > df_k \\ \log\left(\frac{N}{df_k}\right)+1, N = df_k \end{Bmatrix} \tag{2.2}$$

Persamaan (2.2) menjelaskan *N* adalah jumlah seluruh dokumen dalam korpus, sedangkan df_k adalah jumlah dokumen dalam korpus mengandung kata *k*. Kemudian jika *N* bernilai sama dengan df_k. maka IDF_k bernilai satu. Selanjutnya TF-IDF yang merupakan hasil perkalian dari nilai TF dan nilai IDF untuk sebuah kata dalam sebuah dokumen (Lubis, 2021). TF-IDF menggabungkan informasi tentang frekuensi kata dalam dokumen tertentu (TF) dengan informasi tentang seberapa sering atau jarang sebuah kata tersebut muncul dalam korpus (IDF). Penggabungan nilai TF dan IDF terlihat seperti pada persamaan (2.3) berikut ini (Qaiser, 2018):

$$TF - IDF_{k,d} = TF_{k,d} * IDF_k \tag{2.3}$$

Berdasarkan persamaan (2.3), $TF\text{-}IDF_{k,d}$ akan memberikan bobot yang lebih tinggi pada kata-kata yang muncul lebih sering dalam dokumen tetapi lebih jarang muncul dalam keseluruhan korpus teks. Bobot ini membantu untuk mengidentifikasi kata-kata yang paling relevan dan informatif dalam dokumen, sehingga digunakan dalam banyak tugas pemrosesan teks seperti klasifikasi teks, pencarian informasi, dan rekomendasi konten.

Meskipun TF-IDF memberikan informasi yang relevan dan informatif, penggunaannya pada dataset teks dapat menghasilkan matriks yang sangat jarang atau memiliki banyak elemen bernilai nol (*sparsity*). *Sparsity* merujuk pada tingkat kekosongan dalam matriks TF-IDF, yang menyebabkan banyak elemen memiliki nilai nol. Hal ini terutama terjadi ketika terdapat banyak kata unik dalam korpus dan setiap dokumen hanya menggunakan sebagian kecil dari kata-kata unik tersebut. *Sparsity* dapat dihitung menggunakan persamaan (2.4) berikut ini (Witten, 2021):

$$S = 1 - \frac{elemen\ matriks\ bukan\ nol}{keseluruhan\ elemen\ matriks} \tag{2.4}$$

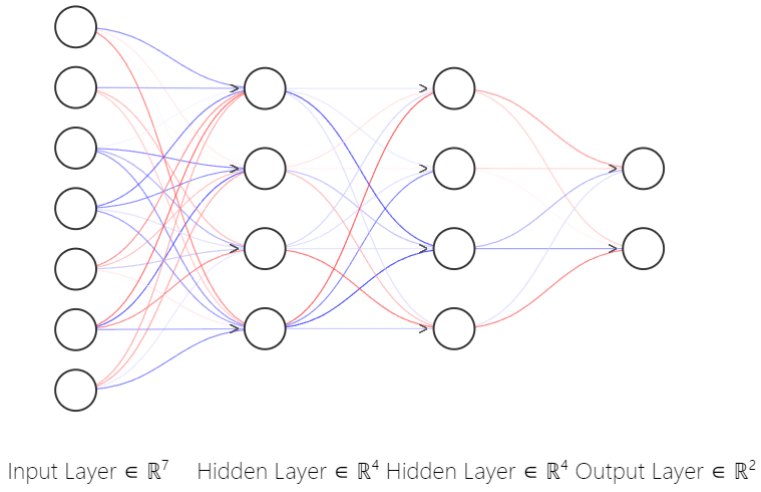
Sparsity pada matriks TF-IDF, mencerminkan seberapa banyak elemen matriks yang memiliki nilai nol, dapat memengaruhi kecepatan komputasi dan nilai akurasi model dalam pemrosesan teks (Mishra, 2021). Berfokus pada fitur-fitur yang memiliki kontribusi signifikan, memotong jumlah fitur yang kurang relevan tidak hanya mengurangi beban komputasi tetapi juga mengoptimalkan penggunaan penyimpanan (*computer memory*). Selain itu, keuntungan efisiensi penyimpanan yang dihasilkan oleh *sparsity* dapat menjadi krusial ketika berurusan dengan ketersediaan memori yang terbatas. Sparsity juga berperan sebagai bentuk seleksi alami dan memungkinkan untuk berfokus pada fitur-fitur yang lebih informatif. Hal yang perlu diperhatikan adalah pemilihan fitur perlu dilakukan dengan cermat seperti jumlah fitur dan persentase dari *sparsity* untuk memastikan kehilangan informasi yang minimal.

2.6 Neural Network

Neural network (NN) atau jaringan saraf merupakan salah satu cabang dari *machine learning* dan sekaligus fondasi dari perkembangan model seperti *deep learning*. Sementara *Machine learning* (ML) adalah pendekatan dalam bidang kecerdasan buatan yang mencoba menirukan bagaimana mahluk hidup belajar dan menggeneralisasi serta berjalan otomatis. ML memiliki dua aplikasi utama yaitu klasifikasi dan prediksi serta memiliki ciri khas yaitu diperlukannya proses pelatihan atau *training* (Retnoningsih, 2020). NN pertama dikembangkan pada bidang yang ditujukan untuk mempelajari atau memodelkan sistem saraf biologis. Kemudian perkembangan NN mengalami kemajuan dalam bidang teknik dan memiliki hasil yang baik dalam menyelesaikan tugas-tugas *machine learning* (Sewak, 2018). NN selanjutnya berkembang dari model sederhana (*Shallow NN*) hingga model *deep learning*. Berikut merupakan penjelasan lanjut mengenai model *shallow NN* dan *deep learning* tersebut.

2.6.1 Shallow Neural Networks

Shallow Neural Networks (SNN) sama dengan model NN dimana setiap koneksi antara memiliki bobot yang menentukan seberapa besar pengaruh dari *node* satu terhadap *node* lainnya. *Node* pada SNN disebut juga fitur atau variabel prediktor yang memiliki peran dalam mengidentifikasi variabel respon. Sama seperti NN, SNN terdiri dari tiga bagian atau *layer* utama yaitu *input layer*, *hidden layer*, dan *output layer*, namun SNN hanya memiliki *hidden layer* paling banyak 2 lapis. Karena model SNN adalah bagian dari arsitektur NN, pembahasan mengenai arsitektur SNN pada penelitian ini akan menggunakan istilah NN. Ilustrasi model sederhana model NN ditampilkan pada Gambar 2.1 sebagai berikut:



Gambar 2.1 Model Neural Network

Gambar 2.1 menunjukkan model NN terdiri dari tiga bagian utama yaitu *input layer*, dua *hidden layer*, dan *output layer*. Masing-masing fitur pada *input layer* akan dihitung nilai kontribusinya menggunakan bobot dan menjadi nilai kumulatif pada tiap *node* atau fitur di *hidden layer*. Nilai dari tiap bobot diilustrasikan sebagai benang berwarna merah dan biru. Benang yang lebih tebal menunjukkan bobot nilai yang lebih tinggi dan sebaliknya, sementara neuron dengan bobot positif ditunjukkan dengan benang biru kemudian negatif ditunjukkan sebagai benang merah. Proses ini dilakukan berulang hingga mencapai *layer* terakhir yaitu *output layer*. Adapun model matematis dari NN dapat ditulis pada persamaan 2.5 dan 2.6 berikut:

$$\mathbf{H} = \mathbf{X}\mathbf{W}^{(\mathbf{h})} + \mathbf{b}^{(\mathbf{h})} \tag{2.5}$$

$$\mathbf{O} = \mathbf{H}\mathbf{W}^{(\mathbf{o})} + \mathbf{b}^{(\mathbf{o})} \tag{2.6}$$

Keterangan:

H : Matriks pada *hidden layer*
O : Matriks pada *output layer*
X : Matriks pada *input layer*
$\mathbf{W}^{(h)}$: Matriks nilai bobot pada *hidden layer* ke *h*
$\mathbf{b}^{(\mathbf{h})}$: Vektor nilai bias pada *hidden layer* ke *h*
$\mathbf{W}^{(\mathbf{o})}$: Matriks nilai bobot pada *output layer*
$\mathbf{b}^{(\mathbf{o})}$: Vektor nilai bobot pada *output layer*

Pada umumnya model NN digunakan untuk mengolah *big data* dengan puluhan hingga ribuan fitur dan menghasilkan ribuan nilai. Jika nilai yang diperoleh selama proses pembobotan

memiliki *range* nilai yang besar akan menimbulkan beban kerja pada mesin dalam mengolah data. Untuk mengatasi hal tersebut digunakan fungsi non-linier untuk menekan (*squash*) ukuran dari fitur yang dihasilkan. Fungsi non-linier atau fungsi aktivasi terdapat beberapa jenis di antaranya *ReLU* dan *Sigmoid*. Berikut persamaan dari fungsi *ReLU* dan *sigmoid* yang ditunjukkan pada persamaan (2.7) dan (2.8):

$$ReLU(x) = max(x, 0) \tag{2.7}$$

$$sigmoid(x) = \frac{1}{1+\exp(-x)} \tag{2.8}$$

Rectified Linier Unit (ReLU) biasa ditempatkan dalam lapisan *hidden layer serta* memiliki fungsi memperkenalkan non-linier ke dalam model, yang memungkinkan NN memodelkan hubungan kompleks dalam data. Dengan menggunakan fungsi *ReLU* akan diperoleh nilai non-linier, yang kemudian dari nilai tersebut akan diubah diteruskan ke dalam *output layer*. Sementara fungsi sigmoid merupakan fungsi yang ditempatkan dalam *output layer* dan memiliki peran merubah nilai dari *hidden layer* ke dalam skala probabilitas dalam rentang (0,1). Nilai probabilitas selanjutnya digunakan untuk mengklasifikasikan dua kategori (hoaks dan fakta).

2.6.2 Deep Learning

Deep learning (DL) merupakan bagian dari *machine learning* (ML) yaitu, sebagai pembaruan dari *neural network multiple layer* yang merupakan salah satu algoritma ML (Yunanto, 2021). Proses pembelajaran DL didasarkan pada ekstraksi fitur secara otomatis dan hierarkis melalui (NN), tanpa memerlukan ekstraksi fitur manual. *Deep learning* memungkinkan komputer membangun konsep kompleks dari yang lebih sederhana misalnya, sistem pengenal citra suatu objek (Sewak, 2018). DL dapat ditunjukkan dengan adanya *layer* untuk mengolah dan mempelajari fitur dalam membuat pola sebagai fitur baru. *Layer* tersebut merupakan lapisan ekstraksi fitur dimana salah satu contohnya menggunakan metode *convolution.*

Sementara itu *feature extraction* sendiri adalah proses pengambilan informasi paling relevan atau fitur-fitur penting dari data yang kompleks. Perbedaan mendasar antara feature extraction dalam ML konvensional dan DL terletak pada tingkat otomatisasi dan kemampuan memahami representasi fitur yang lebih kompleks pada DL. Dalam ML tradisional, manusia cenderung harus secara manual menentukan dan mengekstrak fitur-fitur yang dianggap penting, sementara dalam DL, NN mampu secara otomatis mengidentifikasi dan mengekstrak fitur-fitur tersebut dari data. Dengan demikian, DL memungkinkan penciptaan model yang lebih adaptif dan mampu mengatasi tugas-tugas kompleks, seperti pengenalan citra objek, dengan tingkat akurasi yang lebih tinggi.

2.7 Convolution, Padding, dan *Stride*

Convolution atau konvolusi adalah operasi matematis yang bekerja pada dua fungsi untuk menciptakan fungsi baru yang menggambarkan pola atau karakteristik. Fungsi pertama dapat menggambarkan nilai sebuah data input dalam bentuk matriks atau vektor. Sedangkan fungsi kedua merupakan fungsi pembobotan yang disebut kernel atau filter yang juga dalam bentuk matriks atau vektor. Konvolusi akan menghasilkan fungsi baru yang menggambarkan fitur atau pola berdasarkan hasil pembobotan (Sugiarto, 2021). Adapun konvolusi bertujuan untuk mempelajari representasi fitur dari data input. Data input akan dikonvolusikan dengan filter dan akan membentuk berbagai fitur. Masing-masing fitur menjadi dasar pembeda data satu dengan

yang lain (Gu, 2018). Dalam bentuk matematis kontinu, konvolusi dari dua fungsi *f(t)* dan *g(t)* dapat dinyatakan sebagai persamaan (2.9):

$$C(x) = f(x) * g(x) = \sum_{x=-\infty}^{\infty} \sum_{x=-\infty}^{\infty} f(x)g(x-t) \quad (2.9)$$

Persamaan (2.9) menunjukan nilai *g(x)* merupakan filter konvolusi atau filter yang berfungsi sebagai pemberi bobot nilai pada fungsi *f(x).* Bobot nilai ditentukan secara acak dan jika suatu konvolusi memiliki lebih dari satu filter makan tiap filter akan menghasilkan bobot acak yang berbeda-beda. Hasil konvolusi C(*x*) merupakan hasil dari perkalian dan jumlah kedua fungsi pada setiap elemen matriks dan nilainya berbeda untuk tiap filter yang digunakan.

Pada data dalam format vektor atau matriks hasil konvolusi mempengaruhi ukuran akhir dimensi dari vektor atau matriks berdasarkan dimensi filternya. Perbedaan dimensi terjadi karena proses konvolusi merupakan fungsi berjalan dari kiri ke kanan dan tambahan dari atas ke bawah untuk matriks. Untuk mengetahui bagaimana dimensi akhir suatu vektor atau matriks yang dilakukan konvolusi diperoleh menggunakan persamaan (2.10):

$$(n_h - k_h + 1) \times (n_w - k_w + 1) \quad (2.10)$$

Keterangan:

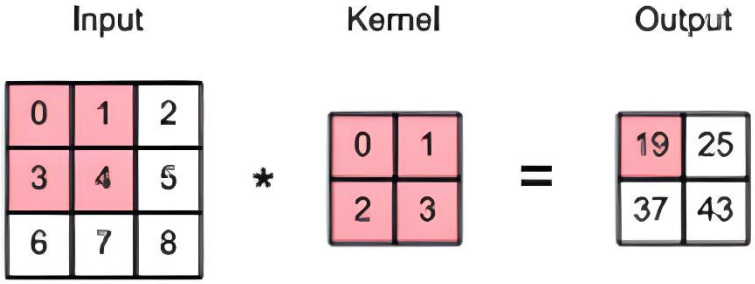
n_h **:** Jumlah baris matriks data input (dimensi tinggi)

n_w **:** Jumlah kolom matriks data input (dimensi panjang)

k_h **:** Jumlah baris matriks filter (dimensi tinggi)

k_w **:** Jumlah kolom matriks filter(dimensi panjang)

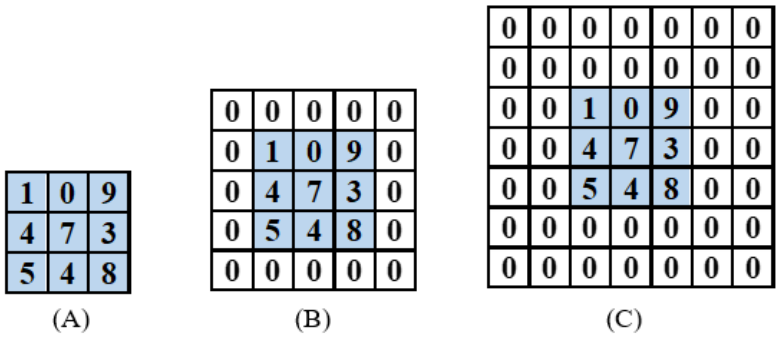
Menggunakan persamaan (2.10) dimensi matriks digambarkan dalam persamaan $n_h \times n_w$ dengan kernel berukuran $k_h \times k_w$, sementara vektor memiliki nilai n_h dan k_h sama dengan 1. Berikut ilustrasi dari konvolusi ditampilkan pada gambar 2.2.



Gambar 2.2 Proses *Convolution Layer*

Dalam operasi konvolusi (*), terdapat parameter penting seperti *padding* dan *stride*. *Padding* merujuk pada penambahan nilai nol di sekitar tepi matriks, dalam konteks ini, gambar sebagai data input atau peta fitur. Penambahan nilai nol ini bertujuan untuk menjaga dimensi akhir matriks gambar atau peta fitur saat mengalami proses konvolusi. Berikut ilustrasi matriks dengan penambahan *padding* ditampilkan Gambar 2.3:

Gambar 2.3 (A) merupakan contoh matriks asli sebelum mendapat penambahan *padding*. Kemudian matriks (B) menunjukkan matriks asli yang telah mendapatkan penambahan *padding* dengan nilai 1. Sementara matriks (C) merupakan matriks asli dengan penambahan *padding* dengan nilai 2. Parameter selanjutnya merupakan *stride*, *stride* menunjukkan seberapa jauh filter bergeser selama proses konvolusi (Assegaf, 2021). Stride dengan nilai satu akan menggerakkan filter satu kolom ketika mendatar dan satu baris ketika menurun selama proses konvolusi.



Gambar 2.3 *Padding* Pada Matriks

Proses penambahan *padding* dan *stride* tidak mengubah persamaan (2.10) jika digunakan secara *default*, yaitu ketika model memiliki *padding* nol dan *stride* satu. Namun, dengan menambahkan *padding* dan menggunakan *stride* yang berbeda, dimensi vektor atau matriks akan mengalami perubahan berdasarkan persamaan (2.11):

$$\left(\frac{(n_h - k_h + p_h + s_h)}{s_h}\right) \times \left(\frac{(n_w - k_w + p_w + s_w)}{s_w}\right) \tag{2.11}$$

Keterangan:

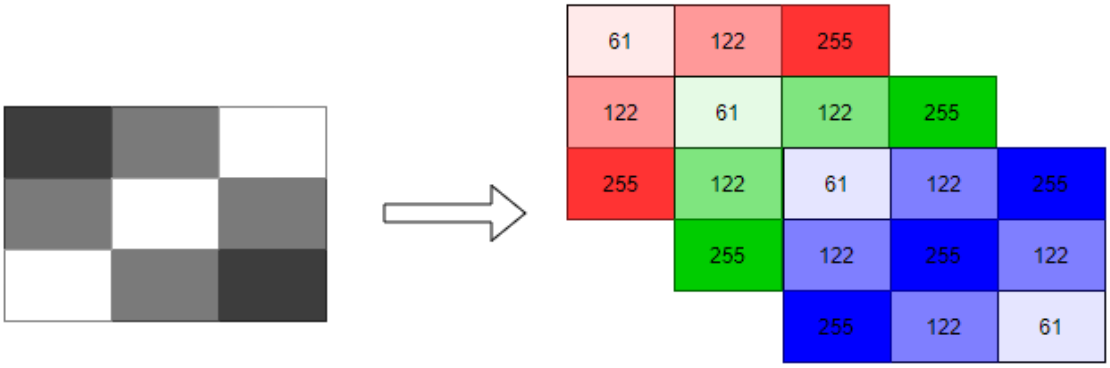
p_h **:** Jumlah padding pada dimensi tinggi matriks data input
p_w **:** Jumlah padding pada dimensi panjang matriks data input
s_h **:** Jumlah *stride* filter ketika bergerak vertikal
s_w **:** Jumlah *stride* filter ketika bergerak horizontal

2.8 Proses dan Bagian *Convolutional Neural Network*

Convolutional Neural Network (CNN) adalah arsitektur *deep learning* yang terkenal dalam pengolahan data gambar yang terinspirasi dari mekanisme persepsi visual alami makhluk hidup. CNN menggabungkan metode pengolahan pola citra menggunakan konvolusi sebagai ekstraksi fitur gambar dan model NN. Model CNN akan mengolah data *training* dan *testing* melewati tiga lapisan utama yaitu *input layer*, *hidden layer*, dan *output layer*. CNN pada bagian *hidden layers* memiliki banyak variasi, contohnya adalah arsitektur yang terdiri dari tiga jenis *layer* yaitu *convolutional layer*, *pooling layer*, dan *fully-connected layer* (Gu, 2018). Untuk mengetahui bagaimana peran beserta proses komputasi pada masing-masing *layer* CNN akan dijelaskan lebih rinci sebagai berikut:

2.8.1 Input Layer

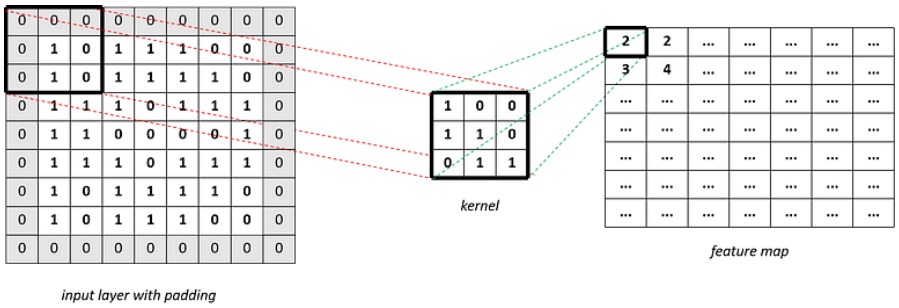
Input layer pada *Convolutional Neural Network* (CNN) menerima format data berupa *array* untuk mewakili yaitu panjang, tinggi, dan kedalaman data gambar. Dimensi panjang dan tinggi mencerminkan ukuran spasial gambar (resolusi), sementara dimensi kedalaman mengacu pada jumlah saluran warna (kanal). Penggunaan *array* memungkinkan CNN untuk mengatasi analisis citra karena citra digital memiliki dimensi spasial (lebar dan tinggi) serta dimensi saluran warna misalnya, merah, hijau, dan biru dalam model RGB. Gambar dengan variabel tinggi, panjang, saluran warna atau kanal jika disusun akan membentuk *array* 3D. Dalam konteks citra RGB seperti yang diilustrasikan dalam Gambar 2.4, setiap *pixel* akan diurai kedalam tiga buah kanal RGB masing-masing membentuk matriks dua dimensi dengan ukuran panjang dan lebar mengikuti gambar original. Selanjutnya masing-masing kanal akan mencerminkan nilai intensitas warna dalam interval 0 hingga 255 untuk tiap *pixel*.



