TUGAS AKHIR – EF234801

PEMBANGKITAN DESKRIPSI CITRA MAKANAN BERDASARKAN ASPEK ESTETIKA MENGGUNAKAN ENCODER-DECODER CNN-LSTM MODEL

ARIF NUGRAHA SANTOSA

NRP 5025211048

Dosen Pembimbing

Shintami Chusnul Hidayati, S.Kom., M.Sc., Ph.D.

NPP 1987202012004

Program Studi S-1 Teknik Informatika

Departemen Teknik Informatika

Fakultas Teknologi Elektro dan Informatika Cerdas

Institut Teknologi Sepuluh Nopember

Surabaya

2025

(Halaman ini sengaja dikosongkan)

TUGAS AKHIR – EF234801

PEMBANGKITAN DESKRIPSI CITRA MAKANAN BERDASARKAN ASPEK ESTETIKA MENGGUNAKAN ENCODER-DECODER CNN-LSTM MODEL

ARIF NUGRAHA SANTOSA

NRP 5025211048

Dosen Pembimbing

Shintami Chusnul Hidayati, S.Kom., M.Sc., Ph.D.

NPP 1987202012004

Program Studi S-1 Teknik Informatika

Departemen Teknik Informatika

Fakultas Teknologi Elektro dan Informatika Cerdas

Institut Teknologi Sepuluh Nopember

Surabaya

2025

(Halaman ini sengaja dikosongkan)

FINAL PROJECT – EF234801

GENERATION OF FOOD IMAGE DESCRIPTIONS BASED ON AESTHETIC ASPECTS USING ENCODER-DECODER CNN-LSTM MODEL

ARIF NUGRAHA SANTOSA

NRP 5025211048

Dosen Pembimbing

Shintami Chusnul Hidayati, S.Kom., M.Sc., Ph.D.

NPP 1987202012004

Undergradeate Study Program of Informatics

Department of Informatics

Faculty of Intelligent Electrical and Informatics Technology

Institut Teknologi Sepuluh Nopember

Surabaya

2025

(Halaman ini sengaja dikosongkan)

LEMBAR PENGESAHAN

PEMBANGKITAN DESKRIPSI CITRA MAKANAN BERDASARKAN ASPEK ESTETIKA MENGGUNAKAN ENCODER-DECODER CNN-LSTM MODEL

TUGAS AKHIR

Diajukan untuk memenuhi salah satu syarat
memperoleh gelar Sarjana Komputer pada
Program Studi S-1 Teknik Informatika
Departemen Teknik Informatika
Fakultas Teknologi Elektro dan Informatika Cerdas
Institut Teknologi Sepuluh Nopember

Oleh : **ARIF NUGRAHA SANTOSA**
NRP. 5025211048

Disetujui oleh Tim Penguji Tugas Akhir:

1. Shintami Chusnul Hidayati, S.Kom., M.Sc., Ph.D. Pembimbing
2. Ratih Nur Esti Anggraini, S.Kom., M.Sc., Ph.D. Penguji
3. Dr.Eng. Radityo Anggoro, S.Kom., M.Sc. Penguji

SURABAYA
Juli, 2025

(Halaman ini sengaja dikosongkan)

APPROVAL SHEET

GENERATION OF FOOD IMAGE DESCRIPTIONS BASED ON AESTHETIC ASPECTS USING ENCODER-DECODER CNN-LSTM MODEL

FINAL PROJECT

Submitted to fulfill one of the requirements

for obtaining a degree Computer Science at

Undergraduate Study Program of Informatics

Department of Informatics

Faculty of Intelligent Electrical and Informatics Technology

Institut Teknologi Sepuluh Nopember

By: **ARIF NUGRAHA SANTOSA**

NRP. 5025211048

Approved by Final Project Examiner Team:

1. Shintami Chusnul Hidayati, S.Kom., M.Sc., Ph.D. Advisor
2. Ratih Nur Esti Anggraini, S.Kom., M.Sc., Ph.D. Penguji
3. Dr.Eng. Radityo Anggoro, S.Kom., M.Sc. Penguji

SURABAYA

July, 2025

(Halaman ini sengaja dikosongkan)

PERNYATAAN ORISINALITAS

Yang bertanda tangan di bawah ini:

Nama mahasiswa / NRP	:	Arif Nugraha Santosa / 5025211048
Program studi	:	S-1 Teknik Informatika
Dosen Pembimbing / NPP	:	Shintami Chusnul Hidayati, S.Kom., M.Sc., Ph.D. / 1987202012004

dengan ini menyatakan bahwa Tugas Akhir dengan judul "PEMBANGKITAN DESKRIPSI CITRA MAKANAN BERDASARKAN ASPEK ESTETIKA MENGGUNAKAN ENCODER-DECODER CNN-LSTM MODEL" adalah hasil karya sendiri, bersifat orisinal, dan ditulis dengan mengikuti kaidah penulisan ilmiah.

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

Surabaya, 24 Juli 2025

Mengetahui
Dosen Pembimbing

Shintami Chusnul Hidayati, S.Kom., M.Sc., Ph.D.
NIP. 1987202012004

Mahasiswa

Arif Nugraha Santosa
NRP. 5025211048

(Halaman ini sengaja dikosongkan)

STATEMENT OF ORIGINALITY

The undersigned below:

Name of student / NRP	:	Arif Nugraha Santosa / 5025211048
Department	:	Undergraduate Informatics
Advisor / NIP	:	Shintami Chusnul Hidayati, S.Kom., M.Sc., Ph.D. / 1987202012004

Hereby declare that Final Project with the title of "GENERATION OF FOOD IMAGE DESCRIPTIONS BASED ON AESTHETIC ASPECTS USING ENCODER-DECODER CNN-LSTM MODEL" is the result of my own work, is original, and is written by following the rules of scientific writing.

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

Surabaya, 24 July 2025

Acknowledge
Advisor

Shintami Chusnul Hidayati, S.Kom., M.Sc., Ph.D.
NIP. 1987202012004

Student

Arif Nugraha Santosa
NRP. 5025211048

(Halaman ini sengaja dikosongkan)

PERNYATAAN KODE ETIK
PENGGUNAAN AI GENERATIF

Code of Conduct Statement: Generative AI or AI-Assisted Usage

Saya yang bertanda tangan di bawah ini:
I, the undersigned:

Nama Mahasiswa / NRP *Full Name / Student ID*	:	Arif Nugraha Santosa / 5025211048
Program Studi *Study Program*	:	S-1 Teknik Informatika
Judul Tugas Akhir *Final Project Title*	:	"Pembangkitan Deskripsi Citra Makanan Berdasarkan Aspek Estetika Menggunakan Encoder-Decoder CNN-LSTM Model"

dengan ini menyatakan bahwa pada Tugas Akhir dengan judul di atas tersebut:
hereby declare that in the Final Project with the above title:

No.	Pernyataan *Statement*	(☑)
1	Saya hanya menggunakan AI generatif sebagai alat bantu untuk memperbaiki tata bahasa. AI generatif tidak digunakan untuk membuat isi Tugas Akhir. *I only used generative AI as a tool to improve the readability or language of the text in my Final Project. It was not used to generate a complete text of my work.*	☑
2	Saya telah memeriksa dan/atau memperbaiki seluruh bagian dari Tugas Akhir saya yang dibantu oleh AI generatif agar sesuai dengan baku mutu penulisan karya ilmiah. *I have reviewed and refined all aspects of my work that generative AI assists with, ensuring it adheres to the standards of academic writing.*	☑
3	Saya tidak menggunakan AI generatif untuk pembuatan data primer, grafik dan/atau tabel pada Tugas Akhir saya. *I did not use generative AI to generate primary data, figures, and/or tables in my work.*	☑
4	Saya telah memberikan atribusi/pengakuan terhadap alat AI yang digunakan, secara rinci pada suatu bagian pada lampiran. *I have acknowledged the use of generative AI in any part of the work in the specific appendix page.*	☑
5	Saya memastikan tidak ada plagiarisme, termasuk hal yang berasal dari penggunaan AI generatif. *I have ensured that there is no plagiarism issue in the work, including any parts generated by AI.*	☑

Surabaya, 25 Juli 2025

Mahasiswa

Arif Nugraha Santosa
NRP. 5025211048

(Halaman ini sengaja dikosongkan)

ABSTRAK

PEMBANGKITAN DESKRIPSI CITRA MAKANAN BERDASARKAN ASPEK ESTETIKA MENGGUNAKAN ENCODER-DECODER CNN-LSTM MODEL

Nama Mahasiswa / NRP	**:**	**Arif Nugraha Santosa / 5025211048**
Departemen	**:**	**Teknik Informatika FTEIC - ITS**
Dosen Pembimbing	**:**	**Shintami Chusnul Hidayati, S.Kom., M.Sc., Ph.D.**

Abstrak

Penelitian ini mengembangkan sebuah sistem *Image Aesthetic Captioning* untuk menghasilkan deskripsi estetika citra makanan. Sistem memiliki dua tahap utama yaitu *Single-Aspect Captioning* berbasis arsitektur CNN-LSTM yang menghasilkan *caption* untuk enam aspek estetika visual, serta *Unsupervised Text Summarization* menggunakan model *Denoising Autoencoder* untuk merangkum berbagai *caption* tersebut menjadi satu narasi utuh. Model *pretrain* CNN-LSTM dilakukan pelatihan menggunakan *dataset* MS COCO 2017 yang telah difilter dengan domain makanan. *Dataset fine-tune* diperoleh dari platform komunitas fotografi dan diproses melalui tahapan pembersihan teks, pengukuran *informativeness* menggunakan *informativeness score* yang terinspirasi dari metode TF-IDF, serta *topic mining* menggunakan LDA yang selanjutnya dilakukan *aspect mapping* secara manual terhadap *caption* mentahnya. Hasil evaluasi menunjukkan bahwa model *Single-Aspect Captioning* mencapai performa tertinggi pada aspek *general impression* dengan skor BLEU-1 sebesar 0.2768, aspek *composition* dengan BLEU-1 sebesar 0.1993 dan METEOR 0.0655, aspek *color & light* dengan BLEU-1 sebesar 0.1793 dan METEOR 0.0639. Model DAE mampu menyusun narasi akhir dengan skor BLEU-1 sebesar 0.4167 dan ROUGE-L sebesar 0.2244. Untuk meningkatkan koherensi, dilakukan proses *retouch* menggunakan LLM berbasis instruksi dengan *prompt* terstruktur. Hasil *retouch* menunjukkan peningkatan pada mayoritas metrik termasuk BLEU-1 menjadi 0.4323, ROUGE-L menjadi 0.2431, dan METEOR menjadi 0.1314. Meskipun skor CIDEr sedikit menurun, pendekatan *retouch* ini terbukti efektif dalam menghasilkan *caption* estetika yang informatif, terstruktur, dan representatif terhadap seluruh aspek visual.

Kata kunci: ***Image Aesthetic Captioning,*** **CNN-LSTM,** ***Denoising Autoencoder, Large Language Model, Citra Makanan, Deskripsi Estetika.***

(Halaman ini sengaja dikosongkan)

ABSTRACT

GENERATION OF FOOD IMAGE DESCRIPTIONS BASED ON AESTHETIC ASPECTS USING ENCODER-DECODER CNN-LSTM MODEL

Student Name / NRP	**:**	**Arif Nugraha Santosa / 5025211048**
Department	**:**	**Teknik Informatika FTEIC - ITS**
Advisor	**:**	**Shintami Chusnul Hidayati, S.Kom., M.Sc., Ph.D.**

Abstract

This study develops an Image Aesthetic Captioning system to generate aesthetic descriptions of food images. The system consists of two main stages Single-Aspect Captioning based on CNN-LSTM architecture, which generates captions for six visual aesthetic aspects, and Unsupervised Text Summarization using a Denoising Autoencoder (DAE) model to merge these captions into a single coherent narrative. The CNN-LSTM model was pretrained using the MS COCO 2017 dataset filtered for the food domain. The fine-tuning dataset was collected from a photography community platform and processed through several steps, including text cleaning, informativeness scoring inspired by TF-IDF, topic mining using LDA, and manual aspect mapping for raw captions. Evaluation results show that the Single-Aspect Captioning model achieved its highest performance on the *general impression* aspect with a BLEU-1 score of 0.2768, along with strong performance on the *composition* aspect (BLEU-1: 0.1993, METEOR: 0.0655) and the *color & light* aspect (BLEU-1: 0.1793, METEOR: 0.0639). The DAE model successfully composed a final narrative with a BLEU-1 score of 0.4167 and a ROUGE-L score of 0.2244. To improve coherence and readability, a retouch process was conducted using an instruction-tuned Large Language Model (LLM) with structured prompts. The retouched captions showed improvements across most metrics, including BLEU-1 (0.4323), ROUGE-L (0.2431), and METEOR (0.1314). Although the CIDEr score slightly decreased, the retouch approach proved effective in producing aesthetic captions that are informative, well-structured, and representative of all visual aspects.