Gambar 2.4 Matrix RGB

2.8.2 Convolution Layer

Convolution Layer merupakan komponen inti dari *Convolutional Neural Network* (CNN) dan bertanggung jawab untuk merepresentasikan pola dari fitur data input. *Layer* ini terdiri dari beberapa kernel (filter) dimana kernel memiliki dimensi dan di dalamnya berisikan nilai atau bobot. Setiap kernel akan melakukan operasi matematis pada data input untuk mendeteksi pola atau fitur tertentu. Semakin banyak kernel yang digunakan semakin banyak pola yang akan dihasilkan. Kemudian terdapat *receptive field* yang memiliki dimensi yang sama dengan kernel. *Receptive field* merupakan bidang atau elemen data input yang akan dilakukan operasi konvolusi dengan kernel. *Receptive field* merupakan representasi dari fungsi pertama *f(x)* sementara kernel merupakan representasi *g(x)* dan dari operasi tersebut diperoleh *feature map* representasi dari *C(x)*. Untuk menggambarkan *convolution layer* ditampilkan pada Gambar 2.4 (Yunus, 2023):



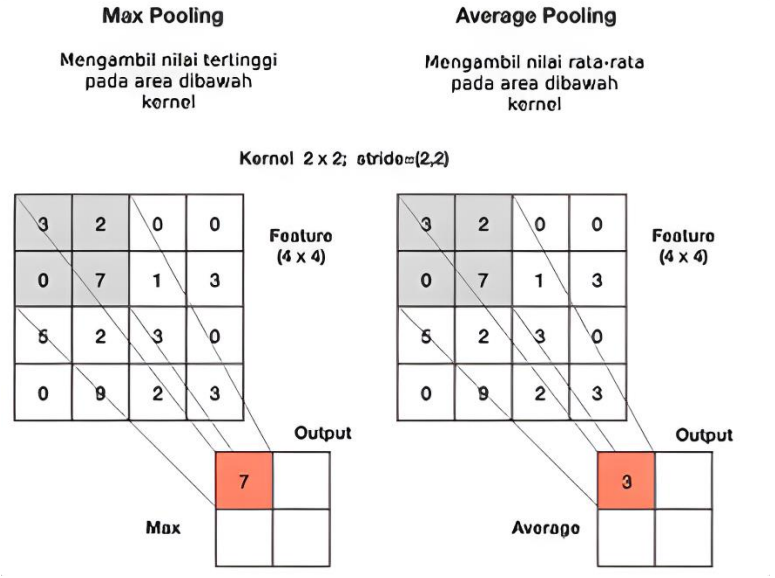
Gambar 2.5 Proses *Convolution Layer*

Ilustrasi pada Gambar 2.5 menunjukkan sebuah gambar berukuran 7 × 7 *pixel* akan dilakukan proses konvolusi menggunakan kernel berukuran 3 × 3 *pixel*. Gambar 2.4 juga menunjukkan penggunaan parameter *padding* dan *stride* sebesar satu. Kasus Gambar 2.4, kernel akan bergeser satu *pixel* pada tiap konvolusi dilakukan. Mengikuti ukuran kernel atau filter maka *receptive field* yang akan terbentuk sebesar 3 × 3 *pixel* beserta posisinya mengikuti kernel. Seluruh *receptive field* pada *input layer* akan digeser (*convolving*) dari kiri atas hingga ke kanan bawah, sehingga dihasilkan *feature map* (peta fitur) berukuran 7 × 7 *pixel*. Pada jenis gambar yang memiliki kanal atau saluran warna dilakukan proses konvolusi seperti sebelumnya untuk tiap kanalnya.

2.8.3 Pooling Layer

Pooling layer biasa ditempatkan di akhir *convolution layer* dan bertujuan untuk mengurangi resolusi (ukuran) dari peta fitur. *Pooling layer* juga menerapkan operasi konvolusi dan mirip dengan *convolution layer*, memiliki sebuah kernel, *padding*, dan *stride* yang

menentukan besarnya *receptive field* (Pan, 2021). Terdapat beberapa pendekatan dalam melakukan *pooling* untuk memperkecil resolusi peta fitur, di antara strategi yang umum digunakan adalah *max-pooling* dan *average-pooling* (de Souza Brito, 2021). Pada *max-pooling*, fungsi kernel atau *g(x)* akan mengambil setiap nilai *receptive field* atau *f(x)* dari peta fitur untuk diambil nilai tertinggi, sementara *average-pooling* akan mengambil nilai rata-rata dari peta fitur. Ilustrasi proses ini diilustrasikan pada Gambar 2.6 (Qayyum, 2023):



Gambar 2.6 Proses *Pooling Layer*

Ilustrasi pada Gambar 2.6 menunjukkan sebuah gambar berukuran berukuran 4 × 4 *pixel* akan dilakukan proses *pooling* menggunakan kernel berukuran 2 × 2 pixel. Berbeda dengan contoh pada Gambar 2.5 sebelumnya, penggunaan parameter *padding* bernilai nol (tidak ada *padding*) dan *stride* sebesar dua. Selanjutnya proses pooling yang terjadi yaitu *receptive field* pada input layer akan digeser sebesar dua *pixel* dari kiri atas hingga ke kanan bawah, sehingga dihasilkan *feature map* (peta fitur) berukuran 2 × 2 pixel.

2.8.4 Flatten Layer

Flatten layer pada CNN berperan penting dalam menyesuaikan representasi data hasil *pooling layer* sebelumnya untuk persiapan menuju *fully connected layer*. Setelah proses *pooling*, data tersebut masih berbentuk matriks atau tensor multidimensional yang merepresentasikan fitur-fitur yang telah diekstraksi dari input. *Flatten layer* kemudian ditempatkan setelah *pooling layer* untuk mengubah bentuk data menjadi vektor satu dimensi. Dengan demikian, *flatten layer* memastikan bahwa output dari *layer* sebelumnya dapat diubah menjadi format yang dapat diterima oleh *fully connected layer* (Ram, 2020). Ini memungkinkan informasi spasial yang telah diambil dari data input melalui proses *pooling* untuk diolah lebih lanjut dalam *fully connected layer*, yang umumnya digunakan untuk pengambilan keputusan atau klasifikasi akhir dalam tugas seperti pengenalan pola atau klasifikasi gambar. Dengan adanya *flatten layer*, integrasi antara *layer* konvolusi, *pooling, dan fully connected* dapat dilakukan dalam arsitektur CNN.

2.8.5 Fully-Connected Layer

Fully-connected layer (FCL) dikenal juga dengan sebutan "*dense layer*" karena setiap neuron dalam *layer* ini terhubung dengan semua neuron pada *layer* sebelumnya. Bagian ini merupakan tempat dimana setiap fitur yang sudah diolah pada tahap konvolusi dilakukan

pembelajaran menggunakan metode NN. Setiap fitur dalam bentuk *array* hasil *feature extraction* akan dilakukan vektorisasi dan menjadi *node* untuk *input layer* FCL. Sama seperti metode NN, pembobotan dilakukan pada variabel **X** (fitur) yang merupakan *layer* terakhir *feature extraction)***,** kemudian memasuki *hidden layer* FCL (**H**) dan dilakukan pembobotan **(W)** hingga mencapai *layer* terakhir yaitu *output layer* (**O**). Pada output akan diperoleh nilai probit yang merepresentasikan prediksi fitur tertentu terhadap label (Ram, 2020).

2.8.6 Block

CNN pada bagian *feature extraction* yaitu konvolusi, memiliki banyak jenis *layer* dengan spesifikasi ukuran dan jumlah *filter* yang bervariasi di tiap *layer*. *Layer* ini dapat disusun dan membentuk pola yang disebut *block*, dimana penyusun dasar suatu *block* adalah rangkaian *layer* pada Tabel 2.9:

Tabel 2.9 Contoh *Block* Sederhana

Bagian	Jenis *Layer*	Keterangan Spesifikasi *Layer*
1	*Convolutional Layer*	Dalam satu *block terdapat* dua atau lebih *layer* konvolusi dengan jumlah dan jenis *filter*, *stride,* dan *padding* yang sama
2	*Non-linier Layer*	Fungsi non-linier memungkinkan model untuk mempelajari dan menangkap pola-pola yang lebih kompleks dalam fitur. Fungsi non-linier dapat diterapkan pada keseluruhan c*onvolutional layer* atau hanya diletakkan di bagian akhir *convolutional layer.* Adapun *layer* fungsi non-linier yang sering digunkan yaitu ReLU.
3	*Pooling Layer*	*Pooling layer* membantu mengurangi dimensi dan fokus pada fitur-fitur yang paling dominan. Pada umumnya setiap *block* menggunakan jenis *pooling layer* yang sama untuk keseluruhan *block* dalam suatu arsitektur atau model. Jenis-jenis p*ooling layer* yaitu *max-pooling*, *average-pooling*, *global-pooling,* dan lain-lain.

2.9 VGG dan Arsitektur Model CNN

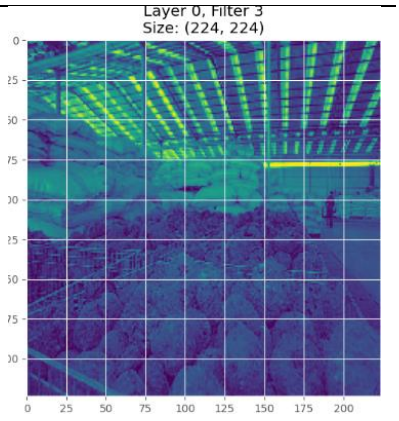
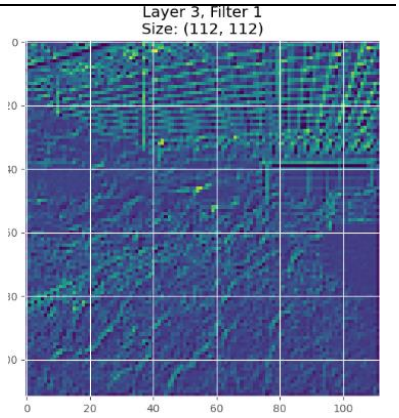
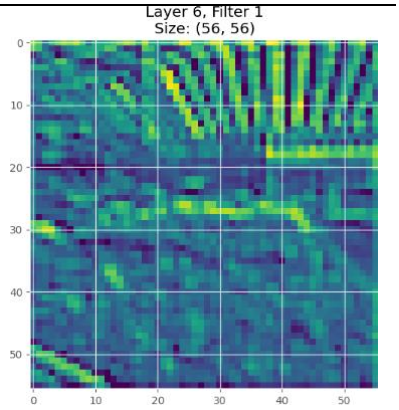
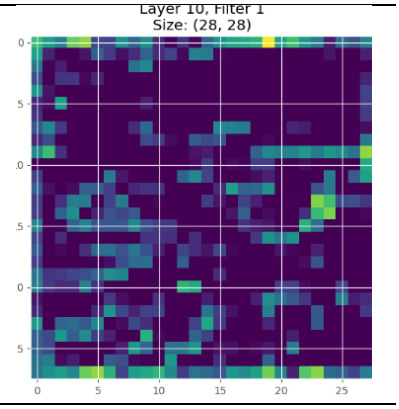
Arsitektur VGG merupakan salah satu terobosan dalam pengembangan CNN yang dikembangkan oleh *Visual Geometry Group* (VGG) di Universitas Oxford. Pertama kali diperkenalkan dalam makalah berjudul "Very Deep Convolutional Networks for Large-Scale Image Recognition," yang disajikan pada Konferensi Pengenalan Pola dan Pengenalan Objek (*ImageNet*) tahun 2014, VGG dikenal karena pendekatannya yang sederhana dan efektif dalam mendesain jaringan yang dalam serta mampu memahami fitur-fitur kompleks pada data visual.

VGG telah memberikan kontribusi besar dalam pengembangan CNN dengan memopulerkan penggunaan *block* konvolusi yang terdiri dari *layer* konvolusi dengan kernel 3 × 3 dan *padding* 1, diikuti oleh *max-pooling layer* 2 × 2 dengan nilai *stride* 2. Pendekatan ini memungkinkan VGG untuk menjaga informasi spasial sambil secara progresif mengurangi dimensi tinggi dan lebar setiap kali *block* diterapkan (Zhang, 2021). Berikut merupakan contoh arsitektur Model CNN berbasis gambar (mengolah data 2D atau matriks) dari VGG ditampilkan pada Tabel 2.10:

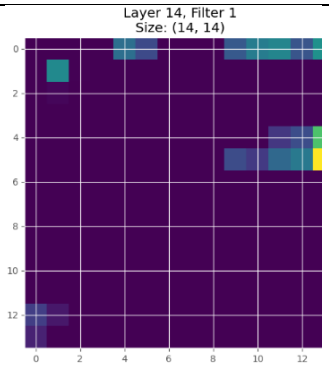
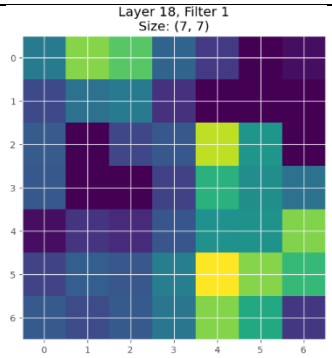
Table 2.10 menunjukkan *layer* pada arsitektur model gambar menggunakan *block*. Pada model terdapat lima buah *block* yang merepresentasikan proses *feature extraction* dalam mengolah data gambar foto berita. Berikut merupakan contoh tahapan proses yang berlangsung pada data sebuah data *training* atau *testing* yang masuk kedalam model:

1. *Input layer* pada arsitektur ini ditunjukkan oleh nomer 0, secara matematis dinotasikan sebagai matriks $\boldsymbol{X_1} \in \mathbb{R}^{224\times224}$ untuk setiap *n* data observasi yang masuk kedalam model.

Tabel 2.10 Contoh Arsitektur Model CNN Berbasis Gambar dari VGG

No.	*Layer*	Dimensi	*Block*	*Block Output*
0	*Input_1(3)*	$\mathbb{R}^{224\times224}$	**0**	
1	*Conv2D(64)*	$\mathbb{R}^{224\times224}$	*1*	
2	*Conv2D(64)*	$\mathbb{R}^{224\times224}$		
3	*Maxpooling2D*	$\mathbb{R}^{112\times112}$		
4	*Conv2D(128)*	$\mathbb{R}^{112\times112}$	*2*	
5	*Conv2D(128)*	$\mathbb{R}^{112\times112}$		
6	*Maxpooling2D*	$\mathbb{R}^{56\times56}$		
7	*Conv2D(256)*	$\mathbb{R}^{56\times56}$	*3*	
8	*Conv2D(256)*	$\mathbb{R}^{56\times56}$		
9	*Conv2D(256)*	$\mathbb{R}^{56\times56}$		
10	*Maxpooling2D*	$\mathbb{R}^{28\times28}$		

Tabel 2.10 Contoh Arsitektur Model CNN Berbasis Gambar dari VGG (Lanjutan)

No.	*Layer*	Dimensi	*Block*	*Block Output*
11	*Conv2D(512)*	$\mathbb{R}^{28\times28}$	*4*	
12	*Conv2D(512)*	$\mathbb{R}^{28\times28}$		
13	*Conv2D(512)*	$\mathbb{R}^{28\times28}$		
14	*Maxpooling2D*	$\mathbb{R}^{14\times14}$		
15	*Conv2D(512)*	$\mathbb{R}^{14\times14}$	*5*	
16	*Conv2D(512)*	$\mathbb{R}^{14\times14}$		
17	*Conv2D(512)*	$\mathbb{R}^{14\times14}$		
18	*Maxpooling2D*	$\mathbb{R}^{7\times7}$		
19	*Flatten Layer*	$\mathbb{R}^{1\times(7\times7\times512)}$		[0, 0, 0, ..., 0.03490932, 0, 0.4599033,]
20	*FCL(512)*	$\mathbb{R}^{1\times512}$	-	[0, 0, 0, ..., 0.01146087, 0, 0,]
21	*FCL(512)*	$\mathbb{R}^{1\times512}$	-	[0, 0, 0, ..., 0, 0.01891793, 0,]
22	*Output(1)*	$\mathbb{R}^{1\times1}$	-	[0.44713253]

2. Selanjutnya dilakukan tahap *features extraction* terhadap $\boldsymbol{X_1}$ melewati lima buah *block* dimana setiap *block* menggunakan jumlah filter 64, 128, 256, 512, dan 512 serta ukuran kernel 3×3 dan *stride* sebesar 1. Selama proses *feature extraction* dimensi $\boldsymbol{X_1}$ akan mengecil yang dapat dihitung menggunakan persamaan (2.8). Pada Tabel 2.10 ditunjukkan dimensi setiap *layer* beserta output dan visualisasi fitur yang berhasil diekstrak dari masing-masing *max-pooling* di setiap *block*.
3. Pada *flatten layer* fitur yang berhasil diekstrak dari *block* terakhir pada tahap *max-pooling* di setiap *filter* akan diubah kedalam bentuk vektor. vektor $\boldsymbol{X_1}$ pada setiap filter pada *block* terakhir *(block* 5 dengan jumlah 512 filter*)* dilakukan *flatten* untuk diubah kedalam bentuk vektor $\boldsymbol{x_{1\,fcl}} \in \mathbb{R}^{1\times(7\times7\times512=25088)}$. vektor $\boldsymbol{x_{1\,fcl}}$ yang telah terbentuk selanjutnya diteruskan menuju FCL.
4. Memasuki FCL, dimana dan dilakukan pembobotan mengikuti persamaan (2.1). Pada FCL tahap pertama (h1), pembobotan dilakukan untuk membentuk 512 node atau fitur. Pembobotan vektor $\boldsymbol{x_{1\,fcl}}$ dilakukan dengan bobot $\boldsymbol{W_1} \in \mathbb{R}^{25088\times512}$, sehingga diperoleh $\boldsymbol{h_1} \in \mathbb{R}^{1\times512}$ atau sama dengan 512 fitur.
5. Proses dilakukan berulang pada FCL tahap dua (h2) untuk memperoleh 512 node atau fitur. Pada FCL tahap kedua (h2), pembobotan dilakukan untuk membentuk 512 node atau fitur. Pembobotan vektor $\boldsymbol{h_1}$ dilakukan dengan bobot $\boldsymbol{W_2} \in \mathbb{R}^{512\times512}$, sehingga diperoleh $\boldsymbol{h_2} \in \mathbb{R}^{1\times512}$ dengan 512 fitur. Selain itu untuk setiap hasil dari proses FCL merupakan nilai non-linier dari fungsi ReLU.

6. Terakhir merupakan tahap output dengan merubah nilai non-linier pada $\boldsymbol{h_2}$ diubah menjadi nilai probabilitas sebagai dasar memprediksi label menggunakan persamaan (2.2). Pada *output layer* dilakukan pembobotan kembali pada $\boldsymbol{h_2}$ dengan bobot $\boldsymbol{W_{out}} \in \mathbb{R}^{\mathbf{512\times1}}$sehingga diperoleh vektor $\boldsymbol{o} \in \mathbb{R}^{\mathbf{1\times1}}$ dengan satu fitur. Hasil dari **o** akan memasuki fungsi *sigmoid* dan menghasilkan nilai non-linier dengan skala 0 hingga 1. Rentang tersebut dalam konteks probabilitas digunakan sebagai penunjuk dalam klasifikasi biner.

Proses transformasi menjadi nilai probabilitas memberi kesempatan dilakukannya analisis statistik. Dengan memanfaatkan fungsi *sigmoid* pada *output layer*, nilai probabilitas yang dihasilkan dapat diinterpretasikan sebagai probabilitas suatu kelas positif atau negatif (misalnya berita fakta). Dalam konteks ini, nilai probabilitas yang lebih kecil dari 0.5 menunjukkan prediksi berita sebagai kelas negatif (misalnya berita fakta), sementara nilai probit yang lebih besar atau sama dengan 0.5 mengindikasikan prediksi berita sebagai kelas positif.

Hasil klasifikasi model CNN berbasis gambar yang dicontohkan pada Tabel 2.10 nomer 22 menunjukkan berita dikategorikan fakta. Hasil klasifikasi menjelaskan indeks pertama dalam data *training* atau testing memiliki nilai probabilitas (*output*) sebesar 0.447, dimana probabilitas tersebut lebih kecil dari 0.5, maka hasil prediksi dapat dikategorikan sebagai label dengan nilai 0, yang menandakan berita dikategorikan fakta. Dengan demikian, proses perubahan nilai non-linier *output layer* menjadi nilai probabilitas, menjadi dasar penentuan klasifikasi dalam interpretasi hasil prediksi model.