Keywords: ***Image Aesthetic Captioning,*** **CNN-LSTM,** ***Denoising Autoencoder, Large Language Model, Food Image, Aesthetic Captioning.***

(Halaman ini sengaja dikosongkan)

KATA PENGANTAR

Puji syukur kehadirat Allah SWT atas rahmat dan karunia-Nya, penulis dapat menyelesaikan Tugas Akhir ini dengan judul "*PEMBANGKITAN DESKRIPSI CITRA MAKANAN BERDASARKAN ASPEK ESTETIKA MENGGUNAKAN ENCODER-DECODER CNN-LSTM MODEL*" sebagai salah satu syarat untuk memperoleh gelar Sarjana Komputer di Institut Teknologi Sepuluh Nopember.

Penulisan Tugas Akhir ini tidak akan terlaksana dengan baik tanpa bimbingan, dukungan, dan motivasi dari berbagai pihak. Oleh karena itu, penulis ingin menyampaikan ucapan terima kasih yang sebesar-besarnya kepada:

1. Allah SWT beserta Nabi Muhammad SAW., karena rahmat dan karunia-Nya penulis dapat menyelesaikan Tugas Akhir dan juga perkuliahan di Teknik Informatika ITS,
2. Keluarga kandung penulis yaitu Ibu Verinita, Bapak Decky, Iyon, dan Alice yang selalu memberikan semangat dan dukungan dalam penulisan Tugas Akhir ini,
3. Serly Nur Isnaini yang selalu menemani, mendukung, dan membantu selama proses eksperimen, penulisan Tugas Akhir ini,
4. Keluarga kedua penulis yaitu Ibu Siti Rosidah, Bapak Solikh, Mas Cois yang merawat, membantu, dan memberikan fasilitas untuk mengerjakan Tugas Akhir ini,
5. Departemen Teknik Informatika, Fakultas Teknologi Elektro dan Informatika Cerdas, dan Institut Teknologi Sepuluh Nopember yang telah menerima dan mendidik penulis,
6. Ibu Shintami Chusnul Hidayati, S.Kom, M.Sc., Ph.D. yang telah membimbing penelitian dan penyusunan buku Tugas Akhir ini serta memberikan dukungan kepada penulis,
7. Seluruh dosen dan karyawan serta ibu kantin Departemen Teknik Informatika ITS yang telah memberikan ilmu dan pengalaman serta semangat kepada penulis selama menjalani masa kuliah di Teknik Informatika ITS,
8. Rayhan, Vasy, Danno, Thalent, Ryan, dan teman-teman kuliah lainnya yang tidak dapat disebutkan satu-satu yang selalu membantu selama perkuliahan melalui dukungan yang diberikan kepada penulis,
9. Seluruh teman-teman dan keluarga “Pakan House” yang memberikan semangat,
10. TBRK ROASTERY yang selalu menjadi tempat penulis menulis buku dari proposal hingga Tugas Akhir,
11. Serta semua pihak yang telah turut membantu penulis dalam perkuliahan dan penyelesaian Tugas Akhir ini.

Penulis menyadari bahwa Tugas Akhir ini masih jauh dari sempurna. Oleh karena itu, penulis mengharapkan kritik dan saran yang membangun untuk perbaikan di masa mendatang. Semoga hasil penelitian ini dapat bermanfaat bagi perkembangan ilmu pengetahuan dan teknologi.

(Halaman ini sengaja dikosongkan)

DAFTAR ISI

(Halaman ini sengaja dikosongkan)

DAFTAR GAMBAR

(Halaman ini sengaja dikosongkan)

DAFTAR TABEL

(Halaman ini sengaja dikosongkan)

DAFTAR KODE SEMU

(Halaman ini sengaja dikosongkan)

BAB 1
PENDAHULUAN

1.1 Latar Belakang

Citra visual makanan memiliki pengaruh signifikan terhadap persepsi dan perilaku manusia karena berbagai mekanisme kognitif dan neurologis. Otak manusia secara otomatis mengevaluasi kepadatan energi dari makanan yang ditampilkan dalam sebuah citra/gambar, sehingga perhatian visual cenderung diarahkan pada makanan yang kaya energi. Respons ini berakar pada mekanisme evolusi yang mendukung kelangsungan hidup, karena makanan dengan energi tinggi dianggap lebih bernutrisi. Selain itu, melihat citra makanan sering kali memicu simulasi mental dari pengalaman makan, seperti membayangkan rasa, aroma, dan tekstur makanan. Faktor estetika, seperti simetri gambar, variasi warna, kilau pencahayaan, dan penyajian yang dinamis, juga memainkan peran penting dalam meningkatkan daya tarik visual makanan. Sebagai contoh, warna-warna cerah pada *salad* sering diasosiasikan dengan kesegaran dan kesehatan. Hal ini menunjukkan bagaimana elemen visual dalam fotografi dan pemasaran makanan dapat memengaruhi psikologi manusia, menjadikan makanan bukan hanya sebagai sumber nutrisi, tetapi juga pengalaman visual dan emosional yang menarik (Spence dkk., 2022).

Penilaian estetika pada citra/gambar bertujuan untuk membedakan kualitas foto secara komputasional berdasarkan aturan fotografi yang biasanya berbentuk klasifikasi biner atau penilaian skor. *Image Aesthetic Captioning* (IAC), adalah tugas *multi-modal* yang umumnya menggunakan jaringan *end-to-end* khusus. Proses tersebut melibatkan pelatihan menggunakan citra/gambar dan ulasan estetika yang sesuai, lalu menghasilkan *caption*/deskripsi tekstual dari perspektif estetika, seperti warna, pencahayaan, komposisi, dan sebagainya dari citra tersebut (Jin dkk., 2022). Dalam konteks yang lebih terperinci, IAC juga dapat diterapkan dalam penilaian estetika citra makanan untuk mendukung bisnis dalam menghasilkan citra makanan yang lebih baik untuk keperluan periklanan dan lain sebagainya, sehingga dapat menarik lebih banyak konsumen (Zou dkk., 2020).

Jurnal penelitian (Zou dkk., 2020) memperkenalkan model *Food* IAC yang menggabungkan dua modul utama, yaitu modul *Single-Aspect Captioning* (SAC) yang merupakan bagian dari arsitektur *deep learning* menggunakan *Convolutional Neural Network* (CNN) sebagai *encoder* dan *Long Short-Term Memory* (LSTM) sebagai *decoder* untuk menghasilkan deskripsi setiap atribut estetika, dan modul *Unsupervised Text Summarization* (UTS) yang merupakan model *deep learning* menggunakan *Denoising Auto-Encoder* (DAE) untuk merangkum dan menggabungkan deskripsi dari modul SAC menjadi satu teks yang lebih koheren.

Saat ini, kualitas *dataset* sangat penting untuk kinerja model. Namun, menambahkan *caption* yang sesuai dengan aturan estetika ke *dataset* berskala besar akan memakan biaya dan waktu yang besar (Jin dkk., 2022). Oleh karena itu, digunakan metode *Term Frequency-Inverse Document Frequency* (TF-IDF), serta model *mining* topik *Latent Dirichlet Allocation* (LDA) untuk memproses citra dan *caption*/deskripsi dari situs/*platform* fotografi agar kualitasnya mendekati *dataset* yang diberi label secara manual (Jin dkk., 2022). Penelitian (Zou dkk., 2020) memperkenalkan *dataset* baru yang terdiri dari sekitar 7.100 citra makanan dan 46.000 komentar. *Dataset* tersebut didapatkan dari *platform* dpchallenge.com, di mana setiap gambar memiliki deskripsi estetika sebanyak 4 atribut dengan proses pembersihan data menggunakan metode yang terinpirasi dari TF-IDF yaitu *Informativeness Score*, dan model *mining* topik LDA.

Dataset tersebut akan diolah menggunakan model *Convolutional Neural Network-Long Short-Term Memory* (CNN-LSTM) untuk menghasilkan deskripsi dengan fokus estetika.

Model *Food* IAC pada penelitian (Zou dkk., 2020) memiliki celah dalam menghasilkan deskripsi dalam modul SAC, yaitu dataset yang digunakan hanya mencakup empat aspek estetika berupa, *general impression*, *composition*, *color & light*, serta *DOF and focus*. Walaupun keempat aspek ini memberikan gambaran dasar mengenai estetika citra makanan, penelitian ini belum mencakup elemen penting lain seperti subjek utama dalam gambar (*subject*) dan teknik penggunaan kamera (*camera use*), yang sebenarnya berperan besar dalam menghasilkan deskripsi estetika yang lebih mendalam dan terarah. Penambahan kedua aspek ini terinspirasi dari analisis penelitian (Jin dkk., 2022) yang menekankan pentingnya atribut visual tambahan dalam memberikan deskripsi estetika yang lebih kaya dan relevan. Dalam penelitian tersebut, disebutkan bahwa aspek seperti subjek visual dan elemen teknis kamera memiliki pengaruh signifikan terhadap evaluasi estetika gambar. Aspek subjek membantu menonjolkan elemen utama dalam gambar, sedangkan penggunaan kamera, seperti *shutter speed* dan *depth of field*, berkontribusi pada kualitas visual keseluruhan melalui teknik pengambilan gambar yang kreatif.

Penelitian Tugas Akhir ini berencana mengembangkan penelitian (Zou dkk., 2020) dengan menambahkan aspek subjek makanan dan penggunaan kamera pada modul SAC, sehingga menghasilkan kalimat yang lebih rinci dalam mendeskripsikan subjek makanan, dan penggunaan kamera pada citra makanan yang diberikan. Pengembangan selanjutnya adalah mencoba menggabungkan hasil inferensi modul SAC menggunakan *Large Language Model* (LLM) untuk menjadi pembanding pada modul UTS yang menggunakan model DAE. Hasil dari penelitian ini diharapkan dapat menjadi referensi terhadap penelitian yang akan datang, serta menjadi acuan untuk menggunakan *dataset* citra makanan dan model CNN-LSTM dalam pembangkitan deskripsi estetika pada citra makanan.

1.2 Rumusan Masalah

Rumusan masalah yang diangkat untuk Tugas Akhir ini adalah sebagai berikut:

1. Bagaimana cara menghasilkan deskripsi estetika citra makanan yang relevan dan terperinci untuk setiap aspek visual menggunakan model *Single-Aspect Captioning* (SAC) berbasis CNN-LSTM?
2. Bagaimana pengaruh penambahan aspek *subject* dan *use of camera* dalam modul SAC terhadap peningkatan kualitas deskripsi estetika citra makanan secara keseluruhan?
3. Sejauh mana efektivitas model *Denoising Autoencoder* (DAE) dalam menggabungkan enam caption aspek menjadi satu deskripsi naratif, dan bagaimana perbandingannya dengan *retouch caption* menggunakan *Large Language Model* (LLM)?

1.3 Batasan Masalah

Batasan masalah yang ditemukan dari Tugas Akhir ini adalah sebagai berikut:

1. Penelitian ini menggunakan *dataset* citra makanan dari situs dpchallenge.com yang terdiri dari kurang lebih 7.200 citra dan 46.000 komentar berbahasa Inggris, tanpa melakukan perluasan atau augmentasi eksternal.
2. Fokus *captioning* terbatas pada enam aspek estetika, yaitu *general impression, subject, use of camera, color & light, composition,* dan *depth of field & focus*.

3. Proses *mapping* komentar estetika dilakukan dengan pendekatan yang terinspirasi TF-IDF (*Informativeness Score*) dan *Latent Dirichlet Allocation* (LDA). Teknik *prompt engineering* atau manual *tagging* tidak digunakan karena keterbatasan waktu dan sumber daya komputasi.
4. Model *captioning* yang digunakan untuk menghasilkan deskripsi per aspek dibangun menggunakan arsitektur CNN-LSTM yang dilakukan *pretraining* terhadap *dataset* luar dengan domain yang sama, dan di-*fine-tune* secara terpisah untuk masing-masing aspek.
5. Penggabungan enam *caption* per aspek menjadi satu kalimat naratif dilakukan menggunakan pendekatan *Unsupervised Text Summarization* (UTS) berbasis *Denoising Autoencoder* (DAE), tanpa melibatkan mekanisme *fine-tuning* berbasis *attention* atau *pretraining* berbasis *transformer*.
6. *Caption* hasil DAE dapat mengalami *postprocessing* tambahan dengan *Large Language Model* (LLM) untuk perbandingan, namun LLM tidak dilatih secara *end-to-end* demi menjaga fokus evaluasi terhadap model DAE.
7. Evaluasi model dilakukan menggunakan metrik *captioning* standar seperti BLEU, ROUGE-L, METEOR, dan CIDEr. Metrik SPICE tidak disertakan dalam seluruh eksperimen karena keterbatasan dependensi eksternal.