Kemampuan arsitektur dari VGG dalam menjaga penurunan resolusi spasial pada data gambar sangat baik. Hal ini dapat dimanfaatkan untuk membangun arsitektur model CNN berbasis teks (mengolah data 1D atau vektor) dalam mengolah data teks suatu berita. Berikut contoh arsitektur sederhana dari model CNN berbasis teks ditampilkan pada Tabel 2.11:

Tabel 2.11 menggambarkan model yang terdiri dari tiga *block* yang merepresentasikan proses ekstraksi fitur pada data teks, serta penggunaan NN dengan *Fully Connected Layers* (FCL) dan Output. Secara garis besar, tahapan yang terjadi pada model CNN berbasis teks adalah serupa. Perbedaan antar model terletak pada ukuran dimensi data input, ukuran filter, dan jumlah filter. *Block* pada contoh di atas terdiri dari *layer* konvolusi dengan filter 1 × 3, *padding* 0, dan *stride* 3, diikuti oleh *max-pooling layer* 1 × 2 dengan nilai *stride* 1. Spesifikasi

Tabel 2.11 Contoh Arsitektur Model CNN Berbasis Teks

No.	*Layer*	Dimensi	Block	*Block Output*
0	*Input_1(2000,1)*	$\mathbb{R}^{\mathbf{1\times8900}}$	0	[0, 0, 0, ..., 0, 0, 0,]
1	*Conv1D(64)*	$\mathbb{R}^{\mathbf{1\times4449}}$	1	
2	*Conv1D(64)*	$\mathbb{R}^{\mathbf{1\times2224}}$		[0.0010103, 0.0010103, 0.0010103, ..., 0.0010103, 0.0010103, 0.0010103]
3	*Maxpooling1D*	$\mathbb{R}^{\mathbf{1\times1112}}$		
4	*Conv1D(128)*	$\mathbb{R}^{\mathbf{1\times555}}$	2	
5	*Conv1D(128)*	$\mathbb{R}^{\mathbf{1\times277}}$		
6	*Maxpooling1D*	$\mathbb{R}^{\mathbf{1\times138}}$		[0, 0, 0, ..., 0, 0, 0,]
8	*Conv1D(512)*	$\mathbb{R}^{\mathbf{1\times33}}$		
9	*Conv1D(512)*	$\mathbb{R}^{\mathbf{1\times16}}$		
10	*Maxpooling1D*	$\mathbb{R}^{\mathbf{1\times8}}$		[0, 0, 0, ..., 0, 0, 0,]
11	*FCL(512)*	$\mathbb{R}^{\mathbf{1\times512}}$	-	[0, 0, 0, ..., 0.01146087, 0, 0,]
12	*FCL(512)*	$\mathbb{R}^{\mathbf{1\times512}}$	-	[0, 0, 0, ..., 0, 0.01891793, 0,]
13	*Output(1)*	$\mathbb{R}^{\mathbf{1\times1}}$	-	[0.44713253]

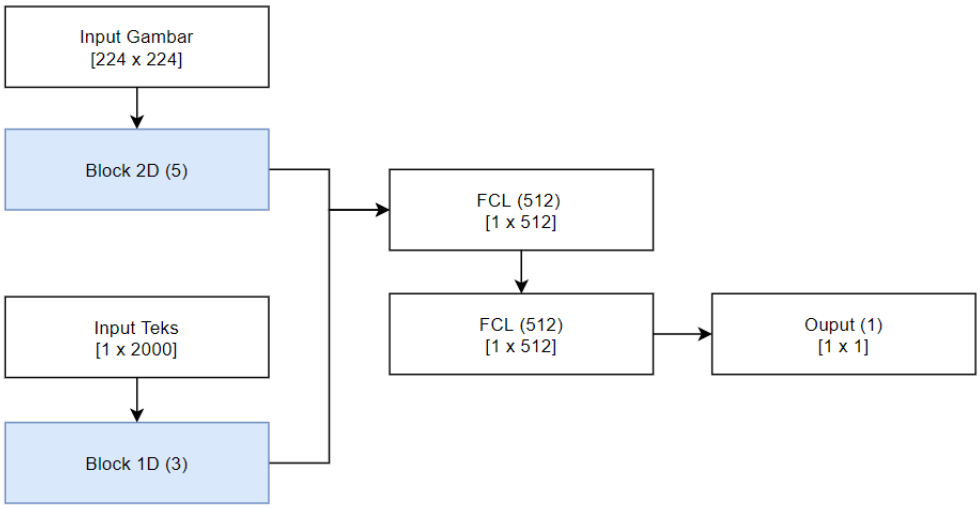
di atas pada gilirannya memengaruhi output dari tiap *layer*. Sebagai contoh, pada Tabel 2.11, output ditampilkan dalam kolom-kolom yang mewakili tiap *block*.

Berdasarkan penjelasan arsitektur CNN di atas menunjukkan model CNN dapat digunakan untuk mengolah data berbasis gambar dan teks. Pada CNN berbasis gambar data yang diolah model berupa matriks sehingga disebut juga model CNN 2D. Sedangkan model CNN berbasis teks mengolah data teks hasil TF-IDF, dalam bentuk vektor, sehingga disebut model CNN 1D. Kedua arsitektur CNN 1D dan 2D disusun berdasarkan model CNN dari VGG sehingga memungkinkan untuk dilakukan penggabungan dua model yang disebut *multimodel* atau model CNN gabungan.

2.10 Multimodel Deep Learning

Multimodel deep learning merupakan model gabungan dua atau lebih model *deep learning*. *Multimodel deep learning* menjadi bidang yang semakin populer karena kemampuannya untuk mengintegrasikan data *heterogen* dalam satu arsitektur yang efektif (Kim, 2018). Penerapan *multimodel deep learning* telah menunjukkan hasil yang baik dalam berbagai permasalahan, termasuk klasifikasi gambar objek yang memiliki kemiripan tinggi dengan menggabungkan informasi dari gambar dan teks. *Multimodel deep learning* dapat meningkatkan akurasi dan interpretabilitas dari model (Zhou, 2021). Secara keseluruhan, *multimodel deep learning* telah membuka pintu bagi penggabungan berbagai jenis data untuk meningkatkan kinerja model dan memberikan wawasan yang lebih jelas tentang bagaimana *model deep learning* membuat keputusan.

Dalam konteks *multimodel deep learning* yang digunakan dalam penelitian ini merupakan model yang mengabungkan CNN berbasis gambar dan teks. Penggabungan model dilakukan dengan memanfaatkan fitur ekstraksi dari model CNN berbasis gambar dan teks yang terletak pada masing-masing *block*, kemudian menggabungkan *output* dari kedua *block*. Berikut contoh arsitektur *multimodel* atau model CNN gabungan yang diilustrasikan pada Gambar 2.7,



Gambar 2.7 Contoh Arsitektur Model CNN Kombinasi

Gambar 2.7 menampilkan *layer* pada arsitektur model yang merupakan kombinasi teks dan gambar, menggunakan CNN *block* satu dimensi sebagaimana terdefinisi pada Tabel 2.10, dan *block* dua dimensi yang mencerminkan arsitektur model berbasis gambar dari VGG pada Tabel 2.9. Menggunakan contoh arsitektur model pada Tabel 2.10 dan 2.9, terdapat dua input utama, yaitu gambar foto berita dan teks, masing-masing melibatkan lima dan tiga *block* untuk melakukan proses ekstraksi fitur. Arsitektur di atas akan disebut sebagai model CNN gabungan dan mewakili *multimodel*.

Namun, untuk pemahaman lebih mendalam, perlu dicatat perbedaan antara *unimodel* dan *multimodel*. *Unimodel* mengacu pada model yang dibangun untuk menangani satu jenis data atau modality, seperti hanya gambar saja atau teks saja. Sebaliknya, *multimodel* menggabungkan beberapa jenis data atau modalitas dalam satu model. Dalam konteks ini, model CNN gabungan diperkenalkan sebagai representasi dari *multimodel*, menggabungkan fitur-fitur ekstraksi dari gambar dan teks untuk meningkatkan kinerja klasifikasi. Perbandingan antara blok satu dimensi dan dua dimensi pada CNN gabungan menunjukkan fleksibilitas dalam menyesuaikan jumlah filter dan kompleksitas fitur sesuai dengan kebutuhan tugas.

2.11 Parameter Pendukung Model *Deep Learning Dalam CNN*

Deep learning, sebagai sub-bidang dari ML, telah mengalami kemajuan pesat dan menjadi kunci utama dalam mengatasi masalah kompleks di berbagai bidang. Dalam pengembangan model DL, pemahaman terhadap parameter-parameter pendukung sangat penting untuk mencapai kinerja yang optimal. Parameter-parameter tersebut, seperti *epoch, batch size, optimizer, loss function,* dan *early stopping*, memiliki peran masing-masing dalam membentuk dan melatih model. Dalam, akan dipelajari secara mendalam mengenai aspek-aspek kritis dari setiap parameter tersebut. Dengan pemahaman yang mendalam terhadap parameter-parameter ini, diharapkan mampu membantu peneliti dan praktisi dalam mengoptimalkan model deep learning untuk tugas-tugas yang beragam. Langsung masuk ke dalam sub-bab, kita akan menjelajahi setiap parameter dan implikasinya pada pembangunan dan pelatihan model *deep learning*.

2.11.1 *Epoch* dan *Early Stopping*

Epoch adalah parameter yang menentukan seberapa banyak seluruh dataset akan digunakan pada saat pelatihan model. Satu *epoch* terjadi ketika seluruh dataset telah dilewati/diproses satu kali oleh model. Jumlah *epoch* yang optimal dapat mempengaruhi kinerja model, dan terlalu sedikit atau terlalu banyak *epoch* dapat menyebabkan *overfitting* atau *underfitting.*

Sementara *Early stopping* adalah teknik yang digunakan untuk menghentikan pelatihan model lebih awal jika tidak terjadi peningkatan kinerja. Hal ini dapat membantu mencegah *overfitting* dan menghemat waktu pelatihan. Penghentian dini dilakukan dengan memonitor metrik performa seperti akurasi, *loss*, MAE, dan lain-lain pada *training* dan *testing*, dimana pelatihan dihentikan ketika performa tidak meningkat selama beberapa *epoch* (Zhang, 2021).

2.11.2 *Batch Size*

Batch size adalah jumlah sampel data yang digunakan dalam satu iterasi pelatihan. Memilih *batch size* yang sesuai dapat memengaruhi kecepatan pelatihan dan penggunaan memori. *Batch size* besar dapat meningkatkan kecepatan pelatihan tetapi menggunakan lebih banyak memori, sedangkan *batch size* kecil dapat memberikan hasil yang lebih stabil tetapi memerlukan waktu pelatihan yang lebih lama. Pemilihan *batch size* pada umumnya dengan memilih salah satu kelipatan 4 seperti 4, 8, 16, 32, dan seterusnya. Nilai *batch size* yang sudah ditetapkan akan digunakan membagi total dataset dan diperoleh nilai iterasi atau disebut *minibatch*. Ketika semua *minibatch* telah diproses maka akan dihitung satu *epoch* (Zhang, 2021).

2.11.3 Loss Function

Fungsi kerugian (*loss function*) dalam domain DL memainkan peran krusial yang sangat terkait dengan jenis tugas yang diberikan kepada model. Dalam konteks tugas regresi, *Mean Squared Error* (MSE) diterapkan sebagai metrik statistik yang mengukur disparitas kuadrat

antara nilai prediksi dan nilai sebenarnya. Sementara itu, untuk skenario klasifikasi binomial, pilihan umum jatuh pada *Binary Cross-Entropy Loss* yang secara efisien mengevaluasi dissonansi distribusi probabilitas prediksi terhadap distribusi probabilitas target. Demikian pula, dalam tugas klasifikasi multinomial dengan lebih dari dua kategori, *Categorical Cross-Entropy Loss* digunakan untuk menilai kesesuaian distribusi probabilitas prediksi dengan target. Pemilihan dengan cermat terhadap *loss function* menjadi suatu aspek yang sangat strategis, sebab pengaruhnya yang signifikan terhadap konvergensi model dan hasil akhirnya (Zhang, 2021).

2.11.4 Optimizer

Optimizer atau algoritma optimisasi dalam konteks DL menjadi elemen esensial yang mengarahkan model ke parameter optimal. Salah satu *optimizer* yang umum digunakan adalah *Stochastic Gradient Descent* (SGD), yang secara statistik dapat diartikan sebagai pendekatan stokastik yang meminimalkan fungsi kerugian dengan mengiterasikan dataset melalui subset acak pada setiap iterasi. Peningkatan dari SGD adalah Adam (*Adaptive Moment Estimation*), yang memanfaatkan momentum gradien dan varians gradien untuk mencapai adaptabilitas yang lebih baik. Secara statistik, Adam diinterpretasikan sebagai pendekatan yang mempertimbangkan perubahan adaptif dari gradien dan sejarahnya (fungsi yang terbentuk selama pelatihan) (Zhang, 2021).

2.12 Evaluasi Model

Untuk mengevaluasi akurasi *multimodel deep learning* dalam klasifikasi berita hoaks dan fakta, digunakan *confusion matrix* untuk memetakan hasil prediksi terhadap keadaan aktual (Ridwan, 2020). *Confusion matrix* adalah sebuah tabel yang digunakan untuk mengevaluasi kinerja suatu model klasifikasi dengan membandingkan empat kemungkinan hasil dari proses klasifikasi yaitu: *true positive* (TP), *false positive* (FP), *true negative* (TN), dan *false negative* (FN). TP mengacu pada jumlah individu yang sebenarnya berada dalam kategori positif dan diprediksi dengan benar; FP mengacu pada jumlah individu yang tidak termasuk dalam kategori positif tetapi salah diklasifikasikan sebagai kategori positif; TN mengacu pada jumlah individu yang tidak termasuk dalam kategori positif dan diprediksi dengan benar; FN mengacu pada jumlah individu yang sebenarnya berada dalam kategori positif tetapi salah diklasifikasikan (Zhou, 2021). Dalam penelitian ini kelas positif merupakan berita dengan label hoaks, sementara berita dengan label bukan hoaks merupakan kelas negatif.

Proses evaluasi dimulai dengan membagi dataset menjadi data *training* dan data *testing*. Data *training* dan *testing* akan dibandingkan dengan model klasifikasi teks saja, gambar saja, dan model gabungan (*multimodel*) menggunakan arsitektur yang sama, sementara aspek yang dinilai yaitu akurasi, AUC dari ROC. Akurasi adalah perbandingan jumlah prediksi yang benar. Akurasi menunjukkan bagaimana model mampu memprediksi berita hoaks secara tepat. Rumus akurasi dapat dilihat dari persamaan (2.12):

$$accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{2.12}$$

Kemudian, untuk mengevaluasi kinerja ketiga model klasifikasi berita hoaks dan fakta, digunakan metode *receiver operating characteristics* (ROC) dan *area under the curve* (AUC). ROC adalah kurva probabilitas yang memvisualisasikan performa model dalam memisahkan antara kelas positif (TP) dan negatif (FP) dengan variasi *threshold*, sehingga merepresentasikan kemampuan model dalam memprediksi data di luar dataset. Sementara itu, AUC merupakan

ukuran dari tingkat pemisahan antara kedua kelas tersebut, dengan nilai berkisar dari 0 hingga 1. Sebuah model yang memiliki AUC mendekati 1 dianggap sangat baik, menunjukkan kemampuan yang baik dalam memisahkan kelas-kelas tersebut. Sebaliknya, jika nilai AUC mendekati 0, model tersebut dianggap buruk dalam memisahkan kelas. Jika nilai AUC sekitar 0,5, itu menunjukkan bahwa model tidak memiliki kemampuan dalam memisahkan kelas. Oleh karena itu, nilai AUC merupakan indikator penting dalam mengevaluasi kualitas sebuah model klasifikasi (Nugroho, 2021).

2.13 Hoaks

Hoaks merujuk pada kabar, berita, informasi palsu atau bohong yang sengaja disebarkan untuk mencapai tujuan tertentu. Terdapat beberapa alasan atau faktor penyebab munculnya berita hoaks, yaitu sebagai hiburan, meramaikan perkembangan tren di internet, mencari sensasi, dan pekerjaan untuk menyudutkan pihak tertentu (Mazaya, 2019). Sebuah informasi hoaks pada umumnya tersebar dalam bentuk terstruktur mengikuti format berita pada umumnya, dimana sebuah berita terdiri memiliki headline, foto berita peristiwa, dan konten berita.

2.14 *Turnbackhoax.co.id* dan *Republika.co.id*

Turnbackhoax.co.id dan *republika.co.id* merupakan dua situs web yang memiliki peran yang berbeda dalam memberikan informasi kepada pembaca. Pertama, *turnbackhoax.co.id* merupakan situs web yang didirikan oleh kelompok Masyarakat Anti Fitnah Indonesia (Mafindo) yang secara khusus berdedikasi untuk mengidentifikasi dan membantah informasi palsu atau hoaks. Situs ini berfungsi sebagai pengecek fakta, memberikan fasilitas bagi anggota komunitas bertukar informasi dan memperoleh pemahaman yang lebih baik mengenai berita yang mungkin tidak akurat atau menyesatkan. Dengan pendekatan yang berfokus pada pembantahan hoaks, *turnbackhoax.co.id* memainkan peran penting dalam mitigasi penyebaran informasi yang dapat merugikan (Haqqo, 2023).

Di sisi lain, *Republika.co.id*, sebuah portal berita daring yang terkemuka di Indonesia, menempati peringkat 34 dari 50 portal berita nasional. Situs ini sering membingkai berita secara netral, tanpa memihak kepada pihak manapun, dengan tujuan menyajikan informasi secara menyeluruh dan berdasarkan fakta yang sebenarnya. *Republika.co.id* juga diakui karena kemampuannya dalam menggunakan bahasa yang sopan dan bijaksana dalam menyampaikan berita, menjaga profesionalitasnya sebagai media massa (Zaen, 2023). Peran Republika.co.id sebagai sumber berita nasional sangat penting, membantu pembaca untuk tetap terinformasi tentang perkembangan berita, baik di dalam maupun di luar negeri. Dengan cakupan berita yang luas, *Republika.co.id* telah menjadi salah satu sumber utama bagi masyarakat Indonesia yang mencari informasi yang aktual dan dapat dipercaya.

BAB III
METODOLOGI

3.1 Sumber Data

Data yang digunakan pada penelitian ini merupakan data sekunder yang diperoleh dari website *turnbackhoax.id*. sebagai sumber berita hoaks dan *republika.co.id* sebagai sumber berita fakta. Data diambil menggunakan proses *web scraping* pada berita hoaks dan fakta selama periode publikasi tanggal 1 Januari 2023 hingga 13 Oktober 2023. Waktu pengambilan data dilakukan pada hari Kamis, 13 Oktober 2023 pukul 09.00 secara online.

3.2 Variable Penelitian

Variabel penelitian yang digunakan dalam penelitian ini diuraikan dalam Tabel 3.1:

Tabel 3.1 Variabel Penelitian

Variabel	Keterangan	Skala	Kategori
X_1	*News Picture* atau gambar berita pada berita yang diterbitkan *turnbackhoax.id* dan *republika.co.id.* Gambar merupakan matriks dengan nilai elemen antara (0,225)	Interval	-
X_2	Konten atau teks berita pada berita yang diterbitkan *turnbackhoax.id* dan *republika.co.id.* Teks merupakan kumpulan kata atau token tanpa tingkatan	Nominal	-
Y	Label berita pada berita yang diterbitkan *turnbackhoax.id* dan *republika.co.id*	Nominal	1: Hoaks 0: Bukan Hoaks

3.3 Struktur Data

Penelitian model klasifikasi berita hoaks dan fakta menggunakan *multimodel deep learning* dengan arsitektur CNN akan mengolah data foto berita dan teks. Pada data teks berita yang telah dilakukan tokenisasi akan membentuk contoh struktur data sebagai yang tertera pada Table 3.2:

Tabel 3.2 Struktur Token Dokumen

Index	Berita
1	['mafindo', 'perlu', 'ada', 'kode', 'etik', 'bagi', 'influencer', 'selamat', 'pagi', 'indonesia', 'beberapa', 'waktu', 'lalu', 'ramai', 'jadi', 'bicara', 'soal', 'konten', 'video', 'yang', 'buat', 'musisi', ..., 'indonesia']
2	['kutip', 'sebut', 'palsu', 'jelas', 'bahwa', 'haedar', 'nashir', 'selaku', 'ketua', 'umum', 'pp', 'muhammadiyah', 'tidak', 'pernah', 'keluar', 'nyata', 'bagaimana', 'yang', 'sekarang', 'jadi', 'cengeng', ..., 'berani']
…	…
d	[…]

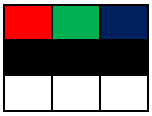
Token teks pada Table 3.2 masing-masing teks akan diolah kedalam bentuk vektor dan dilakukan *embeding* (pembobotan) menggunakan metode TF-IDF. Proses pembobotan membentuk struktur data yang ditampilkan pada Tabel 3.3 untuk masing-masing dokumen atau berita:

Tabel 3.3 Struktur Data TF-IDF

Index	**TF-IDF**				
1	x_{211}	x_{212}	x_{213}	…	x_{21k}
2	x_{221}	x_{222}	x_{223}	…	x_{22k}
3	x_{231}	x_{232}	x_{233}	…	x_{23k}
…	…	…	…	…	…
d	x_{2d1}	x_{2d2}	x_{2d3}	…	x_{2dk}

Pada Tabel 3.3 notasi *d* menunjukkan nomor indeks dokumen atau berita dalam dataset (korpus), notasi *k* menunjukkan nomor indeks kata unik atau token unik dalam korpus. Sementara pada data foto berita akan membentuk struktur data yang mewakili warna setiap *pixel* pada sebuah gambar. Sebuah gambar memiliki tiga buah data pada setiap *pixel* yaitu *red, green dan blue (RGB).* Pada Tabel 3.4 dapat dilihat contoh nilai data RGB yang terbentuk dari contoh gambar sederhana berukuran 3x3 pixel.