1.4 Tujuan

Tujuan dari Tugas Akhir ini adalah sebagai berikut:

1. Mengembangkan model CNN-LSTM untuk captioning estetika per aspek (*Single-Aspect Captioning*) yang mampu menghasilkan deskripsi relevan dan terstruktur terhadap enam dimensi estetika, yaitu *general impression, subject, use of camera, color and light, composition,* serta *depth of field and focus*.
2. Mengevaluasi pengaruh penambahan dua aspek penting, yaitu *subject* dan *use of camera*, terhadap kualitas deskripsi estetika yang dihasilkan oleh model CNN-LSTM. Penambahan ini bertujuan memperluas cakupan informasi visual dan meningkatkan relevansi interpretasi estetika terhadap konteks citra makanan.
3. Membangun modul penggabungan deskripsi estetika dengan pendekatan *unsupervised*, menggunakan *Denoising Autoencoder* (DAE) sebagai metode *Unsupervised Text Summarization* (UTS) yang dirancang untuk menghasilkan satu kalimat naratif estetika yang ringkas dan koheren dari enam *caption* per aspek.

1.5 Manfaat

Penelitian ini diharapkan memberikan manfaat baik secara teoritis maupun praktis. Secara teoritis, penelitian ini dapat memperkaya literatur di bidang *Image Aesthetic Captioning* (IAC) dengan mengintegrasikan elemen subjek makanan dan penggunaan kamera pada modul *Single-Aspect Captioning* (SAC), melanjutkan pengembangan dari penelitian sebelumnya oleh (Zou dkk., 2020). Hasilnya diharapkan menjadi referensi bagi penelitian lanjutan untuk pengembangan model IAC yang lebih kompleks dan aplikatif di masa depan, khususnya dalam konteks citra makanan.

Selain itu, penggunaan metode TF-IDF dan *Latent Dirichlet Allocation* (LDA) dalam pembersihan data menawarkan panduan praktis untuk pengolahan *dataset* berskala besar secara efisien, sehingga dapat diterapkan dalam berbagai penelitian serupa yang memerlukan manajemen data skala besar. Penelitian ini juga memiliki potensi untuk mendukung inovasi di bidang kecerdasan buatan, khususnya dalam meningkatkan pengalaman visual konsumen.

Implementasi dan pengembangan model ini tidak hanya relevan untuk tema makanan, namun juga untuk tema visual lainnya, seperti pariwisata, mode, dan seni visual. Dengan demikian, penelitian ini diharapkan dapat memberikan kontribusi yang luas dan aplikatif dalam pengembangan teknologi berbasis AI.

BAB 2
TINJAUAN PUSTAKA

2.1 Hasil Penelitian Terdahulu

Dalam pengerjaan Tugas Akhir ini, terdapat informasi dari penelitian sebelumnya yang digunakan sebagai pendukung, pembanding, dan referensi saat melakukan pengujian. Informasi tersebut tersaji di dalam Tabel 2.1 yang didapat dari jurnal, buku, dan skripsi yang relevan dengan judul Tugas Akhir ini sebagai acuan dasar teoritis yang kuat.

Tabel 2.1 Fokus Penelitian Terdahulu Terhadap Penelitian Terkait

Judul Penelitian Terdahulu	Fokus Penelitian Terdahulu	Perbedaan dengan Tugas Akhir
Generating Aesthetic Based Critique For Photographs (Yeo dkk., 2021)	Membahas metode untuk menghasilkan kritik berbasis estetika dari citra menggunakan model *encoder-decoder* berbasis CNN-LSTM. Model ini bertujuan untuk memberikan umpan balik yang lebih detail terkait aspek teknis seperti pencahayaan, komposisi, dan fokus pada sebuah citra, serta menghasilkan kritik tekstual untuk meningkatkan kualitas fotografi secara umum.	Menggunakan kombinasi fitur estetika yang dihasilkan dari *aesthetic scorer* dan *factual encoder* untuk menghasilkan *caption* kritis yang mirip manusia. Mengadopsi *Word Mover's Distance* untuk mengevaluasi kesamaan semantik antara *caption* yang dihasilkan dengan *ground-truth*.
Stimulus-driven and Concept-driven Analysis for Image Caption Generation (Ding dkk., 2020)	Mengintegrasikan mekanisme perhatian *stimulus-driven* dan *concept-driven* untuk menghasilkan *caption* yang lebih akurat. Model ini menggunakan kombinasi CNN untuk ekstraksi fitur pada sebuah citra dan LSTM untuk pembangkitan teks, serta perhatian adaptif untuk memilih area citra yang relevan. Penelitian ini juga mengeksplorasi bagaimana perhatian visual manusia dapat diimitasi dalam proses *captioning*.	Mengandalkan dua mekanisme atensi, yang menambah kompleksitas komputasi. Tidak unggul dalam beberapa metrik evaluasi seperti BLEU.
To Be an Artist: Automatic Generation on Food Image Aesthetic Captioning (Zou dkk., 2020)	Mengembangkan model multi-aspek untuk menghasilkan *caption* estetika citra makanan. Model ini menggunakan	Kurangnya keberagaman dalam *caption* yang dihasilkan, ini disebabkan karena *dataset* yang digunakan hanya mencakup

Judul Penelitian Terdahulu	Fokus Penelitian Terdahulu	Perbedaan dengan Tugas Akhir
	CNN untuk ekstraksi fitur visual dan LSTM untuk modul S*ingle-Aspect Captioning*, serta modul *Unsupervised Text Summarization* untuk menghasilkan deskripsi estetika yang koheren dari berbagai aspek seperti warna, pencahayaan, dan komposisi.	sekitar 7000 gambar dan 20000 teks. Serta aspek estetika yang digunakan hanyalah kesan umum, warna dan pencahayaan, komposisi, serta kedalaman dan fokus pada sebuah citra. Penelitian ini juga menggunakan DAE untuk menggabungkan *caption* yang dihasilkan oleh model CNN-LSTM.
VILA: Learning Image Aesthetics from User Comments with Vision-Language Pretraining (Ke dkk., 2023)	Menggunakan pasangan citra dan komentar dari situs fotografi untuk pelatihan model *vision-language*. Model ini berfokus pada mempelajari estetika citra secara *multimodal* tanpa label manual, dengan *pretraining* pada pasangan citra-teks menggunakan strategi kontrastif dan generatif, serta pengadaptasian untuk tugas penilaian estetika berbasis peringkat.	Model ini menggunakan *pretraining vision-language* pada *image-comment pairs*, memungkinkan untuk mempelajari konsep estetika yang lebih kaya tanpa label manusia, dan menghasilkan performa yang superior dalam tugas *zero-shot*. Metode ini bergantung pada *pretraining* skala besar yang membutuhkan banyak data dan daya komputasi tinggi, serta hasil mungkin kurang relevan untuk data dengan jumlah yang lebih kecil atau spesifik.
Aesthetic Attributes Assessment of Images (Jin dkk., 2019)	Mengembangkan jaringan multi-atribut untuk menilai atribut estetika gambar seperti warna, pencahayaan, komposisi, dan fokus. Model ini menghasilkan *caption* dan skor estetika untuk setiap atribut menggunakan dataset beranotasi kecil *Photo Critique Captioning Dataset* (PCCD) dan *dataset* besar yang dianotasi lemah (DPC-Captions).	Memperkenalkan model *Aesthetic Multi-Attribute Network* (AMAN) untuk menghasilkan *caption* dan skor atribut estetik gambar. Menggunakan *pretraining* dan *transfer learning* dari *dataset* kecil ke *dataset* besar dengan hasil yang lebih baik daripada model CNN-LSTM. Model ini membutuhkan sumber daya komputasi yang besar karena proses transfer learning dan multi-task learning, serta memiliki keterbatasan dalam performa atribut yang jarang terkomentari.

Judul Penelitian Terdahulu	Fokus Penelitian Terdahulu	Perbedaan dengan Tugas Akhir
Aesthetic Image Captioning on the FAE-Captions Dataset (Jin dkk., 2022)	Membahas pengembangan *dataset* besar bernama FAE-Captions untuk *captioning* estetika. *Dataset* ini dibersihkan menggunakan LDA dan *active learning*. Model CNN-LSTM digunakan untuk menghasilkan deskripsi estetika pada citra dengan fungsi kerugian estetika khusus untuk meningkatkan relevansi dan akurasi *caption*.	Membangun *dataset caption* gambar estetika terbesar (FAE-Captions) dengan lebih dari 1,2 juta gambar dan komentar dari situs fotografi profesional. Menggunakan model CNN-LSTM untuk menghasilkan *caption* dengan fokus estetika. Dataset ini terbatas pada data yang berasal dari komentar pengguna di situs fotografi, yang mungkin mengandung banyak noise dan informasi yang tidak relevan.
Aesthetic Image Captioning From Weakly-Labelled Photographs (Ghosal dkk., 2019)	Mengembangkan strategi pembersihan data berbasis probabilistik untuk menciptakan *dataset* besar dan bersih bernama AVA-Captions. Model CNN-LSTM dilatih menggunakan data anotasi lemah untuk menghasilkan *caption* estetika dengan mempertimbangkan atribut seperti komposisi, warna, dan pencahayaan. Model ini bertujuan mengatasi kendala ketersediaan *dataset* besar yang berkualitas untuk tugas *captioning* estetika.	Pendekatan baru untuk *cleaning* data yang memiliki banyak *noise* dan membuat *dataset Aesthetic Visual Analysis* (AVA) yang bersih. Menggunakan *weakly supervised learning* CNN untuk menangkap atribut estetikanya. *Dataset* yang diberikan masih memiliki banyak *noise* meskipun sudah dilakukan *cleaning*. CNN *training* dengan menggunakan *weakly supervised training* juga masih belum optimal dibandingkan dengan *fully supervised training*.

2.2 Dasar Teori

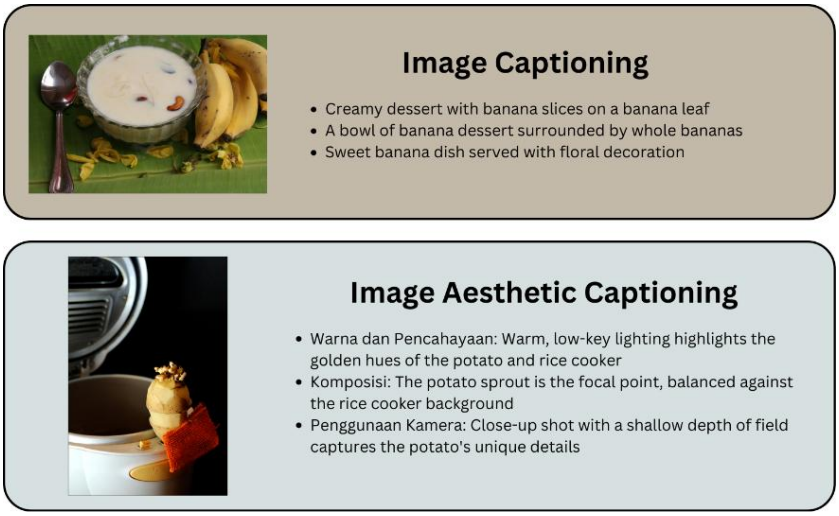
Subbab ini menguraikan tentang landasan teori yang digunakan dalam pembuatan Tugas Akhir ini.

2.2.1 *Image Aesthetic Captioning* (IAC)

Image Captioning (pembangkit deskripsi citra) adalah proses otomatis untuk menghasilkan deskripsi/*caption* teks yang bermakna dari konten visual pada sebuah citra. Teknik ini melibatkan pemahaman visual tentang elemen-elemen di dalam citra, termasuk objek, hubungan antar objek, dan konteks visualnya. Teknologi ini banyak digunakan dalam berbagai aplikasi seperti *visual question answering*, analisis *video* pengawasan, *video captioning*, dan sebagainya (Sasibhooshan dkk., 2023).