Tabel 3.4 Contoh Struktur Gambar 3x3 Pixel

Index	**Struktur Data *Array* Gambar**
1	 [[255,0,0], [0,255,0], [0,0,255], [255,255,255], [255,255,255], [255,255,255], [0,0,0], [0,0,0], [0,0,0]]
2	 [[255,0,0], [0,255,0], [0,0,255], [0,0,0], [0,0,0], [0,0,0] [255,255,255], [255,255,255], [255,255,255]]
…	…

Pada penelitian ini foto berita yang diperoleh dari website *turnbackhoax.id* dan *republika.co.id* diatur sedemikian rupa sehingga memiliki ukuran seragam yaitu 224x224 pixel. Seperti yang diilustrasikan pada Table 3.5 maka, data dari foto berita akan tersusun dari data pixel RGB dengan nilai densitas warna antara 0 hingga 255 untuk setiap berita. Secara keseluruhan struktur data yang terbentuk ditampilkan pada Tabel 3.5:

Tabel 3.5 Struktur Foto berita RGB 224x224 Pixel

Index	**Filter (Kanal)**		
	$x_{1dR}(0)$	$x_{1dG}(1)$	$x_{1dB}(2)$
1	x_{11R11}, x_{11R12}, …, x_{11RPL}, …, $x_{11R\ 224\ 224}$	x_{11G11}, x_{11G12}, …, x_{11GPL}, …, $x_{11G\ 224\ 224}$	x_{11B11}, x_{11B12}, …, x_{11BPL}, …, $x_{11B\ 224\ 224}$
2	x_{12R11}, x_{12R12}, …, x_{12RPL}, …, $x_{12R\ 224\ 224}$	x_{12G11}, x_{12G12}, …, x_{12GPL}, …, $x_{12G\ 224\ 224}$	x_{12B11}, x_{12B12}, …, x_{12BPL}, …, $x_{12B\ 224\ 224}$
…	…	…	…
d	$x_{1dR\ 224\ 224}$	$x_{1dG\ 224\ 224}$	$x_{1dB\ 224\ 224}$

Tabel 3.5 menunjukkan notasi *d* merupakan notasi foto berita menurut nomor indeks dokumen dalam korpus. Kemudian setiap foto berita akan menghasilkan tiga buah filter atau kanal dengan notasi *R, G, dan B* yang menunjukkan nomer indeks filter (0, 1, 2) dari foto berita.

Sementara *P* dan *L* menunjukkan nomor indeks dari koordinat pixel dalam bidang 2D dengan luas bidang 224x224, dimana *P* untuk sumbu x dan *L* untuk sumbu y.

3.4 Teknik *Scrapping* dan *Data Labeling*

Pada tahap *scrapping* (pengumpulan data), metode *web scraping* dijalankan untuk mengakses laman *turnbackhoax.id* dan *republika.co.id*, kemudian mengambil konten dari masing-masing laman. Proses ini melibatkan permintaan HTTP menggunakan *library requests* untuk mengakses laman, kemudian menerjemahkan kode konten HTML menggunakan *library* python yaitu *Beautiful Soup*. Hasil ekstraksi disusun dengan indeks berdasarkan publikasi berita terbaru, dimana indeks pertama merupakan laman *turnbackhoax.id* publikasi 13 Oktober 2023 dan indeks terakhir merupakan laman *republika.co.id* publikasi 1 Januari 2023. Proses ini mengumpulkan data berita berupa tanggal terbit, judul berita, foto berita, dan konten berita yang akan menjadi bahan analisis selanjutnya.

Setelah data berita terkumpul, dilakukan proses *data labeling* (pelabelan data) untuk mengidentifikasi berita hoaks dan fakta. Berita hoaks dapat diidentifikasi berdasarkan kata kunci yang tertera pada setiap judul berita *turnbackhoax.id*, dimana pihak MAFINDO telah berfokus mengidentifikasi berita hoaks yang kemudian dipublikasikan melalui laman tersebut. Setiap judul berita akan diperiksa, jika kata kunci yang merujuk berita hoaks ditemukan maka, berita diberi label sesuai kata kunci. Kata kunci merupakan kata yang diletakkan di awal judul berita pada laman *turnbackhoax.id*. Untuk kata kunci "SALAH", "HOAX", dan "PALSU" akan diberi label sebagai berita hoaks dan dikategorikan dengan angka 1. Sementara berita dengan label fakta dan dikategorikan dengan angka 0 merujuk pada berita yang tidak berlabel hoaks.

Proses *data labeling* dilakukan secara otomatis menggunakan *python* dengan bantuan *library regex*. *Regex* akan mengidentifikasi setiap kata pertama pada tiap judul kemudian membuat kolom yang berisikan hasil penentuan label. Data yang telah diberi label selanjutnya disimpan dalam format Excel menggunakan *library pandas* dan disimpan dalam format file *xlsx* untuk memudahkan akses dalam melakukan analisis.

3.5 Langkah Analisis

Berikut merupakan langkah-langkah analisis yang digunakan dalam mendukung penelitian terkait kalsifikasi berita hoaks dan fakta menggunakan *multimodal deep learning* dijelaskan dibawah ini.

1. Penelitian ini dimulai dengan melakukan pengumpulan data (*scrapping*) dan pemberian label berita dari dua sumber utama, yaitu website *turnbackhoax.id* dan *republika.co.id*. Data yang terkumpul mencakup berbagai informasi seperti teks berita, foto berita, tanggal terbit, dan judul. Selanjutnya, dataset ini dikelompokkan ke dalam dua kategori utama, yaitu berita hoaks dan berita fakta melalui proses *labeling*.
2. Tahapan pre-processing data dalam penelitian ini melibatkan beberapa langkah esensial guna mempersiapkan data agar sesuai dengan model yang akan dikembangkan. berikut langkah-lankah yang dijalankan selama *preprocessing*:
    a. Proses dimulai dengan eksplorasi data yang melibatkan pengumpulan informasi data teks dan gambar terkait dengan struktur dan karakteristik data hoaks dan fakta.
    b. Setelah itu, langkah kedua mengarah pada pembersihan data. Pembersihan data dilakukan dengan menghapus data tidak lengkap dari hasil preprocessing. Sementara pada data teks dilakukan tahapan lebih, yaitu tahapan penghapusan *stopwords*, tokenisasi, dan *lemmatizing* kata-kata ke bentuk dasarnya.
    c. Langkah selanjutnya adalah penyesuaian vektor pada data teks dan gambar foto berita (*vectorization*). Data teks menggunakan metode TF-IDF digunakan untuk

mengubah teks menjadi vektor fitur, di mana sekaligus dilakukan pembobotan. Setiap kata akan diukur menggunakan TF-IDF dan disamakan panjang vektornya. Sementara itu, untuk data gambar foto berita, proses normalisasi diterapkan untuk menyesuaikan nilai *pixel* agar berada dalam rentang dan ukuran resolusi yang sama pada setiap data.

3. Membagi data hasil preprocessing menjadi data *training* dan *testing* dengan ratio pembagian 8:2 menggunakan metode *holdout.*
4. Tahap pembangunan tiga model CNN *deep learning* untuk mengklasifikasikan berita hoaks dan fakta dimulai dari *input layer* hingga output *layer* pada model CNN, baik untuk data teks, gambar, dan gabungan. Berikut adalah penjelasan lebih rinci untuk setiap tahap yang akan dibangun beserta fungsinya:
   a. Model Teks Berbasis CNN (Model CNN 1D):
      1). *Input Layer*: Vektor token, data teks yang telah diproses dan direpresentasikan sebagai vektor menggunakan metode TF-IDF dimasukkan ke dalam *input layer*.
      2). *Convolutional Layer*: Lapisan konvolusi (*Convolution 1D*) mengekstraksi fitur-fitur penting dari vektor.
      3). *Pooling Layer*: Hasil konvolusi melewati lapisan *pooling* untuk mereduksi dimensi data atau vektor.
      4). *Flattening*: Data hasil *pooling* di-flatten menjadi vektor satu dimensi.
      5). *Fully Connected Layer*: Vektor hasil *flattening* dihubungkan dengan *fully connected layer* untuk memproses hubungan antar fitur dan pembobotan.
      6). *Output Layer*: *Node* terakhir menghasilkan nilai probabilitas untuk kelas berita hoaks atau fakta
   b. Model Gambar Berbasis CNN (Model CNN 2D):
      1). *Input Layer*: Matriks *pixel*, matriks yang mewakili nilai *pixel* pada gambar foto berita dimasukkan ke dalam input layer.
      2). *Convolutional Layer*: Lapisan konvolusi (*Convolution 2D*) mengekstraksi fitur-fitur penting dari matriks.
      3). *Pooling Layer*: Hasil konvolusi melewati lapisan *pooling* untuk mereduksi dimensi data atau matriks.
      4). *Flattening*: Data hasil *pooling* di-flatten menjadi vektor satu dimensi.
      5). *Fully Connected Layer*: Vektor hasil *flattening* dihubungkan dengan *fully connected layer* untuk memproses hubungan antar fitur dan pembobotan.
      6). *Output Layer*: *Node* terakhir menghasilkan nilai probabilitas untuk kelas berita hoaks atau fakta
   c. Model Teks dan Gambar Berbasis CNN (Model CNN Gabungan):
      1). *Input Layer*: Data teks dan gambar dimasukkan ke dalam input layer secara terpisah.
      2). *Convolutional Layer*: Jalur teks dan gambar melewati lapisan konvolusi masing-masing.
      3). *Pooling Layer*: Hasil konvolusi diproses melalui lapisan *pooling* pada setiap jalur.
      4). *Flattening*: Data hasil *pooling* dari kedua jalur di-flatten menjadi vektor satu dimensi.
      5). *Fully Connected Layer*: Vektor hasil *flattening* dari kedua jaur dihubungkan dengan *fully connected layer* untuk memproses hubungan antar fitur dan dilakukan pembobotan.
      6). *Output Layer*: *Node* terakhir menghasilkan nilai probabilitas untuk kelas berita hoaks atau fakta
5. Melatih tiga buah model *deep learning* yang sudah dibangun untuk mengklasifikasikan

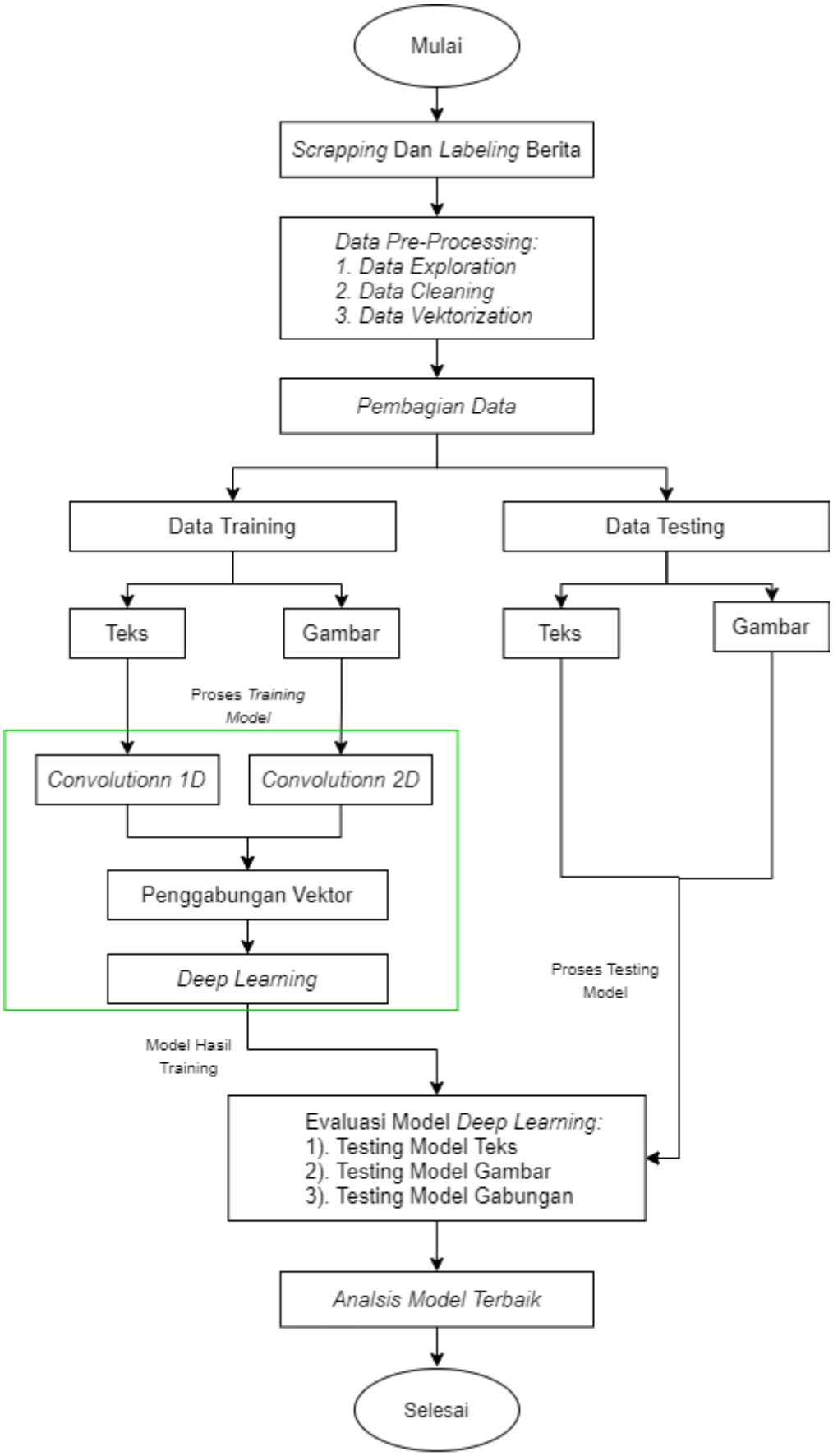
berita hoaks dan fakta menggunakan data *training* dan *testing*. Tahapan pengklasifikasian dilakukan sebagai berikut:

a. Melakukan pelatihan model untuk mengklasifikasikan berita hoaks menggunakan model teks berbasis CNN. Selain itu melatih model teks berbasis SVM dan regresi logistik sebagai model pembanding model CNN terhadap model linier dan non-linier. Model yang telah dilatih dilakukan evaluasi menggunakan data *testing*.
b. Melakukan pelatihan model untuk mengklasifikasikan berita hoaks menggunakan model gambar berbasis CNN. Model yang telah dilatih dilakukan evaluasi menggunakan data *testing.*
c. Melakukan pelatihan model untuk mengklasifikasikan berita hoaks menggunakan model gabungan teks dan gambar berbasis CNN. Model yang telah dilatih dilakukan evaluasi menggunakan data *testing.*

6. Melakukan analisis model terbaik berdasarkan hasil evaluasi model menggunakan nilai akurasi, AUC dari fungsi ROC. Hasil evaluasi terbaik akan dipilih sebagai model terbaik dalam mengklasifikasikan berita hoaks dan fakta.
7. Menarik kesimpulan dan Saran

3.6 Diagram Alir

Diagram alir untuk menggambarkan langkah-langkah analisis secara umum yang dapat disajikan pada Gambar 3.1.



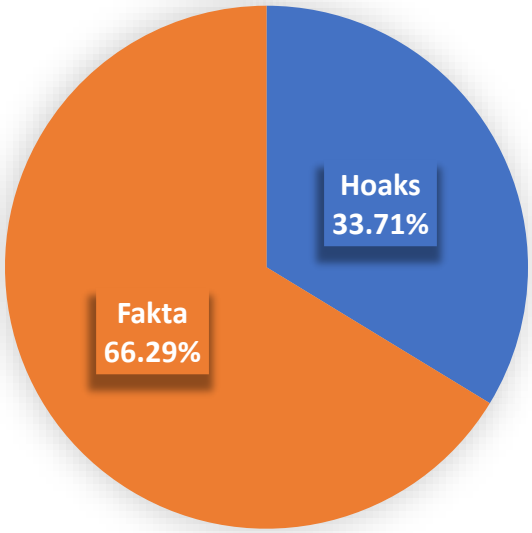
Gambar 3.1 Diagram Alir

BAB IV
HASIL DAN PEMBAHASAN

Bab ini membahas mengenai analisis hasil klasifikasi berita hoaks laman *turnbackhoaks.co.id* dan *republika.co.id*. menggunakan model dengan input foto berita, teks berita, dan gabungan keduanya. Analisis hasil dan pembahasan meliputi hasil *scrapping* berita, proses *labeling*, data *preprocessing*, pembagian data, *feature vectorization*, model klasifikasi, dan evaluasi model.

4.1 Analisis Hasil *Scrapping* dan *Labeling* Berita

Menggunakan metode *scrapping* dan *data labeling* pada website *turnbackhoax.id* dan *republika.co.id*. diperoleh data sebanyak 5165, dimana proporsi kategori ditampilkan pada Gambar 4.1. Sementara *sintax* program untuk model *scrapping* dan *data labeling* dapat dilihat pada Lampiran 3.



Gambar4.1 Proporsi Kategori Berita

Gambar 4.1 menunjukkan 1741 (33,71%) merupakan data *turnbackhoaks.co.id* dengan label hoaks dan 3424 (66,29%) berita berlabel fakta berasal dari *republika.co.id*. Proses *labeling* pada laman *turnbackhoax.id* menunjukkan bahwa data periode publikasi tanggal 1 Januari 2023 hingga 13 Oktober 2023 keseluruhan berita dikategorikan sebagai berita hoaks. Sementara laman *republika.co.id* pada periode publikasi yang sama, secara keseluruhan merupakan berita dengan kategori fakta. Sehingga diperoleh perbandingan antara kategori hoaks dan fakta sebesar 1 banding 1,967 mendekati rasio 1:2 dengan total data 5165 baris. Hal ini menunjukkan terdapat ketidak setimbangan kategori data, oleh karena itu perlu dilakukan penanganan agar data kembali setimbang. Penanganan dilakukan dengan pengurangan data pada kategori fakta dan memprioritaskan publikasi terbaru, sehingga data kategori fakta yang terpilih merupakan berita dengan publikasi 13 Oktober 2023 hingga 7 Maret 2023. Setelah dilakukan pemotongan perbandingan antra kategori hoaks dan fakta serta total keseluruhan data sebanyak data 3878.

4.2 Analisis *Preprocessing* Berita

Pre-processing pada data untuk model *machine learning*, baik untuk data teks maupun gambar, menjadi langkah krusial untuk meningkatkan kualitas dan kinerja model sekaligus mengurangi beban komputasi. Preprocessing melewati tiga tahap yaitu eksplorasi data (*data exploration*), pembersihan data (*data cleaning*), dan vektorisasi data (*data vectorization*). Setiap tahapan akan dijelaskan lebih lanjut sebagai berikut:

4.2.1 Data Exploration

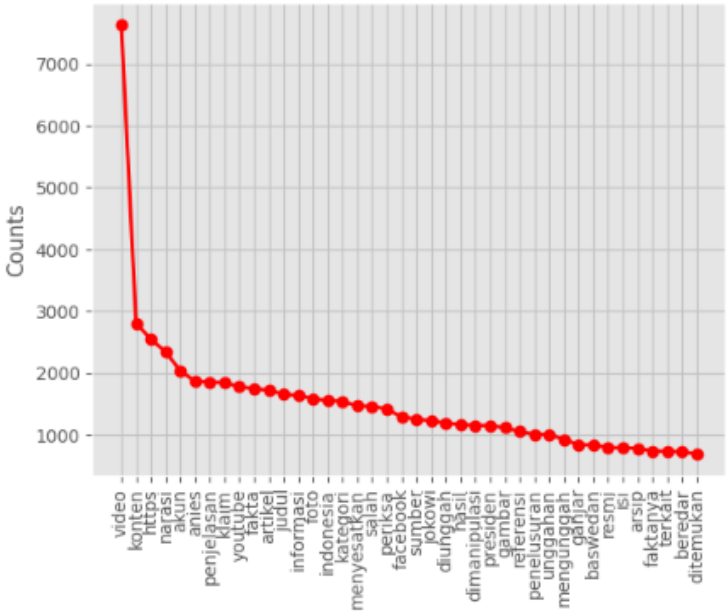
Eksplorasi data dilakukan pada data teks dan gambar foto berita yang diperoleh dari hasil *scrapping* berita dimana *sintax* yang digunakan tertera pada Lampiran 5. Data teks terdiri dari token yang merepresentasikan kata dalam sebuah berita, dilakukan eksplorasi untuk mengetahui berapa frekuensi dari setiap jenis kata dalam korpus. Berikut perbandingan jumlah token dan token unik pada tiap kategori yang ditampilkan pada Tabel 4.1:

Tabel 4.1 Tabel Token

Kategori Berita	Token	Token Unik
Hoaks	246988	20057
Fakta	413377	32174
Hoaks dan Fakta	660365	41353

Tabel 4.1 menunjukkan perbandingan jumlah token dan jumlah token unik pada kategori berita fakta lebih banyak dibandingkan kategori hoaks. Hal tersebut dapat diinterpretasikan kategori fakta memiliki kekayaan informasi, lebih baik secara penulisan berita, dan liputan yang lebih luas. Perbedaan ini juga dapat mencerminkan berita kategori fakta memberikan rincian yang lebih mendalam dan akurat dibandingkan dengan berita kategori hoaks. Selain itu, gaya penulisan, diversitas topik, dan tujuan publikasi juga berkontribusi pada perbedaan dalam struktur dan konten berita antar kategori tersebut sehingga membantu dalam pengenalan pola.