Image Aesthetic Captioning (pembangkit deskripsi citra berdasarkan atribut estetika) atau *Aesthetic Image Captioning* adalah varian dari *image captioning* yang secara khusus bertujuan

untuk menghasilkan *caption*/deskripsi teks berdasarkan atribut estetika citra seperti komposisi, warna, pencahayaan, dan kesan keseluruhan. Deskripsi ini tidak hanya mencerminkan fakta tentang sebuah citra, tetapi juga menyampaikan nilai-nilai estetika dari perspektif kritik visual atau seni fotografi. Contoh penerapan IAC meliputi pengembangan kamera pintar atau aplikasi berbasis situs untuk peringkat dan pengambilan gambar (Ghosal dkk., 2019). Perbedaan pembangkit deskripsi gambar biasa dengan pembangkit deskripsi gambar berdasarkan atribut estetikanya dapat dilihat pada Gambar 2.1.



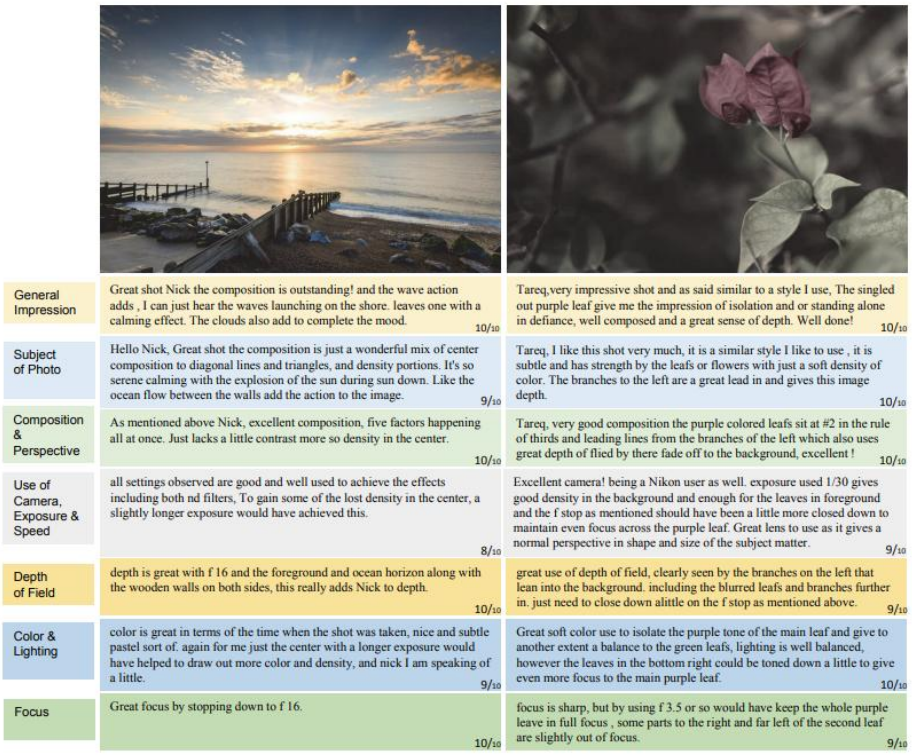
Gambar 2.1 Perbedaan *Image Captioning* dan *Image Aesthetic Captioning*

Dapat dilihat pada Gambar 2.1, *Image Captioning* fokus pada menggambarkan konten sebuah citra secara faktual, seperti identifikasi objek dan hubungan antar objek, dengan tujuan menciptakan deskripsi yang informatif. Sebaliknya, *Image Aesthetic Captioning* berfokus pada aspek estetika citra, seperti warna, pencahayaan, komposisi, dan penggunaan kamera yang relevan untuk aplikasi kreatif dan artistik.

2.2.2 Aspek Estetika

Penelitian (Zou dkk., 2020), aspek estetika didefinisikan sebagai dimensi-dimensi kunci yang secara kolektif membentuk penilaian estetika terhadap sebuah citra. Penelitian tersebut mengidentifikasi empat aspek utama, yaitu; *general impression*, yang merujuk pada persepsi keseluruhan terhadap keindahan gambar; *composition*, yaitu keteraturan elemen visual dalam bingkai gambar; *depth of field and focus*, yang menilai seberapa tajam dan jelas objek utama serta kedalaman visual gambar; serta *color and lighting*, yang berfokus pada kualitas pencahayaan dan penggunaan warna dalam gambar. Keempat aspek ini diambil berdasarkan komentar-komentar pengguna dalam situs dpchallenge.com, yang dianalisis dan diklasifikasikan menggunakan pendekatan berbasis keyword dan jaringan neural multi-atribut (AMAN) (Jin dkk., 2019).

Aspek subjek gambar (*subject of photo*) dan penggunaan kamera (*use of camera*) secara eksplisit disebutkan sebagai bagian dari atribut estetika dalam penelitian (Chang dkk., t.t.) yang mengusulkan *dataset* baru di bidang estetika foto bernama *Photo Caption Critique Dataset* (PCCD). Pada penelitian (Jin dkk., 2019) dua atribut estetika tersebut masing-masing diuraikan melalui kata kunci estetika yang sering muncul dalam komentar. Misalnya, untuk subjek gambar terdapat kata seperti "*subject*", "*interesting*", dan "*capture"*, sedangkan penggunaan kamera dicirikan dengan kata seperti "*exposure*", "*shutter*", dan "ISO".



Gambar 2.2 Sampel dari *Photo Critique Captioning Dataset* (Chang dkk., t.t.)

Gambar 2.2 menampilkan *caption* dan citra dari *dataset* PCCD. Dapat dilihat bahwa setiap *caption* dikategorikan ke dalam suatu aspek estetika yang relevan dengan *caption* tersebut. Sebagai contoh, kata kunci yang muncul pada aspek *use of camera* meliputi aspek teknis seperti "*nd filter*", "*exposure*", "Nikon", dan "*f stop*". Aspek teknis ini menjadi sangat penting terutama ketika penilaian dilakukan oleh fotografer atau penikmat seni visual yang berorientasi pada kualitas dan pengaturan hasil tangkapan kamera (Chang dkk., t.t.).

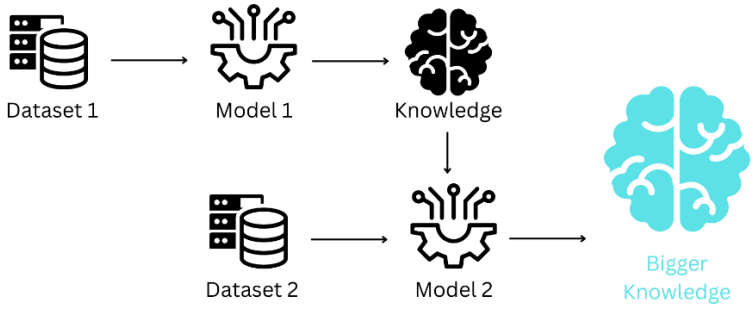
2.2.3 *Transfer Learning*

Transfer Learning adalah teknik dalam pembelajaran mesin di mana model yang telah dilatih pada suatu tugas (*source task*) digunakan kembali untuk membantu proses pelatihan pada tugas lain yang berbeda namun terkait (*target task*). Teknik ini memanfaatkan representasi pengetahuan berupa bobot model, fitur laten, maupun struktur arsitektur yang diperoleh dari *dataset* berskala besar pada domain awal, untuk meningkatkan efisiensi dan performa pelatihan pada domain baru yang memiliki data terbatas. *Transfer Learning* sangat berguna dalam situasi di mana pengumpulan dan pelabelan data baru memerlukan biaya atau waktu yang besar.

Pendekatan ini umumnya melibatkan tiga strategi utama yaitu; *feature extraction*, di mana bagian awal model pra-latih digunakan untuk mengekstrak fitur dari data baru, kemudian *fine-tuning*, di mana sebagian atau seluruh parameter model pra-latih disesuaikan kembali pada *dataset* target, dan *frozen layers* di mana hanya bagian akhir dari model yang dilatih ulang sementara lapisan awal tetap tidak berubah.

Transfer Learning telah berhasil diterapkan dalam berbagai bidang seperti klasifikasi gambar, deteksi objek, pemrosesan bahasa alami (NLP), dan pengenalan pola, karena memungkinkan penggunaan model pra-latih seperti ResNet, BERT, atau GPT pada berbagai tugas baru.

Dalam konteks estetika gambar, *Transfer Learning* memegang peranan penting karena model mampu menggeneralisasi pemahaman tentang karakteristik visual kompleks—seperti komposisi, pencahayaan, dan warna yang telah dipelajari dari *dataset* berskala besar seperti ImageNet. Pengetahuan ini kemudian diadaptasi untuk mengenali dan memahami estetika dalam *dataset* baru yang lebih kecil dan spesifik. Dengan demikian, proses pelatihan menjadi lebih cepat, akurat, dan hemat sumber daya, tanpa harus membangun model dari awal (Hosna dkk., 2022). Ilustrasi *Transfer Learning* dapat dilihat pada Gambar *2.3*.



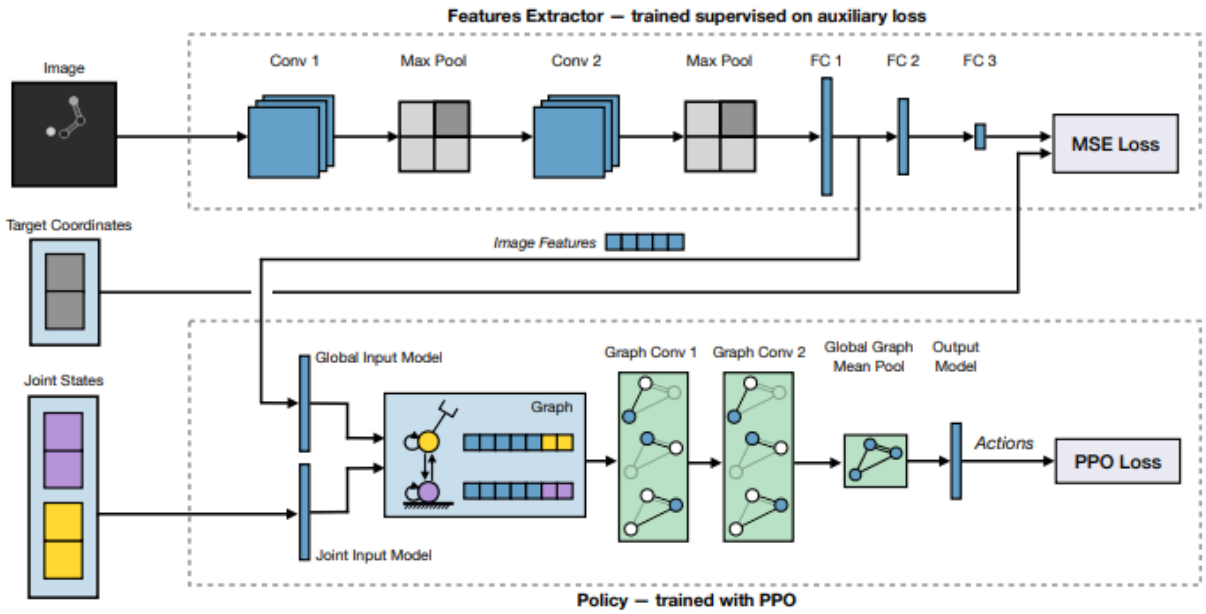
Gambar 2.3 Ilustrasi *Transfer Learning*

2.2.4 *Image Encoder*

Image *encoder* adalah komponen awal dalam arsitektur *image captioning* yang bertugas mengekstraksi fitur visual dari gambar dan mengubahnya menjadi representasi vektor berdimensi tetap. Representasi ini menyimpan informasi semantik penting yang dapat dimanfaatkan oleh komponen lanjutan seperti *decoder* berbasis LSTM untuk menghasilkan deskripsi teks. *Encoder* umumnya menggunakan model *deep Convolutional Neural Network* (CNN) yang telah dilatih sebelumnya untuk mengekstraksi fitur spasial maupun global dari gambar (Oliva dkk., 2022). Proses ini memungkinkan sistem memahami struktur visual seperti bentuk, objek, dan komposisi tanpa perlu melakukan klasifikasi eksplisit.

Dalam praktiknya, *image encoder* tidak hanya menghasilkan satu vektor tunggal tetapi seringkali menghasilkan dua jenis fitur: *spatial features* dan *global features*. *Spatial features* mempertahankan struktur spasial dari gambar dan berguna untuk proses atensi (*attention*), sedangkan *global features* merangkum keseluruhan isi gambar dalam satu vektor padat. Misalnya, dalam *encoder* berbasis ResNet, spatial feature dapat diperoleh dari output konvolusi akhir, sementara *global feature* dapat diambil dari hasil *pooling* rata-rata dan transformasi linier selanjutnya. Kedua fitur ini kemudian digabungkan sebagai input bagi *decoder* untuk meningkatkan kualitas dan konteks *caption* yang dihasilkan.

Gambar 2.4 memperlihatkan arsitektur umum dari sistem ekstraksi fitur gambar (*image encoder*) yang dilatih menggunakan *auxiliary loss*. *Encoder* menerima masukan berupa citra, kemudian diproses melalui serangkaian lapisan konvolusi (Conv), operasi *pooling*, dan lapisan *fully connected* (FC) untuk menghasilkan vektor fitur laten. Vektor ini merepresentasikan informasi semantik dari gambar dan digunakan dalam berbagai modul lanjutan. Dalam ilustrasi tersebut, vektor hasil *encoding* digunakan sebagai masukan untuk *global input model* yang menghasilkan fitur gabungan dari citra dan kondisi sistem, sebelum akhirnya digunakan dalam model kebijakan (*policy network*) yang dilatih dengan pendekatan *reinforcement learning*. Meskipun ilustrasi tersebut ditujukan untuk tugas pengendalian robotik, alur proses ekstraksi fitur gambarnya serupa dengan *pipeline image captioning*, di mana output CNN digunakan sebagai representasi visual yang akan diterjemahkan menjadi *caption* teks oleh komponen *decoder*.



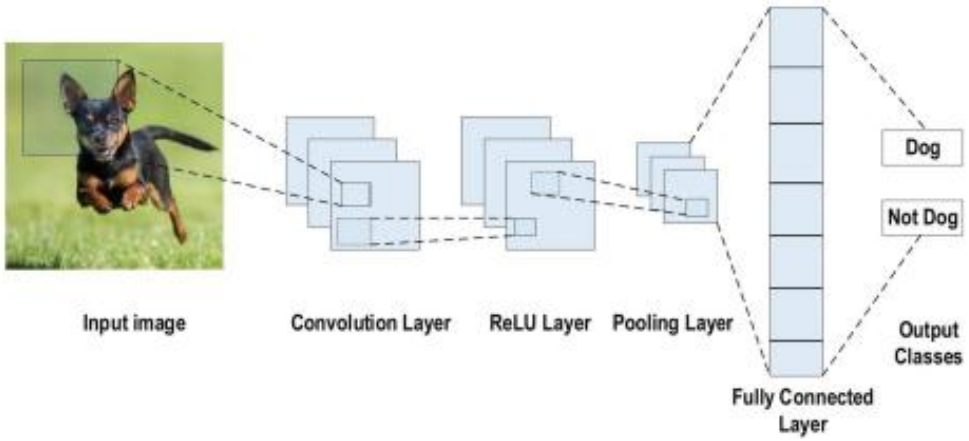
Gambar 2.4 Arsitektur *Image Encoder* yang Dilatih dengan *Auxiliary Loss* (Oliva dkk., 2022)

Penerapan *image encoder* pada tugas *image aesthetic captioning* sangat penting karena *caption* yang dihasilkan tidak hanya mendeskripsikan objek tetapi juga kualitas estetika seperti pencahayaan, kedalaman bidang, dan komposisi. Oleh karena itu, *encoder* harus mampu menangkap fitur-fitur visual yang halus dan semantik tinggi. Penggunaan *pretrained* CNN sebagai *encoder* juga memungkinkan pemanfaatan pengetahuan dari korpus visual besar seperti ImageNet, sehingga meningkatkan generalisasi pada *dataset* yang lebih kecil.

a. *Convolutional Neural Network* (CNN)

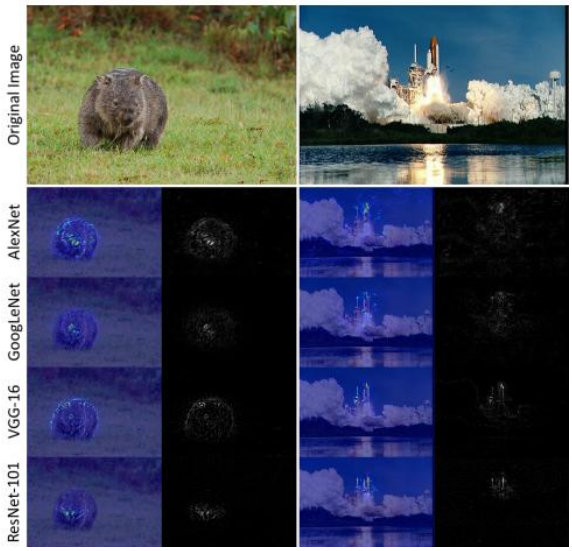
Convolutional Neural Network (CNN) adalah arsitektur deep learning yang dirancang khusus untuk memproses data yang memiliki struktur grid seperti gambar. CNN bekerja dengan menerapkan filter (*kernel*) secara lokal ke seluruh citra untuk mengekstraksi fitur seperti tepi, sudut, dan tekstur pada tahap awal, kemudian membangun representasi tingkat tinggi seperti objek dan pola pada lapisan yang lebih dalam (Alzubaidi dkk., 2021). Setiap filter belajar mendeteksi pola tertentu melalui proses pelatihan, sehingga CNN dapat membedakan berbagai objek atau elemen visual dalam gambar. CNN dikenal efektif dalam mengurangi dimensi input tanpa kehilangan informasi penting melalui operasi *pooling*, serta memanfaatkan *parameter sharing* untuk efisiensi komputasi.

Struktur CNN terdiri dari beberapa jenis lapisan utama, yaitu lapisan konvolusi, lapisan aktivasi (seperti ReLU), lapisan *pooling* (seperti *max pooling*), dan lapisan sepenuhnya terhubung (*fully connected*). Lapisan konvolusi bertugas mengekstrak fitur, sementara *pooling* mereduksi ukuran fitur map untuk mengurangi kompleksitas dan *overfitting*. ReLU digunakan untuk menambahkan non-linearitas agar model mampu mempelajari hubungan yang kompleks. Pada bagian akhir, lapisan *fully connected* bertindak sebagai *classifier* atau digunakan untuk transformasi akhir dalam *pipeline* lain seperti *captioning*. Gambar 2.5 memberikan ilustrasi mengenai struktur CNN seperti yang dijelaskan sebelumnya.



Gambar 2.5 Skema Umum Arsitektur CNN (Alzubaidi dkk., 2021)

Keunggulan CNN terletak pada kemampuannya membangun representasi hierarkis. Lapisan awal fokus pada fitur dasar, seperti garis dan tepi, sedangkan lapisan tengah dan akhir mengenali bagian-bagian objek dan keseluruhan struktur. Representasi hierarkis ini membuat CNN sangat kuat dalam berbagai tugas visi komputer, termasuk klasifikasi citra, deteksi objek, segmentasi semantik, hingga ekstraksi fitur dalam *model captioning*.



Gambar 2.6 Visualisasi Hasil Aktivasi Filter Konvolusi dengan *Guided Backpropagation* (Grün dkk., 2016)

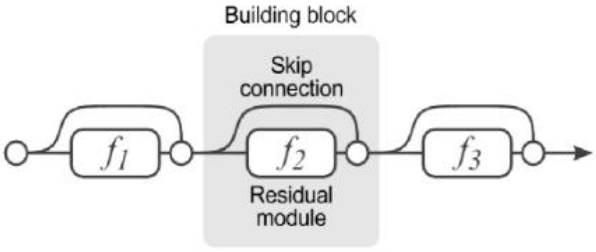
Gambar 2.6 menunjukkan perbandingan visualisasi aktivasi dari beberapa arsitektur CNN popular yaitu AlexNet, GoogLeNet, VGG-16, dan ResNet-101 pada dua buah citra yang berbeda seekor *wombat* dan peluncuran *space shuttle*. Visualisasi ini dihasilkan menggunakan teknik *Guided Backpropagation*, yang merekonstruksi kembali area pada citra *input* yang paling berkontribusi terhadap klasifikasi yang dilakukan oleh jaringan.

Terlihat bahwa setiap arsitektur memiliki pola aktivasi yang berbeda terhadap objek yang sama. AlexNet dan GoogLeNet, misalnya, menghasilkan aktivasi yang lebih menyebar, sedangkan VGG-16 dan ResNet-101 menunjukkan fokus aktivasi yang lebih terlokalisasi pada bagian-bagian penting dari objek, seperti wajah *wombat* atau ekor asap roket. Aktivasi ResNet-101 tampak paling bersih dan tajam, menunjukkan kemampuannya untuk memfokuskan perhatian pada fitur-fitur semantik penting dari gambar (Grün dkk., 2016).

Kedua citra diklasifikasikan dengan benar oleh keempat jaringan, namun visualisasi ini menggambarkan dengan jelas bagaimana kedalaman dan kompleksitas arsitektur CNN memengaruhi cara jaringan memahami struktur visual. Hal ini sekaligus menjadi bukti bahwa representasi internal yang dibentuk oleh CNN menjadi semakin selektif dan informatif seiring dengan bertambahnya jumlah lapisan konvolusi yang dimiliki.

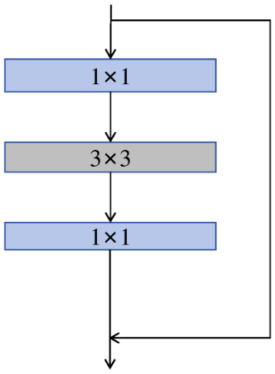
b. *Residual Network* 101 (ResNet-101)

ResNet-101 adalah varian dalam keluarga *Residual Network* (ResNet) yang terdiri dari 101 lapisan untuk mengatasi permasalahan *vanishing gradient* pada pelatihan jaringan dalam. ResNet memanfaatkan pendekatan *residual learning*, di mana jaringan tidak langsung belajar representasi penuh dari *input* ke *output*, tetapi hanya mempelajari perubahannya (residu). Hal ini diwujudkan dalam bentuk *skip connections* atau *shortcut connections* yang memungkinkan informasi dari satu lapisan dilewatkan langsung ke lapisan yang lebih dalam tanpa dimodifikasi (Panigrahi dkk., 2024).



Gambar 2.7 Struktur Blok Residual dengan *Skip Connection* (Xu dkk., 2024)

Gambar 2.7 memperlihatkan struktur dasar dari blok residual, di mana input dilewatkan langsung ke output melalui jalur pintas (*shortcut path*) dan dijumlahkan dengan *output* dari lapisan non-linear. Pendekatan ini mempermudah proses pembelajaran karena jaringan hanya perlu mempelajari fungsi perbaikan terhadap input awal, bukan representasi baru dari nol. Hal ini terbukti efektif dalam mempertahankan performa jaringan ketika jumlah lapisan meningkat secara signifikan.



Gambar 2.8 Arsitektur *Bottleneck* dalam ResNet (Xu dkk., 2024)

Pada ResNet-101, setiap blok residual didesain dengan struktur *bottleneck*, seperti yang ditunjukkan pada Gambar 2.8. Blok ini terdiri dari tiga lapisan konvolusi berurutan; konvolusi 1×1 di awal digunakan untuk menurunkan dimensi, diikuti oleh konvolusi 3×3 sebagai ekstraksi fitur utama, dan diakhiri dengan konvolusi 1×1 untuk menaikkan kembali dimensi ke ukuran semula. Desain ini bertujuan untuk menjaga efisiensi komputasi tanpa mengorbankan kapasitas

representasi jaringan. Pendekatan ini memungkinkan ResNet-101 memiliki 101 lapisan yang dalam namun tetap dapat dilatih secara stabil.

Dalam konteks *image captioning*, ResNet-101 dapat dimanfaatkan sebagai backbone untuk mengekstraksi fitur dari gambar. *Output* dari beberapa blok awal diambil sebagai *spatial feature map*, sedangkan *output global pooling* dari lapisan akhir digunakan sebagai *global feature vector*. Fitur-fitur ini kemudian dapat diteruskan ke *decoder* berbasis LSTM untuk menghasilkan *caption* yang sesuai.

2.2.5 *Text Decoder*

Text decoder merupakan komponen utama dalam arsitektur *image captioning* yang bertugas untuk menghasilkan kalimat deskriptif berdasarkan representasi visual dari gambar. Setelah fitur visual diekstraksi oleh *encoder* (CNN seperti ResNet-101), *decoder* akan menerima representasi tersebut sebagai konteks awal dan kemudian secara *autoregressive* menghasilkan satu kata demi satu kata dalam kalimat *caption*.