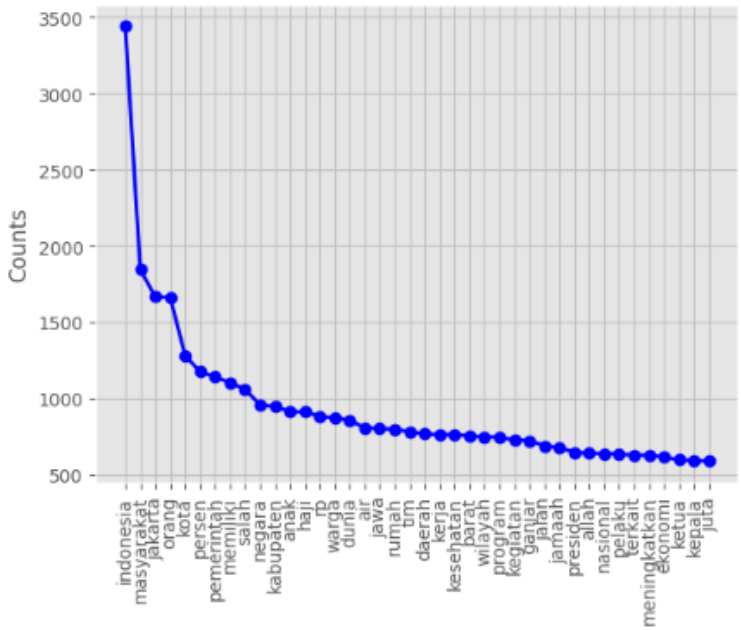
Selanjutnya perbandingan jumlah token (frekuensi) untuk tiap token unik yang muncul pada masing-masing kategori. Pada Gambar 4.2 menampilkan sebaran tiap token unik pada kategori hoaks:



Gambar4.2 Frekuensi Token Unik Kategori Hoaks

Gambar 4.2 menunjukkan 40 token dengan frekuensi terbanyak pada kategori berita hoaks, dimana tiga token terbanyak yaitu video, konten, dan https. Token pada kategori hoaks

sebagian besar merupakan token yang menunjukkan keterangan jejak atau sumber digital. Kemudian ditampilkan sebaran tiap token unik pada kategori fakta pada Gambar 4.3:



Gambar4.3 Frekuensi Token Unik Kategori Fakta

Gambar 4.3 menunjukkan 40 token dengan frekuensi terbanyak pada kategori berita fakta, dimana tiga token terbanyak yaitu indonesia, masyarakat, dan jakarta untuk kategori fakta. Token pada kategori fakta sebagian besar menunjukkan keterangan tempat.

Eksplorasi berikutnya dilakukan pada data foto berita yang telah diperoleh dari proses *scrapping*. Hasilnya adalah adanya perbedaan ukuran atau resolusi pada foto berita kategori hoaks dan fakta dimana, berita hoaks memiliki resolusi foto berita lebih rendah dibanding dengan foto berita fakta seperti yang ditampilkan Gambar 4.4:

(A) Resolusi: 678×381

(B) Resolusi: 830×556

Gambar4.4 Foto Berita

Gambar 4.4 bagian (A) mewakili foto berita yang diperoleh dari laman *turnbackhoax.id* dengan nilai resolusi 678×381 dan memiliki ukuran resolusi yang sama pada setiap berita. Sementara bagian (B) mewakili foto berita dari laman *republika.co.id* dengan nilai resolusi 830×556 dan setiap gambar foto berita memiliki nilai yang beragam. Adanya perbedaan resolusi pada foto berita akan dilakukan penganan dengan memperkecil resolusi, hal ini sekaligus mengurangi beban komputasi.

4.2.2 Data Cleaning

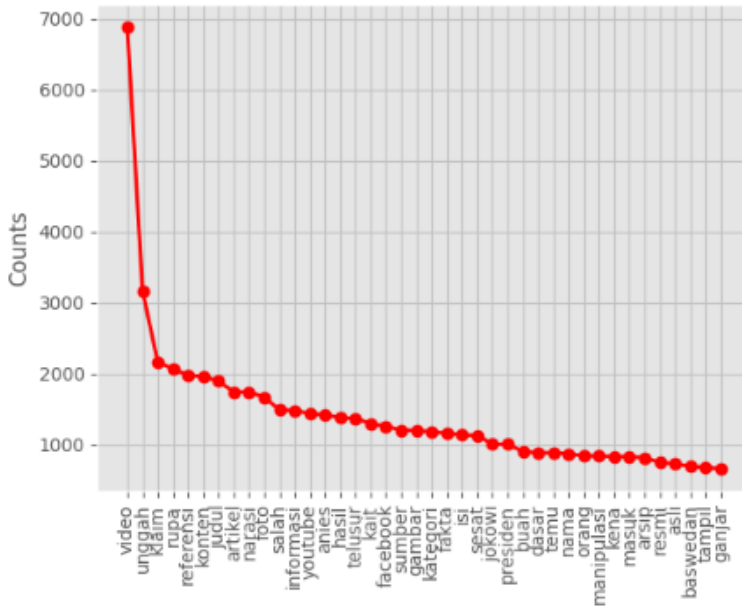
Data cleaning bertujuan menghilangkan data yang tidak signifikan dalam mengklasifikasikan berita hoaks dan fakta menggunakan *sintax* pada Lampiran 5. Tahap *cleaning* pada data teks dilakukan pembersihan lanjutan untuk menghilangkan *casefolding*, tanggal, satuan, link, *punctuation*, *emoticon*, angka, dan karakter khusus lainnya serta tahapan *lemmatizing*. Hasil *cleaning* pada data teks dilakukan eksplorasi kembali untuk melihat perbedaan, berikut perbandingan dari masing-masing kategori ditunjukkan pada Tabel 4.2:

Tabel 4.2 Tabel Token Bersih

Kategori Berita	Token	Token Unik
Hoaks	202811	16043
Fakta	402562	26448
Hoaks dan Fakta	605373	34729

Tabel 4.2 menunjukkan perbandingan jumlah token dan jumlah token unik pada kategori berita fakta tetap lebih banyak dibandingkan kategori hoaks meskipun telah melalui tahap *cleaning*. Hal tersebut dapat di interpretasikan kategori fakta tetap memiliki kekayaan informasi, lebih baik secara berita, dan liputan yang lebih serta memberikan rincian yang lebih mendalam dan akurat dibandingkan dengan berita kategori hoaks.

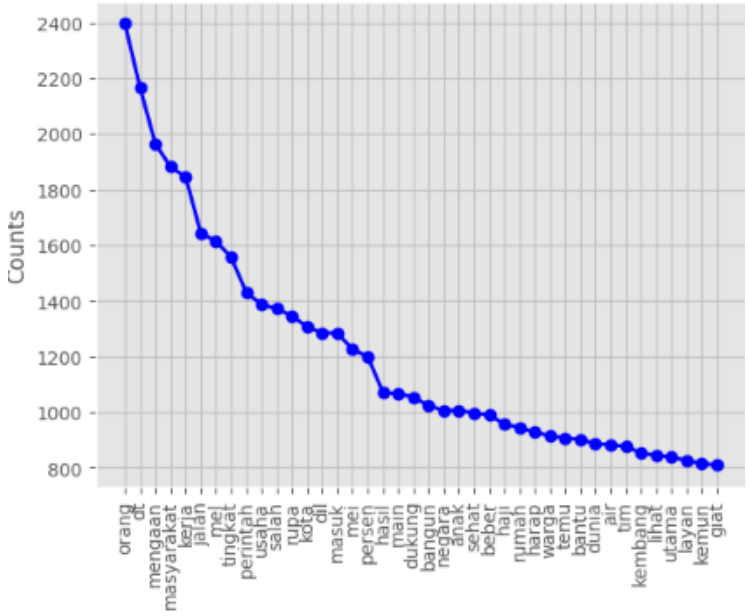
Selanjutnya perbandingan token yang sering muncul pada masing-masing kategori ditampilkan pada Gambar 4.5:



Gambar4.5 Frekuensi Token Unik Kategori Hoaks Bersih

Gambar 4.5 menunjukkan 40 token dengan frekuensi terbanyak pada kategori berita hoaks yang telah dibersihkan, dimana tiga token terbanyak yaitu video, unggahan, dan klaim. Token pada kategori hoaks sebagian besar merupakan token yang menunjukkan keterangan jejak atau sumber digital. Kemudian ditampilkan sebaran tiap token unik pada kategori fakta pada Gambar 4.6:

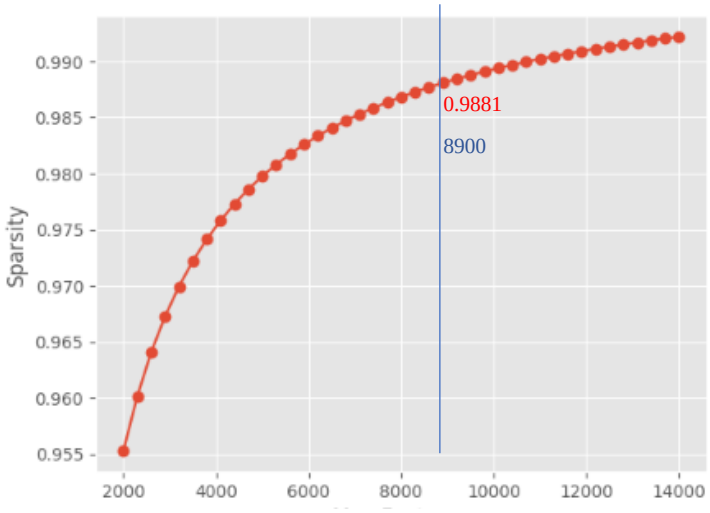
Gambar 4.6 menunjukkan 40 token dengan frekuensi terbanyak pada kategori berita fakta yang telah dibersihkan, dimana tiga token terbanyak yaitu orang, dt, dan mengaan untuk kategori fakta. Token pada kategori fakta sebagian besar menunjukkan kata yang lebih acak atau beragam.



Gambar4.6 Frekuensi Token Unik Kategori Fakta Bersih

4.2.3 Data Vectorization

Data vectorization bertujuan melakukan penyesuaian vektor pada data teks dan gambar foto berita, dimana *sintax* yang digunakan tertera pada Lampiran 5. Pertama hasil vektorisasi pada data teks adalah TF-IDF, berupa vektor yang mewakili token unik. Jumlah elemen vektor (panjang vektor) yang digunakan akan mempengaruhi kecepatan dan akurasi model yang dilatih. Untuk mempercepat pelatihan model tanpa membebani mesin komputasi serta diperoleh akurasi yang optimal dilakukan pengurangan elemen vektor TF-IDF. Oleh karena itu salah satu cara penentuan panjang vektor optimal dapat digunakan nilai *sparsity* yang dihitung menggunakan persamaan (2.4). Berikut grafik hubungan panjang vektor terhadap nilai sparsity pada berita ditampilkan pada Gambar 4.3:



Gambar4.7 Hubungan panjang vektor dan *Sparsity*

Gambar 4.7 mengilustrasikan panjang vektor dalam rentang 2000 hingga 14000, yang menampilkan sebagian dari token unik kategori hoaks dan fakta pada Tabel 4.2. Pemilihan nilai *sparsity* ditentukan ketika perubahan nilainya tidak signifikan terhadap jumlah elemen atau panjang vektor. Pada Gambar 4.7, panjang vektor 8000 hingga 10000 memiliki selisih nilai kurang dari 0,005 dan di atas 98%. Selain itu pada rentang tersebut akan diperoleh panjang vektor yang lebih ringan untuk dilakukan komputasi dibandingkan menggunakan keseluruhan

elemen token unik pada Tabel 4.2 yang nilai *sparsity* mencapai di atas 99% .Dalam konteks model ini, keputusan diambil untuk menetapkan panjang vektor sebanyak 8900, dengan nilai *sparsity* 98,8%. Keputusan ini diambil untuk memastikan bahwa panjang vektor tidak terlalu besar, untuk mengurangi beban proses komputasi, dan sekaligus tetap mempertahankan tingkat *sparsity* yang tinggi serta diharapkan dapat mencapai keseimbangan yang optimal antara akurasi dan efisiensi komputasi dalam model, tanpa mengorbankan akurasi yang signifikan.

Sementara data gambar foto berita, akan diubah menjadi resolusi yang seragam yaitu, gambar beresolusi 224×224 pixel. Representasi hasil vektorisasi gambar foto berita pada masing-masing kategori ditampilkan pada Gambar 4.8:

(A) Resolusi: 224×224 **(B) Resolusi: 224×224**

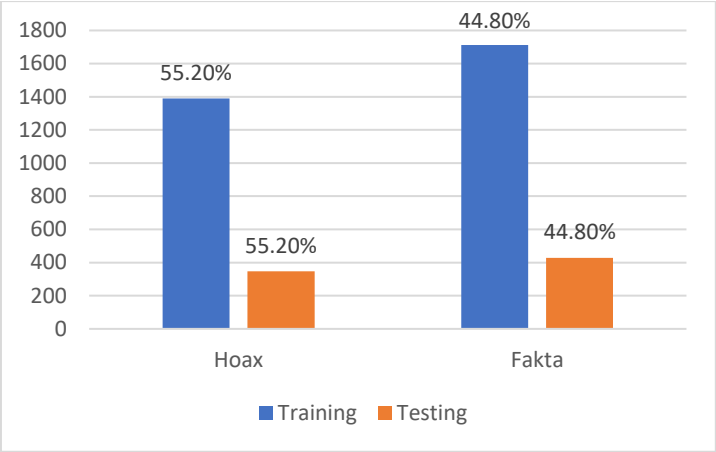
Gambar4.8 Vektorisasi Foto Berita

Gambar 4.8 menunjukkan proses vektorisasi membuat data gambar dengan resolusi lebih besar yaitu berita kategori fakta (B) kehilangan lebih banyak fitur dibanding berita foto berita kategori hoaks (A). Sementara penggunaan resolusi dengan ukuran 224×224 merupakan penyesuaian ukuran dimensi *layer* input berdasarkan arsitektur CNN. Selanjutnya, nilai *pixel* dinormalisasi dari 0 hingga 255 menjadi rentang antara 0 dan 1, mengoptimalkan representasi visual dan memastikan konsistensinya. Adanya pengurangan resolusi dan normalisasi bertujuan mengurangi kompleksitas data sekaligus mempercepat proses *training* dan *testin*g.

4.3 Analisis Pembagian Data

Menggunakan metode *Holdout*, data gambar foto berita dan teks berita yang akan digunakan ke dalam pelatihan telah dibagi kedalam *training* dan *testing* dengan rasio 80:20 berdasarkan *sintax* pada Lampiran 4. Melalui pembagian data yang dilakukan secara acak, 80% data (3103) terpilih sebagai *training* dan 20% data (775) terpilih sebagai data testing. Untuk data training dan testing beserta proporsi label hoaks dan fakta ditampilkan pada Gambar 4.9:

Gambar 4.9, dapat disimpulkan bahwa pembagian data training dan testing untuk kedua label (hoaks dan fakta) memiliki proporsi yang seimbang. Proporsi yang seragam ini mengindikasikan bahwa proses pembagian data berjalan dengan lancar, memastikan bahwa model yang akan dikembangkan akan terlatih dengan baik pada berbagai jenis label. Dengan demikian, penggunaan metode *holdout* dalam pembagian data ini dapat dianggap berhasil dan dapat diandalkan untuk melatih model dengan keberagaman yang cukup, yang kemudian diharapkan dapat memberikan hasil evaluasi yang akurat dan dapat diandalkan pada tahap pengujian.



Gambar4.9 Proporsi *Training-Testing*

4.4 Analisis Model Klasifikasi Berita

Model klasifikasi berita hoaks dan fakta dibangun berdasarkan model prediksi *unimodel* yang berbasis data gambar dan teks serta *multimodel* yang merupakan gabungan keduanya. Ketiga model akan menggunakan *deep learning* dengan model arsitektur CNN, dimana setiap model dilakukan *training* dan *testing*. Model CNN berbasis teks dapat kita sebut sebagai model CNN 1D, model CNN berbasis gambar sebagai model CNN 2D, dan model yang menggabungkan keduanya dapat disebut model CNN gabungan. Berikut hasil dan evaluasi *training* dan *testing* ketiga model:

4.4.1 Model Klasifikasi Berbasis Teks

Membangun model Klasifikasi berbasis teks dengan kebaikan model yang baik membutuhkan percobaan dengan berbagai *parameter* yang mengontrol proses *feature extraction* dan pembobotan pada NN. Dalam hal ini model klasifikasi berbasis teks, model CNN 1D, menggunakan susunan arsitektur dan *parameter* yang sama pada contoh arsitektur model CNN 1D pada Tabel 2.11. *Parameter* yang digunakan pada Model CNN 1D disajikan pada Tabel 4.3:

Tabel 4.3 *Parameter* Model CNN 1D

No	Keterangan Parameter	Nilai Parameter
1	Jumlah *Block*	1
2	*Convolution 1D Layer*	2
3	*Pooling Layer*	*Global-max-pooling*
4	Fungsi Non-linier	*ReLU*
5	Jumlah Epoch	100
6	*Batch Size*	32
7	*Optimizer*	*Adam*
8	*Loss Function*	*Binary-crossentropy*
9	*Early Stopping Patience*	20

Tabel 4.3 menunjukkan *parameter* yang digunakan pada model CNN 1D untuk mengklasifikasikan berita hoaks dan fakta berbasis teks dengan keterangan sebagai berikut:

1. *Block*: Terdapat satu *block* pada arsitektur model CNN 1D.
2. *Convolution 1D layer*: Terdapat dua *layer* konvolusi 1D pada tiap *block*.
3. *Pooling Layer*: Menggunakan *Global-max-pooling* sebagai metode *pooling*, yang berarti nilai maksimum diambil dari seluruh hasil konvolusi pada setiap *block*.

4. Fungsi Aktivasi: Menggunakan *ReLU* sebagai fungsi aktivasi pada berbagai *layer*, *ReLU* menyediakan fungsi non-linear sehingga mampu menangkap kompleksitas fitur.
5. Jumlah *Epoch*: Proses pelatihan dilakukan sebanyak 100 kali.
6. *Batch Size*: Ukuran *batch* pada setiap iterasi pelatihan adalah 32, yang mengontrol seberapa banyak sampel yang diproses sekaligus dalam satu iterasi.
7. *Optimizer*: Menggunakan algoritma *optimizer Adam*.
8. *Loss Function*: Menggunakan *binary-crossentropy* sebagai fungsi *loss* untuk masalah klasifikasi biner.
9. *Early Stopping Patience*: Proses pelatihan akan dihentikan jika nilai validasi akurasi tidak ada kenaikan atau perubahan signifikan dalam 20 *epoch* (sesi) dari total *epoch*.

Selanjutnya Model klasifikasi berbasis teks yang telah dibangun, dijalankan menggunakan *sintax* pada Lampiran 6 yang terdiri dari *feature extraction* satu dimensi dan model NN. Model berbasis teks akan dibandingkan dengan model linier regresi logistik dan model non-linier SVM. Evaluasi menggunakan nilai akurasi dan AUC dari fungsi ROC yang terbentuk menggunakan data testing, berikut hasil yang diperoleh ditampilkan pada Table 4.4:

Tabel 4.4 Evaluasi Model Berbasis Teks

Metode	Akurasi	AUC
SVM	0,5523	0,5005
Regression Logistik	0,5548	0,5029
Model CNN 1D	0,8232	0,8817

Berdasarkan Tabel 4.4 menunjukkan model CNN satu dimensi memiliki nilai akurasi yang berbeda signifikan dibandingkan dengan model regresi logistik dan SVM. Model CNN 1D memiliki nilai lebih baik jika dibandingkan model SVM dan model regresi logistik. Selain itu model CNN 1D miliki akurasi kurang dari 85% sehingga dapat dikatakan kurang reliabel memprediksi data dalam model. Sementara nilai AUC model CNN 1D mendekati di atas 85% atau mendekati 1, sehingga masuk dalam kategori sangat baik dan mampu mengklasifikasikan data di luar model. Model CNN 1D memiliki kinerja sangat baik, mendekati sempurna, dan mampu membedakan dengan tingkat akurasi sangat tinggi dibandingkan model regresi logistik dan SVM.

Adapun metode seperti *deep learning* yang digunakan model CNN 1D, kebaikan model juga harus dilihat dari perubahan nilai *error* atau *loss* dalam setiap *epoch* pelatihan agar tidak terjadi *overfitting* maupun *underfitting*. Perbandingan antara nilai *loss* saat melatih data dan memvalidasi model CNN 1D menggunakan teks disajikan pada Gambar 4.10:

Gambar4.10 Model 1D *Loss* (kiri) dan Akurasi (kanan)

Gambar 4.10 menunjukkan gap nilai antara *loss* pada proses *training* dan *testing (*validasi) model CNN 1D memiliki gap yang semakin besar, dimana nilai validasi *loss* lebih tinggi dibandingkan *training* yang menunjukkan nilai loss lebih tinggi atau model mengalami *underfit.* Begitu juga gap nilai pada akurasi CNN 1D juga mengalami *underfit* , yaitu nilai akurasi validasi lebih rendah dibandingkan *training*. Untuk mengatasi *underfit* pada model dilakukan pengembalian nilai bobot pada epoch ke-37 yang memiliki nilai *loss* terkecil dan nilai akurasi terbesar. *Epoch* ke-37 diperoleh dari pelatihan model dengan 100 epoch dan akan otomatis berhenti setelah diperoleh 20 epoch yang tidak berbeda signifikan (*earlystopping)* dan mengembalikan epoch ke-37 sebagai bobot terbaik. Output model CNN 1D pada *epoch* ke-37 sebagai bobot terbaik setelah dilakukan *training* dan *testing* juga dapat pada Lampiran 7.