Pada umumnya, *decoder* yang digunakan dalam arsitektur CNN-LSTM mengikuti kerangka kerja *sequence-to-sequence* (seq2seq), di mana *Long Short-Term Memory* (LSTM) digunakan untuk memproses urutan kata dalam bentuk *word embeddings* dan mengestimasikan distribusi probabilitas kata selanjutnya. *Decoder* dilatih untuk meminimalkan loss antara prediksi kata dan *ground truth caption* secara berurutan menggunakan teknik *Teacher Forcing*.

Selain LSTM, *decoder* biasanya juga memanfaatkan *layer embedding* yang bertugas mengubah kata-kata menjadi vektor berdimensi tetap. Proses *decoding* dilakukan sampai token *end-of-sequence* (`<end>`) dihasilkan, atau sampai batas maksimum panjang kalimat tercapai. Untuk meningkatkan kualitas *caption* yang dihasilkan, strategi *decoding* seperti *Beam Search* sering digunakan sebagai alternatif dari *greedy decoding*, karena mampu mengeksplorasi lebih banyak kemungkinan kombinasi kata yang lebih bermakna secara semantic (Karpathy & Fei-Fei, 2015).

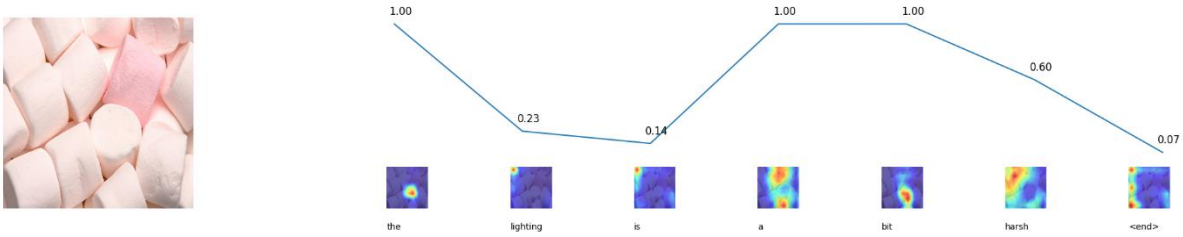
a. *Long Short-Term Memory* (LSTM)

Long Short-Term Memory (LSTM) adalah tipe jaringan saraf rekursif yang dirancang untuk menangani masalah urutan data, seperti teks atau *video*, dengan mengatasi masalah gradien yang hilang atau meledak (*exploding gradient*). Struktur unik LSTM terdiri dari tiga gerbang, yaitu *input*, *forget*, dan *output*, yang memungkinkan model untuk memilih informasi mana yang akan disimpan, dilupakan, atau diteruskan ke langkah berikutnya. Dengan kemampuan ini, LSTM sangat cocok untuk tugas seperti prediksi urutan dan generasi teks (Staudemeyer & Morris, 2019).

Attention Mechanism adalah teknik yang memungkinkan model memfokuskan perhatian pada bagian data yang paling relevan selama proses pelatihan. Dalam pengkodean *sequence-to-sequence*, mekanisme ini membantu model menentukan bagian mana dari *input* yang harus diberi bobot lebih tinggi untuk menghasilkan output yang lebih akurat. Penggunaan *Attention Mechanism* dalam kombinasi dengan LSTM meningkatkan efisiensi model dalam menangani data dengan dimensi tinggi atau urutan yang panjang (Kang dkk., 2023).

Dalam kerangka kerja *encoder-decoder*, LSTM sering digunakan sebagai *decoder* untuk menghasilkan urutan *output* berdasarkan representasi internal dari data *input*. Representasi ini dikodekan oleh *encoder* seperti CNN untuk citra, menjadi vektor latar yang kemudian diterjemahkan oleh LSTM ke dalam urutan *output*, seperti teks deskriptif. Kombinasi dengan *Attention Mechanism* memperkuat kemampuan *decoder* untuk menghasilkan *output* yang lebih relevan dengan memanfaatkan konteks *input* secara selektif (Hosna dkk., 2022). Contoh hasil

generasi deskripsi yang dibuat oleh model *encoder-decoder* CNN-LSTM dapat dilihat pada Gambar *2.9*.



Gambar 2.9 Hasil Generasi Deskripsi Model CNN-LSTM

Model tersebut menerima *input* berupa sebuah citra. Citra tersebut diolah dengan menggunakan CNN untuk mengambil fiturnya dan LSTM mengambil fitur tersebut sebagai inputan yang kemudian menghasilkan deskripsi lengkap dari token kata sebelumnya. Hasil akhir berupa kalimat "*the lighting is a bit harsh*".

b. Caption Generation Strategy

Proses menghasilkan caption dilakukan melalui strategi *decoding* pada tahap inferensi. Tujuan dari proses ini adalah memilih urutan kata yang paling sesuai dan bermakna berdasarkan distribusi probabilitas yang diprediksi oleh *decoder* pada setiap langkah waktu (*time step*).

Strategi *decoding* paling dasar adalah *greedy decoding*, yaitu memilih kata dengan probabilitas tertinggi pada setiap langkah. Meskipun sederhana dan efisien, metode ini sering kali menghasilkan *caption* yang terlalu umum atau kurang bervariasi karena hanya mempertimbangkan satu jalur prediksi.

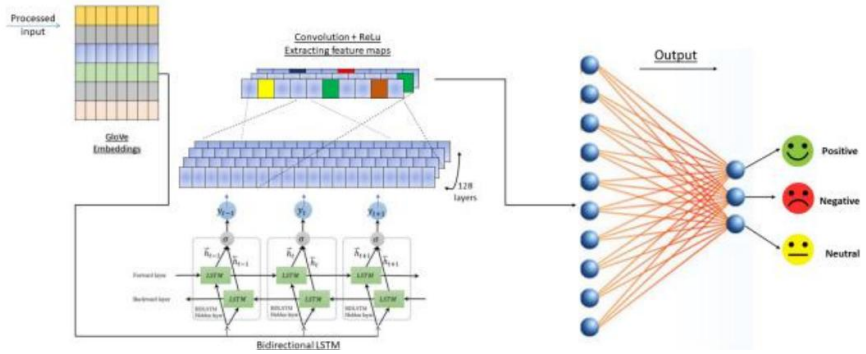
Untuk meningkatkan kualitas hasil *caption*, LSTM sering menggunakan pendekatan *Beam Search*. *Beam Search* adalah strategi pencarian heuristik yang menjaga beberapa jalur prediksi terbaik secara bersamaan berdasarkan skor kumulatif log-probabilitas. Misalnya, pada *beam size* = 5, model mempertahankan 5 kandidat *caption* terbaik pada setiap langkah *decoding*. Setelah mencapai token akhir (`<end>`) atau panjang maksimum, *caption* dengan skor tertinggi di antara kandidat tersebut dipilih sebagai *output* akhir.

Penggunaan *Beam Search* terbukti menghasilkan *caption* yang lebih konsisten, informatif, dan koheren dibandingkan dengan *greedy decoding*. Namun, pemilihan ukuran *beam* harus disesuaikan karena ukuran yang terlalu besar dapat menyebabkan *caption* menjadi terlalu panjang atau terlalu umum (*overgeneralized*) (Wiseman & Rush, 2016).

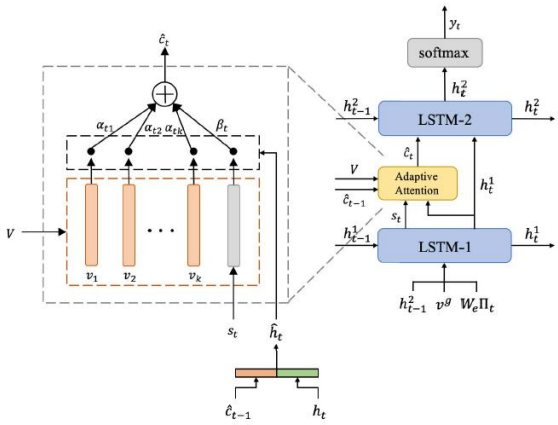
2.2.6 CNN-LSTM *Hybrid* Model

Hybrid model berbasis CNN dan LSTM menjadi salah satu pendekatan arsitektural yang banyak digunakan untuk memadukan representasi spasial dan urutan dalam satu kesatuan model. Dalam konteks analisis teks, CNN memiliki kemampuan mengekstraksi fitur lokal dan pola *n*-gram penting dari *embedding* kata, sedangkan LSTM unggul dalam menangkap dependensi jangka panjang pada urutan kata. Penelitian (Venkatesh dkk., 2021) menunjukkan bahwa integrasi CNN dan LSTM memungkinkan model untuk menangkap baik hubungan semantik lokal (melalui CNN) maupun hubungan semantik jangka panjang (melalui LSTM), sehingga sangat efektif untuk tugas-tugas seperti klasifikasi sentimen dalam teks Twitter yang tidak terstruktur dan penuh *slang*. CNN digunakan sebagai *feature extractor* terhadap *embedding kata* (seperti GloVe), lalu hasil ekstraksi fitur ini diteruskan ke LSTM untuk menghasilkan representasi urutan yang lebih kontekstual.

Implementasi dari arsitektur ini ditunjukkan oleh (Venkatesh dkk., 2021) dalam tugas klasifikasi sentimen Twitter menggunakan GloVe *embedding* sebagai representasi kata berbasis statistik *co-occurence* global, lalu meneruskannya ke CNN untuk mendeteksi fitur lokal seperti kata bernuansa emosi atau opini. Hasil ekstraksi ini kemudian diproses oleh Bidirectional LSTM yang memproses sekuen dari dua arah (maju dan mundur) untuk menangkap konteks penuh dari kalimat. Arsitektur ini digambarkan pada Gambar 2.10, di mana input *embedding* diproses oleh CNN dengan 128 filter, dilanjutkan oleh layer LSTM yang menghasilkan vektor laten sebagai input ke layer *fully-connected*. *Output* akhir adalah klasifikasi ke dalam tiga kategori sentimen: positif, negatif, atau netral. Arsitektur ini menunjukkan bahwa integrasi CNN dan LSTM menciptakan pipeline yang kuat untuk memahami baik aspek lokal maupun temporal dalam teks.



Gambar 2.10 *Hybrid* CNN-LSTM Model dengan Glove (Venkatesh dkk., 2021)



Gambar 2.11 Arsitektur *Encoder-Decoder* CNN-LSTM untuk *Captioning* (Zou dkk., 2020)

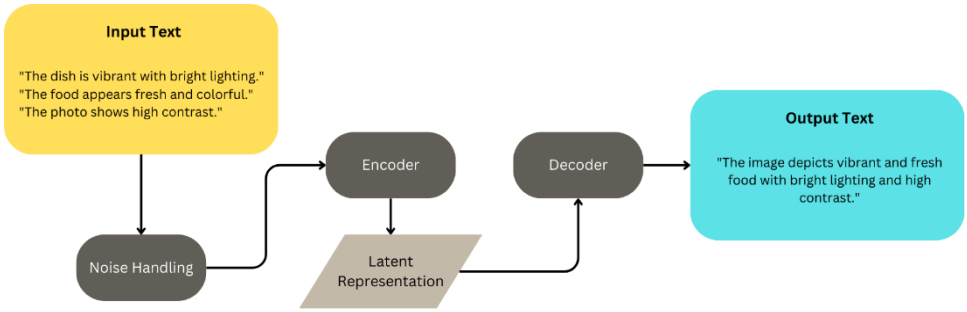
Pada domain yang berbeda, seperti *captioning* gambar makanan yang memiliki elemen multimodal, pendekatan serupa juga digunakan namun dengan sedikit adaptasi. Dalam penelitian (Zou dkk., 2020), *encoder* berbasis CNN (ResNet-101) digunakan untuk mengekstraksi fitur visual dari gambar, sementara *decoder* berbasis LSTM bertugas untuk menghasilkan urutan kata yang membentuk *caption*. Arsitektur ini disebut sebagai *image captioning encoder-decoder framework*, di mana CNN berperan sebagai *image encoder* dan LSTM sebagai *text decoder*. *Decoder* ini bukan hanya sekadar LSTM biasa, melainkan dilengkapi dengan mekanisme *attention* untuk fokus pada bagian tertentu dari fitur visual saat menghasilkan tiap token dalam *caption*. Dengan demikian, arsitektur ini masih masuk dalam kategori hybrid CNN-LSTM, namun dengan orientasi pada visi ke teks. Arsitektur yang digunakan oleh (Zou dkk., 2020) dapat dilihat pada Gambar 2.11.

Struktur *hybrid* CNN-LSTM ini mencerminkan arsitektur *encoder-decoder*, yang awalnya diperkenalkan dalam bidang penerjemahan mesin oleh (Sutskever dkk., 2014), dan telah diadaptasi secara luas di bidang computer vision dan NLP. Dalam versi *image* captioning, *encoder* adalah CNN *pretrained* seperti ResNet-101 atau Inception yang menghasilkan representasi visual, sementara *decoder* adalah LSTM yang dilatih untuk memprediksi urutan kata berdasarkan konteks visual. Beberapa varian dari arsitektur ini menambahkan *attention mechanism*, yang memungkinkan model untuk fokus pada bagian tertentu dari gambar saat menghasilkan token tertentu. Penggunaan strategi *decoding* seperti *greedy* atau *beam search* juga penting dalam tahap inferensi. Dengan pendekatan ini, *pipeline captioning* dapat menangkap semantik visual secara efektif dan mengartikulasikannya ke dalam bahasa alami. Integrasi *embedding pretrained* seperti GloVe semakin memperkuat pemahaman semantik di sisi *decoder*, membuat arsitektur ini tetap relevan dan kompetitif dalam banyak studi terkini (Venkatesh dkk., 2021).

2.2.7 *Unsupervised Text Summarization*

Unsupervised Text Summarization adalah metode untuk menghasilkan ringkasan teks tanpa menggunakan data berlabel atau anotasi manual. Pendekatan ini memanfaatkan struktur dan pola dalam teks untuk mengekstrak atau merangkum informasi penting. Teknik ini sering digunakan ketika *dataset* berlabel sulit diperoleh atau memerlukan banyak waktu dan biaya untuk membuatnya. Metode yang umum mencakup pendekatan berbasis frekuensi kata seperti TF-IDF, klasterisasi, atau penggunaan model generatif untuk merepresentasikan data dalam bentuk yang lebih ringkas dan relevan (Manojkumar dkk., 2022). *Unsupervised Text Summarization* bertujuan untuk menyaring informasi utama dari teks sambil mempertahankan koherensi dan relevansi.

Denoising Auto-Encoder (DAE) adalah model generatif yang digunakan untuk merekonstruksi data dari versi yang dirusak (*noisy*). Dalam konteks *Unsupervised Text Summarization*, DAE mempelajari representasi laten dari teks yang mencerminkan informasi inti, sambil mengabaikan elemen-elemen yang tidak relevan atau redundan. Prosesnya dimulai dengan merusak teks *input*, misalnya dengan menghapus atau mengganti kata-kata tertentu, lalu melatih model untuk memprediksi versi asli teks tersebut (Bengio dkk., 2013). Pada tahap *encoding* model DAE, teks *input* yang telah dirusak diproses menggunakan jaringan neural untuk diubah menjadi representasi laten yang mencerminkan informasi inti. Selanjutnya, pada tahap *decoding*, representasi laten tersebut diolah oleh *decoder* untuk merekonstruksi teks asli, sambil memanfaatkan distribusi probabilistik untuk menyaring informasi penting dan menghilangkan elemen yang tidak relevan.



Gambar 2.12 Ilustrasi Cara Kerja DAE pada *Unsupervised Text Summarization*

Dengan mempelajari pola-pola penting dari data, DAE mampu menghasilkan ringkasan yang koheren dan informatif, bahkan tanpa panduan eksplisit dalam bentuk label. Pendekatan

ini sangat efektif untuk merangkum teks yang panjang dan tidak terstruktur, seperti ulasan pengguna atau dokumen artikel, karena dapat menangkap inti dari informasi yang relevan secara otomatis (Manojkumar dkk., 2022). Gambar *2.12* memperlihatkan bahwa model DAE menerima *input* berupa beberapa deskripsi pada aspek estetika. *Encoder* dari DAE mengubah deskripsi ini menjadi representasi laten, yang menangkap inti informasi tanpa redundansi. *Decoder* memanfaatkan representasi ini untuk menghasilkan ringkasan teks akhir yang koheren dan mencakup semua inti kalimat pada *input* teksnya.

2.2.8 *Term Frequency-Inverse Document Frequency* (TF-IDF)

Term Frequency–Inverse Document Frequency (TF-IDF) adalah salah satu metode pembobotan kata paling populer dalam bidang *Information Retrieval* dan *Natural Language Processing* (NLP). Metode ini mengukur seberapa penting sebuah kata dalam suatu dokumen relatif terhadap keseluruhan korpus. TF-IDF bekerja berdasarkan dua prinsip utama yaitu frekuensi kata dalam dokumen tertentu (*term frequency*) dan kebalikannya dari frekuensi dokumen yang mengandung kata tersebut dalam seluruh korpus (*inverse document frequency*). Perhitungan umum dari metode TF-IDF dapat dilihat di (2.1.

$$\mathrm{TF-IDF}(w, d, D) = f_{w,d} \cdot \log\left(\frac{|D|}{f_{w,D}}\right) \tag{2.1}$$

Di mana:
$f_{w,d}$ adalah jumlah kemunculan kata w dalam dokumen d
$f_{w,D}$ adalah jumlah dokumen dalam korpus D yang mengandung kata w
$|D|$ adalah jumlah total dokumen

Kata-kata yang sering muncul dalam sebuah dokumen namun jarang muncul di korpus secara keseluruhan akan mendapatkan bobot TF-IDF yang tinggi, menandakan bahwa kata tersebut diskriminatif dan relevan secara kontekstual. Metode ini telah digunakan luas dalam berbagai aplikasi seperti klasifikasi teks, ekstraksi kata kunci, penambangan opini, dan pencarian informasi. Namun, meskipun efektif, TF-IDF memiliki keterbatasan dalam menangkap hubungan semantik antar kata. Beberapa penelitian menunjukkan bahwa TF-IDF tidak mampu mengenali sinonim atau bentuk turunan kata, sehingga gagal menghubungkan makna kata seperti "*drug*" dan "*drugs*" (Das dkk., 2023; Jalilifard dkk., 2021).

2.2.9 *Informativeness Score*

Dalam konteks *image aesthetic captioning*, penelitian (Zou dkk., 2020) memperkenalkan suatu metrik yang disebut *Informativeness Score*. Metrik tersebut digunakan untuk menyaring *caption* yang informatif dan menghindari komentar-komentar generik seperti "*Nice color*" atau "*Beautiful shot*". Metode ini terinspirasi langsung dari prinsip TF-IDF dengan asumsi bahwa *n*-gram yang jarang muncul di seluruh korpus mengandung informasi yang lebih berharga. Perhitungan *Informativeness Score* dapat dilihat pada (2.2.

$$\mathrm{Score} = -\frac{1}{2}\left[\log\prod_{i}^{N} P(u_i) + \log\prod_{j}^{M} P(b_j)\right] \tag{2.2}$$

Di mana:
u_i adalah *unigram* berupa kata benda (*noun*)
b_j adalah *bigram* berupa pasangan *descriptor-object*

$P(\omega) = \frac{C_\omega}{\sum_i C_i}$ probabilitas kemunculan *n*-gram ω dalam korpus

Skor ini menghasilkan nilai kecil jika *caption* terdiri dari *unigram* dan *bigram* yang sangat umum, dan nilai tinggi jika *caption* mengandung kata-kata yang langka namun bermakna.

Tabel 2.2 Perbedaan TF-IDF dan *Informativeness Score*

Aspek	**TF-IDF**	***Informativeness Score***
Tujuan	Menilai pentingnya kata dalam konteks dokumen dan korpus	Menyaring kalimat berdasarkan *rarity* *n*-gram
Input	Per dokumen dan korpus	Per kalimat dan korpus
Formula	TF × log(IDF)	Log probabilitas total *unigram* dan *bigram*
Skala	Linear terhadap kata	Logaritmik terhadap kalimat
Fokus	Relevansi kata	Informatifnya *caption* secara keseluruhan
Output	Per kata	Per kalimat

Perbedaan utama antara TF-IDF dan *Informativeness Score* adalah pada bentuk dan tujuan skala pembobotan. Perbedaannya dapat dilihat pada Tabel 2.2. Dengan kata lain, TF-IDF bersifat kata-sentris, sementara *Informativeness Score* bersifat kalimat-sentris.

2.2.10 *Latent Dirichlet Allocation* (LDA)

Latent Dirichlet Allocation (LDA) adalah salah satu metode yang umum digunakan untuk melakukan ekstraksi topik secara tidak terawasi atau *unsupervised topic modeling* dalam kumpulan dokumen teks. Penelitian (Blei dkk., 2003) memperkenalkan model LDA dan sejak itu telah menjadi metode utama dalam berbagai aplikasi seperti klasifikasi dokumen, penyusunan ringkasan teks, analisis opini, dan ekstraksi tema dari media sosial.

LDA bekerja dengan mengasumsikan bahwa setiap dokumen merupakan campuran dari beberapa topik laten, dan setiap topik merupakan distribusi probabilistik atas sekumpulan kata. Proses generatif dari LDA dijelaskan sebagai berikut, pertama-tama untuk setiap dokumen diambil distribusi topik dari distribusi *Dirichlet*. Kemudian, untuk setiap kata dalam dokumen tersebut, dipilih sebuah topik dari distribusi tersebut, dan selanjutnya dipilih sebuah kata dari distribusi kata-kata yang terkait dengan topik tersebut. Secara matematis penelitian (Blei dkk., 2003) memodelkan probabilitas gabungan dari dokumen, topik, dan kata-kata atau probabilitas dari sebuah korpus sebagai (2.3).

$$p(D|\alpha,\beta) = \prod_{d=1}^{M} \int p(\theta_d|\alpha) \left(\prod_{n=1}^{N_d} \sum_{z_{dn}} p(z_{dn}|\theta_d) p(w_{dn}|z_{dn},\beta) \right) d\theta_d \qquad (2.3)$$

Di mana:
D adalah seluruh dokumen dalam korpus
M adalah jumlah dokumen
N_d adalah jumlah kata dalam dokumen ke- d
θ_d adalah distribusi topik dalam dokumen ke- d
z_{dn} adalah topik untuk kata ke- n dalam dokumen ke- d
w_{dn} adalah kata ke- n dalam dokumen ke- d

α, β merupakan parameter *Dirichlet*

Dalam konteks penelitian (Zou dkk., 2020), LDA digunakan untuk mengelompokkan caption estetika menjadi beberapa aspek visual, yaitu *Composition*, *Color & Light*, *Depth of Field & Focus*, dan *General Impression*. *Caption-caption* yang telah difilter menggunakan *Informativeness Score* akan diproses menggunakan model LDA untuk menemukan distribusi topik laten. Setiap topik yang dihasilkan kemudian dievaluasi berdasarkan kata-kata dominannya, dan dipetakan secara manual ke dalam salah satu dari enam aspek estetika tersebut. Misalnya, topik yang banyak mengandung kata seperti "*blur*", "*sharp*", atau "*focus*" akan dipetakan ke dalam aspek *Depth of Field & Focus*.

Keunggulan utama dari LDA adalah kemampuannya dalam menemukan struktur topik tersembunyi dari data teks tanpa membutuhkan label eksplisit. Namun, interpretasi topik yang dihasilkan masih bergantung pada analisis manual, karena LDA hanya memberikan distribusi kata, bukan label topik secara langsung. Oleh karena itu, kombinasi antara inferensi statistik dan interpretasi manusia sangat diperlukan dalam penggunaan LDA untuk klasifikasi aspek visual dalam domain estetika gambar. LDA memungkinkan untuk membuat *subset* data yang merepresentasikan masing-masing aspek estetika tanpa harus melakukan pelabelan manual secara langsung. *Subset-subset* tersebut kemudian digunakan sebagai data pelatihan untuk model *captioning* yang fokus pada satu aspek visual tertentu.

2.2.11 *Global Vectors for Word Representation* (GloVe)

Global Vectors for Word Representation (GloVe) adalah metode pembelajaran representasi vektor kata berbasis statistik yang dikembangkan oleh penelitian (Pennington dkk., 2014) di Stanford University. GloVe dirancang untuk menggabungkan keunggulan dari dua pendekatan utama dalam pembelajaran *embedding* kata, yaitu metode global *matrix factorization* seperti *Latent Semantic Analysis* (LSA), dan metode *local context window prediction* seperti *Skip-gram* dan CBOW (Mikolov dkk., 2013).