4.4.2 Model Klasifikasi Berbasis Gambar

Membangun model Klasifikasi berbasis gambar dengan kebaikan model yang baik membutuhkan percobaan dengan berbagai *parameter* yang mengontrol proses *feature extraction* dan pembobotan pada *NN*. Dalam hal ini salah satu model klasifikasi berbasis gambar, model CNN 2D, menggunakan susunan arsitektur dan *parameter* yang sama pada contoh arsitektur model CNN pada Tabel 2.10. *Parameter* yang digunakan pada Model CNN 2D disajikan pada Tabel 4.5

Tabel 4.5 *Parameter* Model CNN 2D

No	Keterangan Parameter	Nilai Parameter
1	Jumlah *Block*	5
2	*Convolution 2D Layer*	2 dan 3
3	*Pooling Layer*	*max-pooling*, (2×2), *stride* 2
4	Fungsi Non-linier	*ReLU*
5	Jumlah Epoch	100
6	*Batch Size*	32
7	*Optimizer*	*Adam*
8	*Loss Function*	*Binary-crossentropy*
9	*Early Stopping Patience*	20

Tabel 4.5 menunjukkan *parameter* yang digunakan pada model CNN 2D untuk mengklasifikasikan berita hoaks dan fakta berbasis teks dengan keterangan sebagai berikut:

1. *Block*: Terdapat 5 *block* pada arsitektur model CNN 2D.
2. *Convolution 2D Layer*: *block* 1 dan 2 menggunakan dua *layer* sedangkan *block* 3, 4, dan 5 menggunakan tiga *layer*.
3. *Pooling Layer*: Menggunakan *max-pooling* sebagai metode *pooling*, yang berarti nilai maksimum dari filter berukuran (2×2) dan *stride* 2
4. Fungsi Aktivasi: Menggunakan *ReLU* sebagai fungsi aktivasi pada berbagai *layer*. *ReLU* menyediakan fungsi non-linearitas sehingga mampu menangkap kompleksitas fitur.
5. Jumlah *Epoch*: Proses pelatihan dilakukan sebanyak 100 kali.
6. *Batch Size*: Ukuran *batch* pada setiap iterasi pelatihan adalah 32, yang mengontrol seberapa banyak sampel yang diproses sekaligus dalam satu iterasi.
7. *Optimizer*: Menggunakan algoritma *optimizer Adam*.
8. *Loss Function*: Menggunakan *binary-crossentropy* sebagai fungsi loss untuk masalah klasifikasi biner.
9. *Early Stopping Patience*: Proses pelatihan akan dihentikan jika nilai validasi akurasi tidak ada kenaikan atau perubahan signifikan dalam 20 *epoch* (sesi) dari total *epoch*.

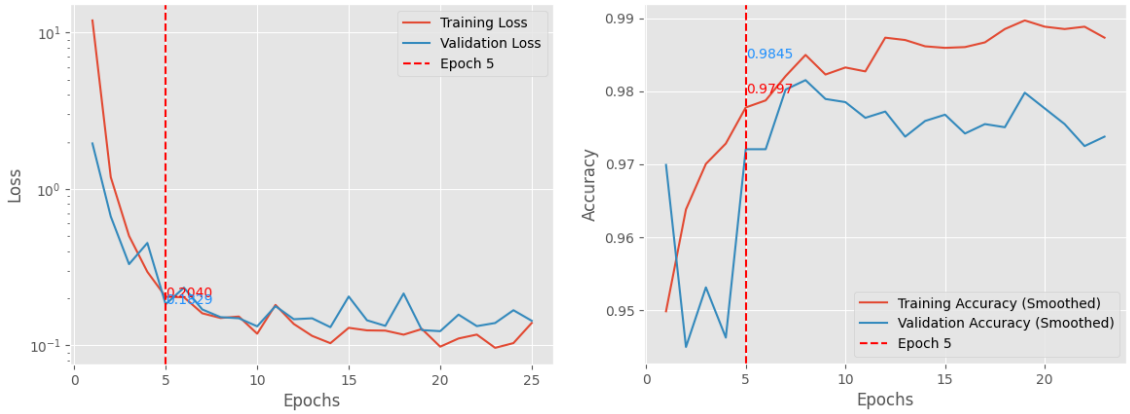
Model klasifikasi berbasis gambar menggunakan yang sudah dibangun dijalankan menggunakan *sintax* pada Lampiran 6 yang terdiri dari *feature extraction* konvolusi dua dimensi dan model NN. Hasil testing model klasifikasi berbasis gambar dilakukan evaluasi dimana, evaluasi menggunakan nilai akurasi dan AUC dari fungsi ROC yang terbentuk menggunakan data *testing*. Berikut hasil yang diperoleh ditampilkan pada Table 4.6:

Tabel 4.6 Evaluasi Model Berbasis Gambar

Metode	Akurasi	AUC
Model CNN 2D	0,9845	0,9984

Berdasarkan Tabel 4.6 menunjukkan model CNN dua dimensi memiliki nilai akurasi yang dan nilai AUC lebih baik dari model satu dimensi sebelumnya. Model miliki akurasi di atas 85% sehingga dapat dikatakan model reliabel atau mampu mengklasifikasikan data dalam model. Sementara nilai AUC dari fungsi ROC di atas 85% atau mendekati 1, sehingga kemampuan prediksi bersifat tidak acak dan mampu mengklasifikasikan data di luar model.

Seperti model CNN 1D sebelumnya, model *deep* learning yang menggunakan model CNN 2D, kebaikan model juga harus dilihat dari perubahan nilai *loss* dan akurasi pada tiap *epoch* pelatihan agar tidak terjadi *overfitting* maupun *underfitting*. Perbandingan antara nilai *error* atau *loss* saat melatih data dan validasi model CNN berbasis gambar disajikan pada Gambar 4.11:



Gambar4.11 Model CNN 2D *Loss* (kiri) dan Akurasi (Kanan)

Gambar 4.11 menunjukkan hal yang sama seperti Gambar 4.10, yaitu gap nilai antara *loss* dan akurasi pada proses *training* dan *testing* model CNN 2D semakin besar sehingga model mengalami *underfit*. Kemudian dilakukan penanganan pengembalian nilai, dari 100 *epoch training* telah diperoleh 20 epoch yang tidak berbeda signifikan dan mengembalikan *epoch* ke-5 sebagai bobot terbaik. Output model CNN 2D pada *epoch* ke-5 sebagai bobot terbaik setelah dilakukan *training* dan *testing* juga dapat pada Lampiran 7. Sementara hasil pengembalian nilai model CNN 2D memberikan hasil lebih baik dibandingkan model CNN 1D dalam mengklasifikasikan berita hoaks.

4.4.3 Model Klasifikasi Berbasis Teks dan Gambar

Model klasifikasi berbasis teks dan gambar (model CNN gabungan) menggunakan arsitektur CNN dibangun menggunakan *sintax* pada Lampiran 1. Model gabungan terdiri dari *feature extraction* pada model CNN 1D dan model CNN 2D serta parameter yang sama seperti dua model yang telah dianalisis sebelumnya. Selanjutnya fitur teks dan gambar akan dilakukan pembobotan bersama pada model NN. *Parameter* yang digunakan pada Model CNN kombinasi disajikan pada Tabel 4.7:

Tabel 4.7 *Parameter* Model CNN kombinasi

No	Keterangan Parameter	Nilai Parameter
1	Jumlah *Block*	2(CNN 1D dan CNN 2D)
2	Jumlah Epoch	100
3	*Batch Size*	32
4	*Optimizer*	*Adam*
5	*Loss Function*	*Binary-crossentropy*
6	*Early Stopping Patience*	20

Tabel 4.5 menunjukkan *parameter* yang digunakan pada model CNN kombinasi untuk mengklasifikasikan berita hoaks dan fakta berbasis teks dengan keterangan sebagai berikut:

1. *Block*: Terdapat dua *block* pada arsitektur model CNN kombinasi yaitu *block* yang sama pada model CNN 1D dan CNN 2D, dimana parameter *pooling*, fungsi non-linier, dan *konvolusi* di dalamnya sama seperti model sebelumnya.
2. Jumlah *Epoch*: Proses pelatihan dilakukan sebanyak 100 kali.
3. *Batch Size*: Ukuran *batch* pada setiap iterasi pelatihan adalah 32, yang mengontrol seberapa banyak sampel yang diproses sekaligus dalam satu iterasi.
4. *Optimizer*: Menggunakan algoritma *optimizer Adam*.
5. *Loss Function*: Menggunakan *binary-crossentropy* sebagai fungsi *loss* untuk masalah klasifikasi biner.
6. *Early Stopping Patience*: Proses pelatihan akan dihentikan jika nilai validasi akurasi tidak ada kenaikan atau perubahan signifikan dalam 20 *epoch* (sesi) dari total *epoch*. Setelah itu model akan mengembalikan bobot pada *epoch* dengan nilai validasi akurasi tertinggi.

Model CNN kombinasi yang telah dilakukan *training* selanjutnya dilakukan *testing* dan evaluasi untuk mengetahui kemampuan model membedakan kategori berita aktual dengan hasil prediksi menggunakan nilai akurasi dan fungsi ROC. Evaluasi menggunakan nilai akurasi dan AUC dari fungsi ROC yang terbentuk menggunakan data *testing*, berikut hasil yang diperoleh ditampilkan pada Table 4.8:

Tabel 4.8 Model Berbasis Teks dan Gambar

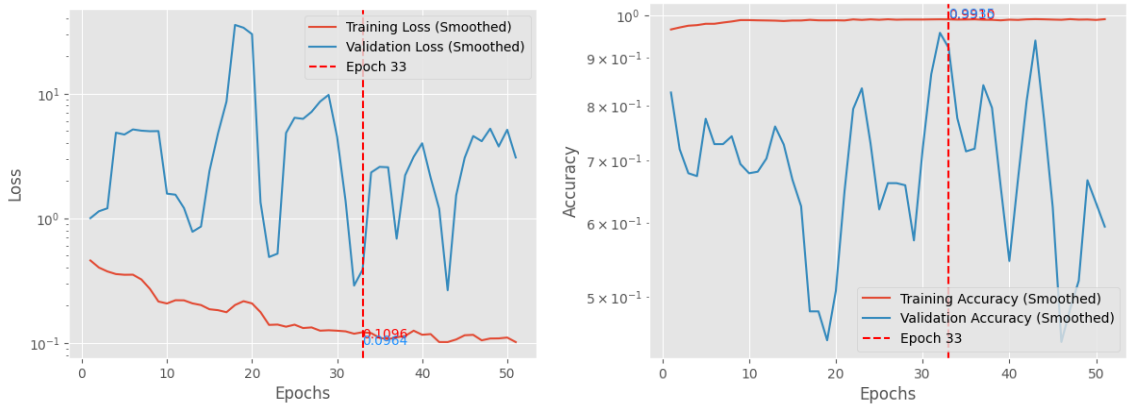
Metode	Akurasi	AUC
Model CNN Gabungan	0,9935	0,9981

Tabel 4.8 menunjukkan model kombinasi memiliki nilai akurasi yang tinggi dan nilai AUC lebih baik dari model satu dan dua dimensi sebelumnya. Model miliki akurasi di atas 85% sehingga dapat dikatakan model reliabel dalam mengklasifikasikan data dalam model. Sementara nilai AUC dari fungsi ROC mendekati 1 atau masuk dalam kategori sangat baik dan mampu mengklasifikasikan data di luar model. Model CNN kombinasi memiliki kinerja sangat baik, mendekati sempurna, dan mampu membedakan dengan tingkat akurasi sangat tinggi.

Model CNN gabungan yang telah dilakukan *training* dan *testing*, model juga harus dilihat dari perubahan nilai *loss* dalam setiap *epoch* pelatihan agar tidak terjadi *overfitting* maupun *underfitting*. Perbandingan antara nilai *error* atau *loss* saat melatih data dan validasi model CNN berbasis gambar disajikan pada Gambar 4.12:

Sama seperti model CNN 1D dan 2D, Gambar 4.12 menunjukkan gap nilai antara *loss* dan akurasi pada proses *training* dan *testing* model CNN gabungan memiliki gap yang semakin besar sehingga menimbulkan *underfit*. Selanjutnya dilakukan penganan yang sama seperti kedua model sebelumnya dimana dari 100 *epoch* telah diperoleh 20 *epoch* yang tidak berbeda signifikan dan mengembalikan *epoch* ke-33 sebagai bobot terbaik. Output model CNN gabungan pada *epoch* ke-33 sebagai bobot terbaik setelah dilakukan *training* dan *testing* juga

dapat pada Lampiran 7. Selain itu proses pengembalian bobot memberikan hasil model CNN gabungan mampu mengungguli model CNN 1D dan 2D.



Gambar4.12 Model CNN Kombinasi *Loss* (kiri) dan Akurasi (kanan)

4.5 Model Evaluasi

Model klasifikasi berita hoaks dan fakta dibangun berdasarkan model prediksi berbasis data gambar, teks, dan gabungan keduanya. Ketiga basis model dilakukan pelatihan model (*training*) dan pengujian model (*testing*) untuk mendapatkan nilai akurasi dan AUC. Nilai tersebut yang kemudian digunakan sebagai evaluasi dalam mengetahui performa model terbaik untuk mengklasifikasikan berita hoaks dan fakta. Berikut Tabel 4.9 ditampilkan perbandingan keseluruhan model dilihat dari nilai akurasi dan AUC dari fungsi ROC:

Tabel 4.9 Perbandingan Model Klasifikasi

Metode	Akurasi	AUC	Jenis Model	Jenis Input
SVM	55,226%	50,055%	*Unimodel*	Teks
Logistic Regression	55,484%	50,288%	*Unimodel*	Teks
Model CNN Teks (1D)	82,320%	88,170%	*Unimodel*	Teks
Model CNN Gambar (2D)	98,450%	99,840%	*Unimodel*	Gambar
Model CNN Gabungan	99,350%	99,850%	*Multimodel*	Teks dan Gambar

Tabel 4.9, dapat disimpulkan bahwa model CNN gabungan, yang mengintegrasikan data gambar dan teks, menunjukkan kinerja yang sangat baik dalam memprediksi berita hoaks dan fakta dibandingkan dengan model-model lainnya. Model CNN gabungan mencapai akurasi sebesar 99,35% yang menunjukkan kemampuan model dalam mengklasifikasikan berita hoaks dan fakta pada data *testing*. Sementara nilai AUC sebesar 99,85%, menandakan kemampuannya yang sangat baik dalam membedakan antara berita hoaks dan fakta pada situasi di mana distribusi kelas atau ambang batas klasifikasi dapat berubah. Hal ini membuat *multimodel* berbasis gabungan foto berita dan teks berita unggul dibandingkan dengan *unimodel*.

Sementara itu, *unimodel* terbaik, yaitu model CNN berbasis gambar (model CNN 2D), memiliki akurasi yang sedikit lebih rendah, yakni 98,45% dan memiliki AUC yang lebih rendah sebesar 99,84% dibandingkan dengan *multimodel*. Secara umum, *multimodel*, khususnya Model CNN gabungan, dianggap sebagai model terbaik dalam mengklasifikasikan berita hoaks, didasarkan pada akurasi yang tinggi dalam akurasi model. Di sisi lain, SVM, *Logistic*

Regression, dan Model CNN berbasis teks (1D) yang fokus pada data teks menunjukkan performa yang relatif lebih rendah. Model CNN berbasis teks (model CNN 1D) lebih baik dibandingkan dengan dua model berbasis teks menggunakan metode linier dan non-linier sebelumnya.

Selain itu model terbaik juga mengindikasikan bahwa pemanfaatan gambar foto yang telah dikategorikan lebih efektif dalam kasus ini, dimana kemampuan visual dari gambar memberikan kontribusi besar dalam membedakan antara berita hoaks dan fakta. Model-model berbasis teks mungkin memerlukan penyesuaian atau pengembangan parameter untuk meningkatkan kinerjanya dalam memprediksi berita hoaks dan fakta pada dataset yang bersangkutan. Selain itu, jumlah dan filter yang dipelajari pada model berbasis gambar lebih banyak dibandingkan dengan model berbasis teks.

Halaman ini sengaja dikosongkan

BAB V
KESIMPULAN DAN SARAN

5.1 Kesimpulan

Berdasarkan hasil analisis mengenai proses dalam membangun dan mengevaluasi model klasifikasi berbasis teks, gambar, dan model gabungan (*multimodal*), diperoleh kesimpulan sebagai berikut:

1. *Multimodel deep learning* atau model CNN Gabungan dibangun berdasarkan model CNN berbasis teks (model CNN 1D) dan Model CNN berbasis gambar (model CNN 2D). Model gabungan kemudian dilatih menggunakan data gambar dan teks yang telah di *preprocessing* sejumlah 3103 sebagai data *training* dan 775 sebagai data *testing*. Model selanjutnya dilatih dalam 100 epoch dan 20 *early stopping steps* serta evaluasi menggunakan akurasi dan AUC. Hasil evaluasi menunjukkan terdapat indikasi *underfit* sehingga dilakukan pengembalian nilai pada epoch ke-33, sehingga diperoleh nilai akurasi 99,35% dan nilai AUC 99,81%. Nilai akurasi berada di atas 85% menunjukkan model baik dalam mengklasifikasikan data dalam model dan nilai AUC berada di atas 85% menunjukkan model baik dalam mengklasifikasikan data di luar model.
2. Hasil analisis perbandingan *multimodel deep learning* dengan arsitektur CNN terhadap *unimodel* CNN berbasi teks dan CNN berbasis gambar menunjukkan *multimodel deep leaning* lebih unggul. *Multimodel deep learning*, model CNN gabungan mampu mengungguli akurasi dan AUC *unimodel* berbasis teks SVM, Regresi Logistik, dan Model CNN 1D. Sementara model terbaik dari kelompok *unimodel*, model berbasis gambar, model CNN 2D, memiliki nilai akurasi tertinggi dibanding *unimodel* lainnya. Berdasarkan hasil di atas mengindikasikan bahwa pemanfaatan gambar foto yang telah dikategorikan lebih efektif dalam klasifikasi berita, dimana kemampuan visual dari gambar memberikan kontribusi besar dalam membedakan antara berita hoaks dan fakta.

5.2 Saran

Berdasarkan kesimpulan yang diperoleh selama proses membangun dan membandingkan model CNN dapat disarankan beberapa hal sebagai berikut:

1. Hasil Evaluasi model menunjukkan model CNN gabungan menunjukkan nilai mendekati sempurna dengan tingkat akurasi 99,35% yang dinilai terlalu tinggi. Sehingga diperlukan penelitian lebih lanjut untuk mengetahui apakah akurasi model tetap tinggi atau menurun dengan memperbanyak *epoch*, jumlah data *training*, dan *testing*.
2. Model CNN 2D dan gabungan menunjukkan nilai akurasi hampir sama namun berbeda jika dibandingkan dengan model CNN 1D. Hal tersebut menunjukkan terdapat kemungkinan dataset yang melibatkan gambar lebih mudah dikenali oleh mesin atau data mirip satu sama lain, sehingga disarankan untuk *training* model menggunakan data berita hoaks dan fakta berbasis teks yang lebih beragam untuk mengimbangi model berbasis gambar.

Halaman ini sengaja dikosongkan

DAFTAR PUSTAKA

(2023, April 6). Retrieved from Kominfo: https://www.kominfo.go.id/content/detail/48363/siaran-pers-no-50hmkominfo042023-tentang-triwulan-pertama-2023-kominfo-identifikasi-425-isu-hoaks/0/siaran_pers

Arintasari, S. A. (2023). "Klasifikasi Berita Pada Laman Website TurnBackHoax. id Menggunakan Algoritma Random Forest.". *(Doctoral dissertation, Institut Teknologi Sepuluh Nopember)*.

Asriyar, E. &. (2019). Aplikasi Machine Learning Issue Detection and Alert System Dengan Menggunakan Metode K-Nearest Neighbors Pada PT Andalabs. *JI-Tech*, 15(1), 1-7.

Assegaf, M. R. (2021). Klasifikasi Spesies Tanaman Monstera Berdasarkan Citra Daun Menggunakan Metode Convolutional Neural Network (Cnn). *eProceedings of Engineering*, 8(4).

de Souza Brito, A. V. (2021). Combining max-pooling and wavelet pooling strategies for semantic image segmentation. *Expert Systems with Applications*, 183, 115403.

Dilla, A. N. (2019). Komunikasi Persuasif dalam Kampanye Gerakan Anti Hoaks oleh Komunitas Mafindo Jakarta. *Koneksi*, 3(1), 199-206.

Djufri, M. (2020). Penerapan Teknik Web Scraping Untuk Penggalian Potensi Pajak (Studi Kasus Pada Online Market Place Tokopedia, Shopee Dan Bukalapak). *Jurnal BPPK: Badan Pendidikan Dan Pelatihan Keuangan*, 13(2), 65-75.

Dulkiah, M. &. (2020). Pola Penyebaran Hoaks pada Kalangan Mahasiswa Perguruan Tinggi Islam di Kota Bandung. *Jurnal SMART (Studi Masyarakat, Religi, Dan Tradisi)*, 6(2), 1-16.

Fahmi, R. N. (2021). Analisis Sentimen Pengguna Twitter Terhadap Kasus Penembakan Laskar FPI Oleh Polri Dengan Metode Naive Bayes Classifier. *JIKO (Jurnal Informatika dan Komputer)*, 5(2), 61-66.

Gu, J. W. (2018). Recent advances in convolutional neural networks. *Pattern recognition*, 77, 354-377.

Haqqo, A. &. (2023). Turnbackhoax. id: Upaya Pemutusan Disinformasi Berita-berita Pemilu 2024. *Indonesian Journal of Applied Linguistics Review*, 4(1), 9-18.

Harnum, R. P. (2022). Sosialisasi Buku Saku Tangkal Hoaks Covid-19 untuk Mahasiswa Sebagai Agent of Change. *Journal of Servite*, 4(2), 91-103.

Hawari, T. N. (2019). Analisis Kriminologis Penggunaan News Picture dalam Hoax yang Tersebar di Media Sosial (Analisis Isi Hoax pada Turnbackhoax. Id). *Deviance Jurnal kriminologi*, 3(2), 91-109.

Huroh, K. I. (2022). Penekanan Penyebaran hoax DI Media Sosial Sebagai upaya meningkatkan Persatuan Negara Indonesia. *Jurnal Kalacakra: Ilmu Sosial Dan Pendidikan*, 3(2), 72. https://doi.org/10.31002/kalacakra.v3i2.6377.

Kim, E. &. (2018). Multimodal deep learning using images and text for information graphic classification. *In Proceedings of the 20th International ACM SIGACCESS Conference on Computers and Accessibility*, (pp. 143-148).

Kurniawan, A. A. (2021). Implementasi Deep Learning Menggunakan Metode CNN dan LSTM untuk Menentukan Berita Palsu dalam Bahasa Indonesia. *Jurnal Informatika Universitas Pamulang*, 5(4), 544-552.

Li, L. L. (2020). A multi-modal method for satire detection using textual and visual cues. *arXiv preprint arXiv*, 2010.06671.

Lubis, A. R. (2021). The effect of the TF-IDF algorithm in times series in forecasting word on social media. Indones. *J. Electr. Eng. Comput.*, Sci, 22(2), 976.

Mazaya, V. (2019). Cyberdakwah sebagai Filter Penyebaran Hoax. *Islamic Communication Journal*, 4(1), 14-25.

Mishra, A. L. (2021). Accelerating sparse deep neural networks. *arXiv*, arXiv preprint arXiv:2104.08378.

Mouzannar, H. R. (2018). Damage Identification in Social Media Posts using Multimodal Deep Learning. *In ISCRAM*.

Muchtar, K. N. (2021). Pendeteksian Septoria pada Tanaman Tomat dengan Metode Deep Learning berbasis Raspberry Pi. *Jurnal RESTI (Rekayasa Sistem Dan Teknologi Informasi)*, 5(1), 107-113.

Mundzir, H. H., & Anshori, M. A. (2021). Sosialisasi Penangkalan Berita hoax di Perkumpulan Ibu-Ibu PKK Kelurahan Karang Besuki Kecamatan Sukun Kota Malang. *Jurnal Pengabdian Polinema Kepada Masyarakat*, 8(2), 38–43. https://doi.org/10.33795/jppkm.v8i2.88.

Nugroho, V. A. (2021). Klasifikasi jenis pemeliharaan dan perawatan container crane menggunakan algoritma machine learning. MATICS. *urnal Ilmu Komputer dan Teknologi Informasi (Journal of Computer Science and Information Technology)*, 13(1), 21-27.

Nurtikasari, Y. A. (2022). Analisis Sentimen Opini Masyarakat Terhadap Film Pada Platform Twitter Menggunakan Algoritma Naive Bayes. *INSOLOGI: Jurnal Sains dan Teknologi*, 1(4), 411-423.

Ofli, F. A. (2020). Analysis of social media data using multimodal deep learning for disaster response. *arXiv preprint arXiv*, 2004.11838.

Pan, Z. Z. (2021). Scalable vision transformers with hierarchical pooling. *In Proceedings of the IEEE/cvf international conference on computer vision*, (pp. 377-386).

Perdana, K. &. (2020). Identifikasi Berita Hoax dengan Recurrent Neural Network. *Jurnal Bangkit Indonesia*, 10(02), 14-16.

Pranesti, D. A. (2019). Pranesti, Dewi Ayu, and Ridwan Arifin. "Perlindungan Korban dalam Kasus Penyebaran Berita Hoax di Media Sosial di Indonesia. *Jurnal Hukum Media Bhakti*.

Prasanth, S. N. (2022). CEN-Tamil@ DravidianLangTech-ACL2022: Abusive comment detection in Tamil using TF-IDF and random kitchen sink algorithm. *Proceedings of the Second Workshop on Speech and Language Technologies for Dravidian Languages*, (pp. 70-74).

Qaiser, S. &. (2018). Text mining: use of TF-IDF to examine the relevance of words to documents. *International Journal of Computer Applications*, 181(1), 25-29.

Qayyum, R. (2023, July 8). *Introduction To Pooling Layers In CNN*. Retrieved from Medium.com: https://pub.towardsai.net/introduction-to-pooling-layers-in-cnn-dafe61eabe34

Rahmaniar, F. R. (2019). Perbandingan Metode K-Means Dan Fuzzy C-Means Untuk Pengelompokan Tweet Yang Ditujukan Kepada PT. Kereta Api Indonesia (@ KAI121) Dengan Menggunakan N-Gram (Doctoral dissertation, Institut Teknologi Sepuluh Nopember).

Ram, A. &.-A. (2020). The relationship between Fully Connected Layers and number of classes for the analysis of retinal images. *arXiv preprint arXiv*, 2004.03624.

Ramayasa, I. P. (2023). PERBANDINGAN METODE VEKTORISASI PADA ANALISA SENTIMENT, STUDI KASUS: CYBERBULLYING PADA KOMENTAR INSTAGRAM. *Jurnal Teknologi Informasi dan Komputer*, 9(5).

Retnoningsih, E. &. (2020). Mengenal Machine Learning Dengan Teknik Supervised Dan Unsupervised Learning Menggunakan Python. *Bina Insani Ict Journal*, 7(2), 156-165.

Ridwan, A. (2020). Penerapan Algoritma Naïve Bayes Untuk Klasifikasi Penyakit Diabetes Mellitus. *J. SISKOM-KB (Sistem Komput. dan Kecerdasan Buatan)*, 4(1), 15-21.

Sewak, M. K. (2018). *Practical convolutional neural networks: implement advanced deep learning models using Python.* Packt Publishing Ltd.

Siahaan, C. N., & Tampubolon, M. (2021). Strategy for identification of hoax news in digital media in facing case of racism of Papua students in Indonesia. *Technium Social Sciences Journal*, 21, 457–467. https://doi.org/10.47577/tssj.v21i1.3959 .

Simarmata, J. I. (2019). *Hoaks dan media sosial: saring sebelum sharing.* Yayasan Kita Menulis.

Sugiarto, N. K. (2021). PEMODELAN HYBRID CONVOLUTIONAL BACKPROPAGATION NEURAL NETWORK UNTUK PERAMALAN BEBAN JANGKA SANGAT PENDEK BERDASARKAN MINIMALISASI BIAYA LISTRIK. *JURNAL TEKNIK ELEKTRO*, 10(2), 463-472.

Witten, D. &. (2021). *An introduction to statistical learning with applications in R (Second Edition).* springer publication.

Yuliani, S. S. (2019). Hoax News Classification using Machine Learning Algorithms. *International Journal of Engineering and Advanced Technology*, 9(2), 2249-8958.

Yunanto, R. P. (2021). Survei Literatur: Deteksi Berita Palsu Menggunakan Pendekatan Deep Learning. *Jurnal Manajemen Informatika (JAMIKA)*, 11(2), 118-130.

Yunus, M. (2023, Juli Sabtu). *#11 Artificial Neural Network (ANN) — Part 6 Konsep Dasar Convolutional Neural Network (CNN).* Retrieved from Medium: https://yunusmuhammad007.medium.com/11-artificial-neural-network-ann-part-6-konsep-dasar-convolutional-neural-network-cnn-3cc10fd9cf69

Zaen, Z. A. (2023). ANALISIS FRAMING TENTANG PANDAWARA GROUP DAN PEMERINTAH SUKABUMI TERKAIT PANTAI LOJI PADA MEDIA ONLINE KOMPAS. COM DAN REPUBLIKA. CO. ID. *Triwikrama: Jurnal Ilmu Sosial*.

Zhang, A. L. (2021). *Dive into deep learning.* arXiv preprint arXiv:2106.11342.

Zhou, J. L. (2021). Crop disease identification and interpretation method based on multimodal deep learning. *Computers and Electronics in Agriculture*, 189, 106408.

LAMPIRAN

Lampiran 1. *Data Hasil Scrapping*

No	Link	Judul	LinkImg	Tgl	Paragraf (Konten)
0	https://turnbackhoax.id/2023/10/13/salah-kantor-pusat-nasdem-dibakar-akibat-surya-paloh-ingkar-janji/	[SALAH] KANTOR PUSAT NASDEM DIBAKAR AKIBAT SURYA PALOH INGKAR JANJI	https://turnbackhoax.id/wp-content/uploads/2023/10/Klaim-10-678x381.jpg	Oktober 13, 2023	['Hasil periksa fakta Pekik Jalu Utomo.Tidak ditemukan informasi kredibel terkait klaim narasi yang beredar. Thumbnail yang menampilkan bangunan terbakar tersebut merupakan Kantor Inspektorat dan Kominfo Natuna dan bukan kantor pusat Partai NasDem.', '=======', '[KATEGORI]: Konten yang dimanipulasi', '=======', '[SUMBER]:\xa0https://archive.cob.web.id/archive/1696845109.922199/index.html\xa0(Youtube)', '=======', '[NARASI]:KANTOR PUSAT NASDEM DI BAK4R !! AKIBAT SURYA PALOH INGKAR JANJI', 'BREAKING NEWS …
1	https://turnbackhoax.id/2023/10/13/salah-anies-dan-surya-paloh-ngemis-jadi-cawapres-ganjar/	[SALAH] ANIES DAN SURYA PALOH NGEMIS JADI CAWAPRES GANJAR	https://turnbackhoax.id/wp-content/uploads/2023/10/Klaim-9-678x381.jpg	Oktober 13, 2023	['Hasil periksa fakta Pekik Jalu Utomo.Faktanya tidak ditemukan pemberitaan terkait klaim narasi yang beredar. Selain thumbnail merupakan hasil rekayasa, Video tersebut hanya berisi cuplikan dari beberapa peristiwa berbeda yang tidak berkaitan.', '=======', '[KATEGORI]: Konten yang dimanipulasi', '=======', '[SUMBER]:\xa0https://archive.cob.web.id/archive/1696835985.598299/index.html\xa0(Youtube)', '=======', '[NARASI]:TAK PUNYA HARGA DIRI !! ANIES DAN PALOH NGEMIS-NGEMIS MAU JADI CAWAPRES GANJAR PRANOWO…
2	https://turnbackhoax.id/2023/10/13/salah-video-metro-tv-mempromosikan-situs-judi-online/	[SALAH] Video “Metro TV mempromosikan situs judi online”	https://turnbackhoax.id/wp-content/uploads/2023/10/Screenshot_200-678x381.png	Oktober 13, 2023	['Informasi palsu. Faktanya, di kanal Youtube dan akun Facebook serta situs resmi milik Metro TV, tidak ditemukan adanya konten berita yang mempromosikan situs judi online seperti yang diunggah sumber klaim.', 'Selengkapnya di bagian PENJELASAN dan REFERENSI.==========Kategori: Konten yang Menyesatkan==========', 'Akun Facebook Lucifer…
…	…	…	…	…	…
5163	https://ihram.republika.co.id/berita/rnt7oc313/utang-budi-kedokteran-modern	Utang Budi Kedokteran Modern	https://static.republika.co.id/uploads/images/inpicture_slide/dokter-muslim-saat-mengobati-pasien-ilustrasi-_120612210522-497.jpg	Ahad 01 Jan 2023 21:35 WIB	['IHRAM.CO.ID,\xa0Konsep kesehatan mental atau al-tibb al-ruhani pertama kali diperkenalkan dunia kedokteran Islam oleh seorang dokter dari Persia bernama Abu Zayd Ahmed ibnu Sahl al-Balkhi (850-934). Dalam kitabnya berjudul Masalih al-Abdan wa al-Anfus (Makanan untuk Tubuh dan Jiwa), al-Balkhi berhasil menghubungkan penyakit antara tubuh dan jiwa. Ia biasa menggunakan istilah al-Tibb al-Ruhani untuk menjelaskan keseharan spritual dan kesehatan psikologi.', 'Sedangkan untuk\xa0 kesehatan mental dia kerap menggunakan istilah Tibb al-Qalb. Ia pun sangat terkenal dengan teori yang dicetuskannya tentang kesehatan jiwa yang berhubungan dengan tubuh. …
5164	https://ekonomi.republika.co.id/berita/rnt9jv490/alih-fungsi-lahan-pertanian-tembus-100-ribu-hektare-per-tahun	Alih Fungsi Lahan Pertanian Tembus 100 Ribu Hektare per Tahun	https://static.republika.co.id/uploads/images/inpicture_slide/042501800-167103100-830-556.jpg	Ahad 01 Jan 2023 21:28 WIB	['REPUBLIKA.CO.ID,\xa0JAKARTA -- Kementerian Pertanian (Kementan) mencatat, alih fungsi lahan pertanian mencapai kisaran 90 ribu\xa0hingga 100 ribu hektare per tahun. Konversi lahan pertanian itu menjadi salah satu ancaman terhadap sektor pertanian dalam meningkatkan produksi untuk memenuhi kebutuhan masyarakat.', '"Konversi lahan sangat mengkhawatirkan.\xa0Ini kalau tidak diatasi akan menurunkan produksi," kata Kepala Balai Besar Litbang Sumberdaya Lahan Pertanian Kementan,…

Lampiran 2. Sintaksis *Library* Python

```
import requests
from bs4 import BeautifulSoup
import re
import tensorflow as tf
from tensorflow import keras
from tensorflow.keras import layers
from tensorflow.keras.preprocessing.image import ImageDataGenerator
from google.colab import files
import os
import numpy as np
import matplotlib.pyplot as plt
import pandas as pd
from sklearn.preprocessing import LabelEncoder
from sklearn.model_selection import train_test_split
from tensorflow.keras.preprocessing.image import load_img, img_to_array
import cv2
import nltk
from Sastrawi.Stemmer.StemmerFactory import StemmerFactory
from sklearn.feature_extraction.text import TfidfVectorizer
from nltk.corpus import stopwords
plt.style.use('ggplot')
from sklearn.feature_extraction.text import TfidfVectorizer
import statistics
import math
from nltk.probability import FreqDist
import seaborn as sns
import scipy.stats as stats
from sklearn.svm import SVC
from sklearn.tree import DecisionTreeClassifier
from sklearn.linear_model import LogisticRegression
from sklearn.neighbors import KNeighborsClassifier
from sklearn.neural_network import MLPClassifier
from sklearn.metrics import roc_curve, roc_auc_score
from sklearn.model_selection import GridSearchCV, RandomizedSearchCV
from sklearn.metrics import accuracy_score, confusion_matrix
from tensorflow.keras.layers import concatenate
from tensorflow.keras.callbacks import ModelCheckpoint
from tensorflow.keras.applications import VGG16
from tensorflow.keras.models import load_model
from tensorflow.keras.regularizers import l2
from tensorflow.keras.models import Sequential
from sklearn.metrics import confusion_matrix, precision_score, recall_score,
f1_score
from tensorflow.keras.layers import BatchNormalization
import pickle
```

Lampiran 3. Sintasis *Scrapping* Python

```
import requests
from bs4 import BeautifulSoup
base_url = "https://turnbackhoax.id/"
data = []
for year in range(2023, 2022, -1):
  print("^"*100)
  for month in range(12, 0, -1):
    print("&"*100)
    for day in range(30, 0, -1):
      print("*"*100)
      url = f"{base_url}{year:04d}/{month:02d}/{day:02d}/"
      print (url)
      response = requests.get(url)
      if response.status_code == 404:
        continue # lanjutkan loop jika laman tidak ditemukan
      # Parsing HTML response
      html = BeautifulSoup(response.text, 'html.parser')
      # Mencari semua artikel berita
      articles = html.find_all('article')
      for article in articles:
        print("()"*100)
        # Mencari link artikel
        link = article.find('a')
        if link:
          content_link = link.get("href")
          response = requests.get(content_link)
          # Parsing HTML response
          html_c = BeautifulSoup(response.text, 'html.parser')
                    # Mencari semua artikel berita
          # Inisialisasi variabel-variabel ke None
          title = None
          tag = None
          image = None
          date = None
          try:
            title = html_c.find('h1', class_="entry-title").get_text()
          except AttributeError:
            title = None
          try:
            tag = html_c.find('span', class_="entry-meta-categories").get_text()
          except AttributeError:
                      tag = None
```

Lampiran 3. Sintasis *Scrapping* Python (lanjutan)

```
            try:
              image_element = html_c.find('figure', class_='entry-
thumbnail').find('img')
              image = image_element['src'] if image_element else None
            except AttributeError:
              image = None
            try:
              date = html_c.find('span', class_='entry-meta-date
updated').find('a').get_text()
            except AttributeError:
              date = None
            # Mencari elemen dengan kelas "entry-content" dan mengambil paragrafnya
            entry_content = html_c.find(class_='entry-content')
            paragraphs = entry_content.find_all('p')
            # Merapikan paragraf
            clean_paragraphs = [p.get_text().strip() for p in paragraphs if
p.get_text().strip()]
            # Menyimpan data dalam bentuk dictionary
            article_data = {
                          'link': content_link,
                          'title': title,
                          'tag': tag,
                          'image': image,
                          'date': date,
                          'paragraphs': clean_paragraphs
                      }
            print(content_link)
            print(title)
            print(image)
            print(tag)
            print(image_element['src'])
            print(date)
            print(clean_paragraphs)
            print('\n')
            data.append(article_data)
```

Lampiran 4. Sintaksi *Preprocessing* Gambar dan Pembagian *Training-Testing*

```
# Mengubah data ukuran data gambar
width = 224
height = 224
batch_size = 4
aug_rps_train = ImageDataGenerator(
    rescale=1./255,
    preprocessing_function=lambda x: tf.image.resize_with_pad(x,
target_height=height, target_width=width),
    validation_split=0.2)
aug_rps_val = ImageDataGenerator(
    rescale=1./255,
    validation_split=0.2)
# membuat data trainig dan testing
train_gen = aug_rps_train.flow_from_directory(
    '/content/drive/MyDrive/Colab
Notebooks/ml_storage/TA/turnbackhoax_2023_2022',
    target_size= (height, width),
    batch_size=batch_size,
    class_mode='binary',
    shuffle=False,
    subset='training')
val_gen = aug_rps_val.flow_from_directory(
    '/content/drive/MyDrive/Colab
Notebooks/ml_storage/TA/turnbackhoax_2023_2022',
    target_size= (height, width),
    batch_size=batch_size,
    class_mode='binary',
    shuffle=False,
    subset='validation')
# Mengubah gambar menjadi array untuk data gambar
# run ulang dari atas jika ValueError: could not broadcast input array from shape
(224,224,3) into shape (150,200,3)
X_train_images = []
X_train_filenames = []
X_test_images = []
X_test_filenames = []
i = 0
k=0
# iterations = 1  # Ganti dengan jumlah iterasi yang diinginkan
# Mengambil data dari train_gen
```

Lampiran 4. Sintaksi *Preprocessing* Gambar dan Pembagian *Training-Testing* (Lanjutan)

```
for _ in range(len(train_gen)):
    i += 1
    print(i)
    images, labels = next(train_gen)
    filenames = train_gen.filenames  # Mendapatkan nama file dari generator
    X_train_images.extend(images)
    X_train_filenames.extend(filenames)  # Menambahkan nama file ke dalam array
# Mengambil data dari val_gen
for _ in range(len(val_gen)):
    k += 1
    print(k)
    images, labels = next(val_gen)
    filenames = val_gen.filenames  # Mendapatkan nama file dari generator
    X_test_images.extend(images)
    X_test_filenames.extend(filenames)  # Menambahkan nama file ke dalam
# # Simpan array NumPy ke dalam file
# np.save('X_train_images_fix2.npy', X_train_images)
# np.save('X_test_images_fix2.npy', X_test_images)
# Membaca data dari file
X_train_images_np = np.load('/content/drive/MyDrive/Colab
Notebooks/ml_storage/TA/X_train_images_fix2.npy')
X_test_images_np = np.load('/content/drive/MyDrive/Colab
Notebooks/ml_storage/TA/X_test_images_fix2.npy')
# Inisialisasi DataFrame
df_gen_train = pd.DataFrame({'image_path': train_gen.filepaths, 'label':
train_gen.labels})
df_gen_val = pd.DataFrame({'image_path': val_gen.filepaths, 'label':
val_gen.labels})
# Memisahkan nama file dan indeks
df_gen_train[['image_name', 'index_terpilih']] =
df_gen_train['image_path'].str.rsplit('_', n=1, expand=True)
# Konversi kolom 'index_terpilih' menjadi integer
df_gen_train['index_terpilih'] =
df_gen_train['index_terpilih'].str.extract('(\d+)').astype(int)
# Mengatur kolom 'image_path' menjadi indeks DataFrame
df_gen_train.set_index('index_terpilih', inplace=True)
# Memisahkan nama file dan indeks
df_gen_val[['image_name', 'index_terpilih']] =
df_gen_val['image_path'].str.rsplit('_', n=1, expand=True)
# Konversi kolom 'index_terpilih' menjadi integer
df_gen_val['index_terpilih'] =
df_gen_val['index_terpilih'].str.extract('(\d+)').astype(int)
# Mengatur kolom 'image_path' menjadi indeks DataFrame
df_gen_val.set_index('index_terpilih', inplace=True)
```

Lampiran 5. Sintaksi *Preprocessing* Teks dan Pembagian *Training-Testing*

```
massa_regex =
r'\b(kilogram|kg|ton|t|kiloton|kt|megaton|mt|gigaton|gt|kuintal|kwintal|kg|gram|g
|mg|miligram|pound|lb|ounce|oz|stone|st)\b'
jarak_regex =
r'\b(meter|m|kilometer|km|sentimeter|cm|milimeter|mm|mil|mile|yard|kaki|inchi|inc
i|thou|leauge|hektar|hektare|hasta|feet)\b'
volume_regex = r'\b(liter|l|galon|gallon|qt|kuintal|kwintal|ml)\b'
suhu_regex = r'\b(celsius|c|fahrenheit|f|kelvin|k)\b'

def bersih_bersih(sentence):
  sentence = sentence.lower()
  sentence = re.sub(r'(REFRENCE|REFRENSI|RFERENSI|RFRENSI).*https://\S+', '',
sentence, flags=re.DOTALL | re.IGNORECASE)
  sentence = re.sub(r'REFERENSI', '', sentence)  # Menghapus kata "REFERENSI"
  sentence = re.sub(r'\\n', ' ', sentence)  # Menghapus newline (\n)
  unicode_chars_to_remove = ['\\xa0', '\\u2063', '\\u2063\'', '\\u2022',
'\\u2013', '\\u2026'] # Daftar karakter Unicode yang ingin dihapus
  for char in unicode_chars_to_remove:
      sentence = sentence.replace(char, '') # Menghapus karakter Unicode yang
tidak diinginkan
  sentence = re.sub(r'^.*\[narasi\]:?["“‘]?(.*?)["”’]?(?=\[|\Z)', '', sentence,
flags=re.DOTALL) # Menghapus teks sebelum [narasi]
  sentence = re.sub(r'^(\\\\|-\\\\|-)\s*', '', sentence) # Menghapus kalimat yang
diawali dengan '\\', '-\\', atau '-'
  sentence = re.sub(r'@\w+\s*', '', sentence) # Menghapus mention (pemberian tag)
  sentence = re.sub(r'#\w+\s*', '', sentence)# Menghapus hashtag
  sentence = re.sub(r'\d{4}-\d{2}-\d{2}\s\d{2}:\d{2}:\d{2}\.','',sentence) #
Menghilangkan format tanggal dan waktu
  sentence = re.sub(r'\d{2}\s\w{3,}\s\d{4}.','', sentence) # Menghilangkan format
tanggal "XX NAMA BULAN TAHUN"
  sentence =
re.sub(r'(menit|mnt|thn|tahun|tahunan|minggu|mingguan|mg|hari|harian|hr|jam|jm|de
tik|dtk|sekon|bulan|bulanan|bln)*','', sentence) # Menghilangkan satuan waktu
  sentence = re.sub(r'(\d{1,}\s*gb|\d{1,}\s*kb|\d{1,}\s*mb|\d{1,}\s*tb|lte)',"",
sentence) # Menghilangkan satuan byte dan kata lte
  sentence =
re.sub(r'(ribu|ribuan|rb|jt|juta|jutaan|milyar|miliaran|miliar|triliun|triliunan|
trilyun|rp|rupiah|dolar|yen|ringgit)',"", sentence) # Menghilangkan satuan uang
  sentence = re.sub(r'\w*\.*\w{1,}\.*\/\w{1,}','',sentence) # Menghilangkan
pecahan
  sentence = re.sub(r'rp\s*\d{1,}\s','',sentence) # Menghilangkan jumlah tarif
  sentence = re.sub(massa_regex, '', sentence, flags=re.IGNORECASE)
  sentence = re.sub(jarak_regex, '', sentence, flags=re.IGNORECASE)
```

Lampiran 5. Sintaksi *Preprocessing* Teks dan Pembagian *Training-Testing* (lanjutan)

```
  sentence = re.sub(volume_regex, '', sentence, flags=re.IGNORECASE)
  sentence = re.sub(suhu_regex, '', sentence, flags=re.IGNORECASE)
  sentence = re.sub(r"\*\d{3,}\*\d{3,}\#","", sentence) # Menghilangkan kode
aktivasi-1
  sentence = re.sub(r"\*\d{3,}\#","", sentence) # Menghilangkan kode aktivasi-2
  sentence = re.sub(r'https?\S+', '', sentence)  # Menghilangkan URL
  sentence = re.sub(r"https?://\S*|www\.\S+","", sentence) # Menghilangkan web
  sentence = re.sub(r'(\d{1,}\.*\d{0,})','', sentence) # Menghilangkan angka
  sentence = re.sub(r'[\U0001F600-\U0001F6FF]|[\U0001F300-
\U0001F5FF]|[\U0001F900-\U0001F9FF]|[\U0001F1E0-\U0001F1FF]|[\U00002600-
\U000027BF]|[\U0001F100-\U0001F1FF]|[\U0001F600-\U0001F64F]|[\U0001F680-
\U0001F6FF]|[\U0001F1E6-\U0001F1FF]|[\U0001F910-\U0001F96B]|[\U0001F980-
\U0001F9E0]',
                     '', sentence)  # Menghilangkan emoticon
  sentence = re.sub(r'[\u4E00-\u9FFF\u1100-\u11FF\u3130-\u318F\uAC00-
\uD7AF\u30A0-\u30FF]+', '', sentence)# Menghapus teks dalam bahasa Tionghoa
(Cina), Korea, dan Jepang
  sentence = re.sub(r'[\u0600-\u06FF\u0750-\u077F\u08A0-\u08FF\uFB50-
\uFDFF\uFE70-\uFEFF]+', '', sentence)# Menghapus teks dalam bahasa Arab
  exclude_punctuation = set(string.punctuation).union(set([chr(i) for i in [34,
39, 8220, 8221, 8212, 8230, 8216, 8217, 8211 ]])).union(set(chr(j) for j in
range(8208,8332))) # Menghapus tanda baca dari judul artikel, kecuali tanda kutip
dan tanda petik dua
  sentence = ''.join(char for char in sentence if char not in
exclude_punctuation)
  jenis_kata =
f'{kata_depan_tempat}|{kata_depan_arah}|{kata_depan_waktu}|{kata_depan_asal}|{kat
a_depan_tujuan}|{kata_depan_perbandingan}|{kata_depan_sebab_akibat}|{kata_depan_l
ain}|{kata_keterangan}|{kata_ganti_objek}|{kata_ganti_subjek}|{kata_hubung}' #
Menggabungkan semua kata depan dalam satu ekspresi reguler
  sentence = re.sub(jenis_kata, '', sentence) # Menghapus kata depan dari teks
  singkatan_pattern = r'\b(?:' + '|'.join(singkatan_to_remove) + r')\b' # regex
untuk menghapus singkatan-singkatan
  sentence = re.sub(singkatan_pattern, '', sentence, flags=re.IGNORECASE) # Hapus
singkatan dengan regex
  sentence = re.sub(r'\b\w\b', '', sentence) # Menghapus kata-kata dengan satu
karakter
  sentence = re.sub(r'\b\w{26,}\b', '', sentence)# hapus kata lebih dari 25
characters
  # sentence = re.sub(r'\s+', ' ', sentence)  # Mengganti multiple spaces menjadi
single space
  sentence = sentence.strip()
  print(sentence + "---------berhasil-----------" + '\n')
  return sentence
```

Lampiran 5. Sintaksi *Preprocessing* Teks dan Pembagian *Training-Testing* (lanjutan)

```
df['text_bersih'] = df['paragraphs'].apply(bersih_bersih)
# proses Lemmatizing
# create stemmer
factory = StemmerFactory()
stemmer = factory.create_stemmer()
# Fungsi Lemmatizer
i = 0
def lemmatizer(sentence):
  global i
  i += 1
  output = stemmer.stem(sentence)
  print(str(i) + ": " + output + " ---------berhasil----------" + '\n')
  return output
df['text_bersih'] = df['text_bersih'].apply(lemmatizer)
# Data training dan testing menyesuaikan dengan data training dan testing gambar
yang terpilih
# Menghapus ekstensi file dari daftar nama file training
train_img_files = [os.path.splitext(os.path.basename(filepath))[0] for filepath
in train_gen.filenames]
# Menghapus ekstensi file dari daftar nama file validasi
test_img_files = [os.path.splitext(os.path.basename(filepath))[0] for filepath in
val_gen.filenames]
# Menghapus ekstensi file dan hanya menyisakan nomor setelah tanda "_"
train_img_files = [re.search(r'_(\d+)',
os.path.splitext(os.path.basename(filepath))[0]).group(1) for filepath in
train_gen.filenames]
test_img_files = [re.search(r'_(\d+)',
os.path.splitext(os.path.basename(filepath))[0]).group(1) for filepath in
val_gen.filenames]
train_indices = [int(idx) for idx in train_img_files]
test_indices = [int(idx) for idx in test_img_files]
df_terpilih_train = df[df.index.isin(train_indices)]
df_terpilih_test = df[df.index.isin(test_indices)]
df_terpilih_train = df_terpilih_train[['title','text_bersih']]
df_terpilih_test = df_terpilih_test[['title','text_bersih']]
df_terpilih_train['index_terpilih'] = df_terpilih_train.index
df_terpilih_test['index_terpilih'] = df_terpilih_test.index
# Gabungkan
train_df = df_gen_train.merge(df_terpilih_train, on='index_terpilih')
test_df = df_gen_val.merge(df_terpilih_test, on='index_terpilih')
# Data Mengubah data training dan testing Kedalam vektor dangan TF-IDF
max_features = 8900
train_tfidf_vectorizer = TfidfVectorizer(max_features=max_features)  # Sesuaikan
```

Lampiran 5. Sintaksi *Preprocessing* Teks dan Pembagian *Training-Testing* (lanjutan)

```
jumlah fitur TF-IDF sesuai kebutuhan
train_tfidf_matrix =
train_tfidf_vectorizer.fit_transform(train_df['text_bersih'])
max_features = 8900
test_tfidf_vectorizer = TfidfVectorizer(max_features=max_features)  # Sesuaikan
jumlah fitur TF-IDF sesuai kebutuhan
test_tfidf_matrix = test_tfidf_vectorizer.fit_transform(test_df['text_bersih'])
# Label encoding untuk label
label_encoder = LabelEncoder()
X_train = train_tfidf_matrix
y_train = label_encoder.fit_transform(train_df['label'])
X_test = test_tfidf_matrix
y_test = label_encoder.fit_transform(test_df['label'])
# Array dari X_train, X_test, y_train, dan y_test
X_train_np = X_train.toarray()
X_test_np = X_test.toarray()
X_train_reshaped = X_train_np.reshape(X_train_np.shape[0], X_train_np.shape[1],
1)
X_test_reshaped = X_test_np.reshape(X_test_np.shape[0], X_test_np.shape[1], 1)
# Input untuk data teks
max_sequence_length = X_train_reshaped.shape[1]  # Sesuaikan dengan panjang
maksimum urutan teks
embedding_dim = X_train_reshaped.shape[2]  # Sesuaikan dengan dimensi embedding
print("Bentuk train_tfidf_vectorizer:",
train_tfidf_vectorizer.transform(train_df['text_bersih']).shape)
print("Bentuk test_tfidf_vectorizer:",
test_tfidf_vectorizer.transform(test_df['text_bersih']).shape)
print("Bentuk X_train:", X_train.shape)
print("Bentuk X_test:", X_test.shape)
print("Bentuk X_train_np:", X_train_np.shape)
print("Bentuk X_test_np:", X_test_np.shape)
print("Bentuk X_train_reshaped:", X_train_reshaped.shape)
print("Bentuk X_test_reshaped:", X_test_reshaped.shape)
print("Bentuk y_train:", y_train.shape)
print("Bentuk y_test:", y_test.shape)
```

Lampiran 6. Sintaksi *Multimodal* dan *Unimodal*

```
# Mendefinisikan EarlyStopping callback
early_stopping = EarlyStopping(monitor='val_accuracy',  # Memantau val_loss atau
val_accuracy untuk early stopping
                               patience=20,          # Jumlah epoch tanpa
perbaikan sebelum berhenti
                               restore_best_weights=True,  # Mengembalikan bobot
terbaik pada model
                               verbose=1)           # Menampilkan pesan jika
terjadi early stopping
# true Positif (tp) # True Negatif (tn) # False positif (fp) # False Negatif (fn)
class LossAccHistory(tf.keras.callbacks.Callback):
    def __init__(self, file_path):
        self.history = {
            'loss': [], 'accuracy': [],
            'val_loss': [], 'val_accuracy': [],
            'tp': [], 'tn': [],
            'fp': [], 'fn': [],
            'val_tp': [], 'val_tn': [],
            'val_fp': [], 'val_fn': [], 'auc':[], 'val_auc':[]}
        self.file_path = file_path
    def on_epoch_end(self, epoch, logs=None, validation_data=None):
        # Update the history dictionary
        self.history['loss'].append(logs['loss'])
        self.history['accuracy'].append(logs['accuracy'])
        self.history['val_loss'].append(logs['val_loss'])
        self.history['val_accuracy'].append(logs['val_accuracy'])
        self.history['tp'].append(logs['true_positives'])
        self.history['tn'].append(logs['true_negatives'])
        self.history['fp'].append(logs['false_positives'])
        self.history['fn'].append(logs['false_negatives'])
        self.history['val_tp'].append(logs['val_true_positives'])
        self.history['val_tn'].append(logs['val_true_negatives'])
        self.history['val_fp'].append(logs['val_false_positives'])
        self.history['val_fn'].append(logs['val_false_negatives'])
        self.history['auc'].append(logs['auc'])
        self.history['val_auc'].append(logs['val_auc'])
        # Simpan history ke dalam file menggunakan pickle
        with open(self.file_path, 'wb') as file:
            pickle.dump(self.history, file)
```

Lampiran 6. Sintaksi *Multimodal* dan *Unimodal*

```
pretrained_vgg = VGG16(include_top=False, weights='imagenet', input_shape=(width,
height, 3)) # Load the VGG16 model
for layer in pretrained_vgg.layers:
    layer.trainable = False # Freeze the layers

# Model Conv2D
model_conv2d = tf.keras.models.Sequential([
                                    pretrained_vgg,
                                    tf.keras.layers.Flatten(),
                                    tf.keras.layers.Dense(512, activation='relu',
kernel_regularizer=l2(0.05))])
# Model Conv1D
model_conv1d = tf.keras.models.Sequential([
    tf.keras.layers.Conv1D(128, 5, activation='relu',
input_shape=(max_sequence_length, embedding_dim)),
    tf.keras.layers.BatchNormalization(),
    tf.keras.layers.Conv1D(128, 5, activation='relu'),
    tf.keras.layers.BatchNormalization(),
    tf.keras.layers.GlobalMaxPooling1D(),
    tf.keras.layers.Flatten(),
    tf.keras.layers.Dense(512, activation='relu')])

# Model Gabungan
# Gabungkan output menggunakan lapisan concatenate
merged = concatenate([output_conv2d, output_conv1d])


# Lapisan-lapisan Dense Menyesuaikan (model_conv2d, model_conv1d, atau merged)
x = tf.keras.layers.Dense(512, activation='relu')(merged)#ganti merged
x = tf.keras.layers.BatchNormalization()(x)
x = tf.keras.layers.ReLU()(x)

# Output layer
output = tf.keras.layers.Dense(1, activation='sigmoid', name='output')(x)

# Buat Output model2D
model2d = tf.keras.Model(inputs= model_conv2d.input,
                         outputs=output)
# Buat Output model2D
model1d = tf.keras.Model(inputs= model_conv1d.input,
                         outputs=output)
```

Lampiran 6. Sintaksi *Multimodal* dan *Unimodal* (Lanjutan)

```
# Buat Output model gabungan
combined_model = tf.keras.Model(inputs=[model_conv2d.input, model_conv1d.input],
outputs=output) # inputs menyesuaikan (model_conv2d, model_conv1d, atau merged)

# Compile model
# Compile menyesuaikan model1d, model2d dan combine_model
combined_model.compile(loss='binary_crossentropy',
                       optimizer='adam',
                       metrics=['accuracy',
                        tf.keras.metrics.TrueNegatives(name='true_negatives'),
                        tf.keras.metrics.TruePositives(name='true_positives'),
                        tf.keras.metrics.FalsePositives(name='false_positives'),
                        tf.keras.metrics.FalseNegatives(name='false_negatives'),
                        tf.keras.metrics.AUC(name='auc')
                       ])

# Membuat instance dari LossAccHistory dengan menyimpan history ke dalam file
'history.pkl'
loss_acc_history_comb = LossAccHistory(file_path='[berinama].pkl')

# Train model
# Model Fit menyesuaikan model1d, model2d dan combine_model
model2d.fit(X_train_images_np, y_train, epochs=100, batch_size=32,
validation_data=(X_test_images_np, y_test), callbacks=[loss_acc_history_2D,
early_stopping])

# Fungsi menampilkan Hasil training dan testing
def plot_metric(history, metric, log_scale=False, smooth_window=None,
vertical_line_epoch=None):
    if smooth_window:
        smooth_metric = np.convolve(history[metric],
np.ones(smooth_window)/smooth_window, mode='valid')
        plt.plot(range(1, len(smooth_metric) + 1), smooth_metric,
label=f'Training {metric.capitalize()} (Smoothed)')
    else:
        plt.plot(range(1, len(history[metric]) + 1), history[metric],
label=f'Training {metric.capitalize()}')
    val_metric = f'val_{metric}'
    if val_metric in history:
        if smooth_window:
            smooth_val_metric = np.convolve(history[val_metric],
np.ones(smooth_window)/smooth_window, mode='valid')
            plt.plot(range(1, len(smooth_val_metric) + 1), smooth_val_metric,
label=f'Validation {metric.capitalize()} (Smoothed)')
```

Lampiran 6. Sintaksi *Multimodal* dan *Unimodal* (Lanjutan)

```
        else:
            plt.plot(range(1, len(history[val_metric]) + 1), history[val_metric],
label=f'Validation {metric.capitalize()}')
    if vertical_line_epoch:
        plt.axvline(x=vertical_line_epoch, color='r', linestyle='--',
label=f'Epoch {vertical_line_epoch}')
        # Tampilkan nilai metric pada epoch yang berpotongan dengan garis
vertical
        intersect_value_train = history[metric][vertical_line_epoch - 1]
        intersect_value_val = history[val_metric][vertical_line_epoch - 1]
        plt.text(vertical_line_epoch, intersect_value_train,
f'{intersect_value_train:.4f}', color='r')
        plt.text(vertical_line_epoch, intersect_value_val,
f'{intersect_value_val:.4f}', color='dodgerblue')

    plt.title(f'Training and Validation {metric.capitalize()}')
    plt.xlabel('Epochs')
    plt.ylabel(metric.capitalize())
    if log_scale:
        plt.yscale('log')
    plt.legend()
    plt.show()
# Tampilkan
metrics_to_plot = ['accuracy', 'auc']
for metric in metrics_to_plot:
    plot_metric(history.history, metric, log_scale=False, smooth_window=3,
vertical_line_epoch=5)
```

Lampiran 7. Output Model Pada Pengembalian Bobot Terbaik

Metrik	Model CNN 1D	Model CNN 2D	Model CNN Gabungan
loss	0.396	0.204	0.11
accuracy	0.83	0.98	0.991
val_loss	0.437	0.183	0.096
val_accuracy	0.823	0.985	0.994
tp	1127.0	1353.0	1378.0
tn	1449.0	1687.0	1697.0
fp	264.0	26.0	16.0
fn	263.0	37.0	12.0
val_tp	277.0	343.0	345.0
val_tn	361.0	420.0	425.0
val_fp	67.0	8.0	3.0
val_fn	70.0	4.0	2.0
auc	0.902	0.997	0.999
val_auc	0.882	0.998	0.998

Keterangan:

val_ : Ouput pada proses validasi atau *testing*

tp : *True Positive*

tn : *True Negative*

fp : *False Positive*

fn : *False Negative*

Halaman ini sengaja dikosongkan

BIODATA PENULIS

Penulis lahir di Kediri pada tanggal 19 Mei 2000, dan merupakan anak pertama dari dua bersaudara. Asal usul penulis berasal dari Kabupaten Kediri. Penulis menempuh pendidikan formal mulai dari sekolah dasar di SDN Purwoasri 2, kemudian melanjutkan ke jenjang menengah pertama di MTsN Purwoasri, dan menyelesaikan pendidikan menengah atas di SMAN 2 Pare. Setelah berhasil lulus dari SMAN pada tahun 2019, Penulis mengikuti seleksi masuk perguruan tinggi melalui jalur Vokasi dan diterima di Departemen Statistika Bisnis FV-ITS pada tahun 2020.

Selama menjalani masa studi di ITS, saya aktif terlibat dalam Unit Kegiatan Mahasiswa (UKM) URI. Saya memulai perjalanan organisasional saya sebagai anggota Departemen Media Kreasi dan kemudian naik menjadi Wakil Kepala Departemen. Melalui pengalaman ini, saya terlibat dalam pengembangan diri, baik dari segi *hard skill* maupun *soft skill*.

Minat saya yang mendalam terhadap teknologi *machine learning* dan seni desain menjadi pendorong utama saya untuk terus mengasah kemampuan di kedua bidang tersebut. Saya aktif mengikuti berbagai pelatihan dari Dicoding, dengan fokus khusus pada *data science*. Keterlibatan intensif saya dalam kegiatan-kegiatan ini mencerminkan tekad saya untuk terus belajar dan berkembang di dunia teknologi.

Prestasi yang berhasil saya raih selama perjalanan akademik saya mencerminkan komitmen dan dedikasi saya terhadap pengembangan diri. Untuk membahas lebih lanjut mengenai proyek Tugas Akhir atau topik terkait, jangan ragu untuk menghubungi saya melalui email di ahmad.rizal789.arb@gmail.com. Saya dengan senang hati akan berbagi pengalaman dan pengetahuan, serta bersedia berkontribusi dalam proyek-proyek yang menantang.