Berbeda dengan metode prediksi lokal yang mempelajari vektor kata melalui konteks terbatas dalam jendela kalimat, GloVe menggunakan statistik *co-occurence* global dari seluruh korpus teks untuk membentuk *embedding* kata. Gagasan utama dalam GloVe adalah bahwa rasio probabilitas *co-occurence* antar kata mengandung informasi semantik penting yang dapat ditangkap dalam bentuk relasi linear dalam ruang vektor. Sebagai gambaran, hubungan semantik antara kata "*ice*" dan "*steam*" dapat direpresentasikan melalui perbandingan rasio probabilitas kemunculannya terhadap kata-kata konteks seperti "*solid*", "*gas*", atau "*fashion*". Jika rasio probabilitas kemunculan kata konteks k yang diberikan kata target "*ice*" dibandingkan dengan "*steam*" memiliki nilai tinggi untuk kata "*solid*", maka hal ini menunjukkan bahwa kata tersebut lebih erat kaitannya dengan "*ice*". Rasio semacam ini menjadi dasar dalam model GloVe untuk membedakan nuansa makna antar kata berdasarkan distribusi *co-occurence* global di seluruh korpus. Model GloVe memodelkan representasi vektor dengan pendekatan regresi *log-bilinear* terhadap logaritma frekuensi *co-occurence* antara kata target dan kata konteks. Fungsi objektif GloVe dinyatakan dalam bentuk fungsi biaya pada (2.4).

$$J = \sum_{i,j=1}^{V} f\left(X_{ij}\right)\left(w_i^T \tilde{w}_j + b_i + \tilde{b}_j - \log X_{ij}\right)^2 \tag{2.4}$$

Di mana:
X_{ij} adalah jumlah kemunculan kata j dalam konteks kata i

w_i & $\tilde{w}_j$ adalah vektor kata dan konteks yang akan dilatih

b_i & $\tilde{b}_j$ adalah bias untuk kata dan konteks

$f(X_{ij})$ adalah fungsi pembobotan untuk mengendalikan kontribusi pasangan kata berdasarkan frekuensi

Fungsi pembobotan $f(x)$ dirancang agar mengabaikan *co-occurence* yang sangat jarang atau *noisy* maupun yang terlalu sering atau tidak informatif dan umumnya dapat dirumuskan dalam (2.5).

$$f(x) = \begin{cases} (x/x_{max})^{\alpha}, & x < x_{max} \\ 1, & x \geq x_{max} \end{cases} \quad (2.5)$$

dengan nilai parameter empiris $\alpha = \frac{3}{4}$ dan $x_{max} = 100$.

Keunggulan utama dari GloVe adalah kemampuannya menangkap struktur semantik linear yang bermakna. Sebagai contoh, relasi seperti "*king*" – "*man*" + "*woman*" ≈ "*queen*" dapat direpresentasikan dalam ruang vektor hasil pelatihan GloVe. Selain itu, GloVe juga menunjukkan performa unggul dalam berbagai tugas NLP seperti *word analogy*, *similarity*, dan *Named Entity Recognition* (NER), serta memungkinkan efisiensi pelatihan yang tinggi karena hanya menggunakan elemen non-*zero* dari matriks *co-occurrence* (Pennington dkk., 2014).

Dalam implementasi praktis, model GloVe disediakan dalam berbagai versi dimensi vektor dan ukuran korpus pelatihan. Model ini telah dilatih secara *pretrained* oleh tim dari Stanford NLP pada beberapa korpus besar, dan tersedia secara bebas untuk digunakan dalam berbagai aplikasi NLP. Salah satu versi yang paling umum digunakan adalah GloVe 300D, yaitu model GloVe dengan representasi 300 dimensi per kata. Artinya, setiap kata dalam korpus diwakili oleh sebuah vektor dengan panjang 300 elemen numerik (*float*), yang mengandung informasi semantik dari kata tersebut. Selain GloVe 300D, tersedia pula variasi lain seperti;

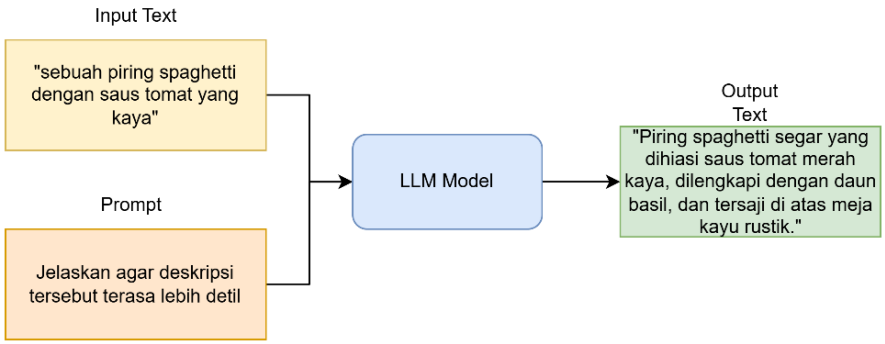
a) GloVe 50D yaitu vektor berdimensi 50, lebih ringan dan cepat, namun kurang akurat dalam menangkap nuansa semantik.
b) GloVe 100D dan 200D yang merupakan kompromi antara efisiensi dan akurasi.
c) GloVe 300D (Wikipedia + Gigaword) yang dilatih pada gabungan data Wikipedia 2014 dan Gigaword 5.
d) GloVe 840B 300D adalah versi 300D yang dilatih pada korpus sangat besar (Common Crawl 840 miliar token), cocok untuk aplikasi yang membutuhkan cakupan kosakata yang sangat luas.

Dalam Tugas Akhir ini, digunakan GloVe 300D sebagai *pretrained embedding* yang memetakan kata-kata dalam *caption* menjadi representasi numerik padat yang sarat makna. *Embedding* ini kemudian digunakan sebagai *input* untuk model CNN-LSTM, yang menghasilkan *caption* berdasarkan representasi semantik tersebut.

2.2.12 *Large Language* Model (LLM)

Large Language Model (LLM) adalah sistem kecerdasan buatan canggih yang dirancang untuk memahami dan menghasilkan teks dengan kualitas menyerupai manusia. Model ini dibangun menggunakan arsitektur *transformer*, yang memanfaatkan mekanisme perhatian (*attention mechanism*) untuk memahami hubungan antara kata-kata dalam suatu urutan. Dengan dilatih pada dataset yang sangat besar dari berbagai domain, LLM mampu menjalankan beragam tugas pemrosesan bahasa alami (*Natural Language Processing* atau NLP), seperti menjawab pertanyaan, meringkas informasi, menerjemahkan, hingga menghasilkan teks secara kreatif (Naveed dkk., 2023).

ChatGPT menjadi salah satu contoh terkemuka LLM yang dapat melakukan interaksi berbasis percakapan secara alami. Model ini bekerja dengan menganalisis *input* teks, memahami konteks, dan menghasilkan respons dengan mempertimbangkan hubungan antar kata melalui mekanisme perhatian (*attention mechanism*). Keunggulannya terletak pada kemampuannya untuk memahami instruksi yang kompleks dan menghasilkan keluaran yang menyerupai gaya komunikasi manusia (Kasneci dkk., 2023). Ilustrasi penggunaan LLM dapat dilihat pada Gambar 2.13.



Gambar 2.13 Ilustrasi Penggunaan LLM

Cara kerja LLM dimulai dengan proses tokenisasi, di mana teks *input* dipecah menjadi unit-unit kecil yang disebut *token* (kata, sub-kata, atau karakter). Setiap *token* ini kemudian diubah menjadi representasi vektor, yang diproses lebih lanjut menggunakan mekanisme perhatian untuk menentukan bagian teks mana yang paling relevan dalam konteks tertentu. Berdasarkan pola yang telah dipelajari selama pelatihan, model menghasilkan prediksi untuk token berikutnya secara iteratif hingga terbentuk keluaran yang utuh (Naveed dkk., 2023).

2.2.13 Pemolesan Teks atau *Retouch* dengan *Large Language Model*

Retouch merujuk pada proses pemolesan kalimat deskriptif hasil model agar menjadi lebih alami, koheren, dan estetis tanpa menghilangkan makna inti. Proses ini dilakukan setelah tahap inferensi awal oleh *model captioning*, dengan tujuan menyempurnakan struktur kalimat, pemilihan diksi, dan alur naratif agar selaras dengan gaya bahasa manusia yang baik. Proses *retouch* dilakukan dengan bantuan *Large Language Model* (LLM) yang bertindak sebagai *refiner* atau *editor* otomatis. Dalam bidang pemrosesan bahasa alami, *retouch* atau pemolesan teks merupakan proses penyempurnaan hasil generasi otomatis agar menjadi lebih alami, ekspresif, dan sesuai konteks, tanpa mengubah makna semantis yang terkandung di dalamnya. Proses ini menjadi semakin penting dalam sistem generatif yang melibatkan langkah-langkah kompleks seperti *image captioning* atau *speech translation*, di mana hasil awal sering kali masih bersifat kaku, literal, atau kurang estetis dari segi bahasa. Ilustrasi *retouch* dapat dilihat pada Gambar 2.14.

Gambar 2.14 Ilustrasi Pemolesan Kalimat

Pendekatan ini didasarkan pada penelitian terkini yang memperkenalkan teknik *Language Model Refinement*, yakni penggunaan LLM tambahan untuk menyempurnakan keluaran dari model utama. Salah satu metode yang relevan adalah *Streaming-VR (Verification and Refinement)*, yang memungkinkan LLM kedua untuk mengevaluasi dan memperbaiki kesalahan secara *real-time* saat teks sedang dihasilkan. Menurut (Ko dkk., 2025) metode ini efektif dalam mencegah propagasi kesalahan yang terjadi akibat token yang salah di awal proses generasi, sekaligus menghemat jumlah token yang harus diperbaiki dibanding metode tradisional yang memproses teks secara keseluruhan. Menurut (Chen dkk., 2024), pendekatan ini melibatkan pemberian *prompt* kepada LLM untuk memperbaiki terjemahan atau teks yang sudah ada, tanpa perlu melakukan pelatihan ulang. Dalam proses ini, LLM diberi masukan berupa pasangan kalimat sumber dan hasil awal, kemudian diminta menghasilkan versi yang lebih alami dan lancar. Proses ini bisa dilakukan berulang kali (*multi-turn refinement*), yang memungkinkan model untuk memperbaiki gaya, struktur, dan kealamian bahasa secara progresif.

(Dou dkk., 2025) memperkenalkan pendekatan *Natural Language Retouching* sebagai bagian dari sistem penyempurnaan hasil *Speech Translation* (ST) menggunakan model LLM seperti GPT-3.5-turbo. Dalam pendekatan ini, hasil terjemahan awal dari sistem ST tidak langsung digunakan, melainkan terlebih dahulu disempurnakan melalui LLM yang diberi *prompt* instruksional. Model kemudian memperbaiki gaya bahasa, memperhalus struktur kalimat, dan memastikan koherensi antar frasa, sehingga teks akhir menyerupai gaya bahasa manusia yang alami dan profesional. Proses ini disebut *refinement*, dan dapat dipahami sebagai bentuk *retouch* pada tingkat kalimat. Salah satu kekuatan utama dari pendekatan *retouch* berbasis LLM adalah kemampuannya untuk melakukan penyempurnaan dengan mempertahankan makna semantik. LLM tidak hanya merevisi sintaksis permukaan, tetapi juga memahami hubungan semantik antarkata, konteks domain, serta gaya bahasa yang diinginkan. Hal ini memungkinkan proses *retouch* menghasilkan teks yang tidak hanya akurat, tetapi juga komunikatif dan ekspresif, sebagaimana dicontohkan dalam penelitian (Dou dkk., 2025) pada *domain speech translation*.

2.2.14 Metrik Evaluasi

Metrik evaluasi dalam ilmu komputer merujuk pada kriteria tertentu yang digunakan untuk mengukur kinerja sistem atau algoritma. Metrik ini dapat berbasis akurasi, berbasis peringkat, berbasis eror, atau lainnya, tergantung pada jenis evaluasi yang dilakukan (Çetin dkk., 2021). Beberapa metrik evaluasi yang akan digunakan untuk mengukur kinerja model *machine learning* pada Tugas Akhir ini adalah sebagai berikut. Perhitungan umum metode TF-IDF

a. Bilingual Evaluation Understudy (BLEU)

BLEU adalah metrik evaluasi yang sering digunakan untuk mengukur performa model NLP dalam menghasilkan teks yang menyerupai referensi atau *ground truth*. BLEU membandingkan kemiripan antara hasil teks yang dihasilkan model dengan referensi berdasarkan tingkat presisi *n*-gram (potongan kata yang tumpang tindih). Skor BLEU berkisar antara 0 hingga 1, di mana nilai 0 menunjukkan hasil teks yang sangat berbeda dengan referensi, sementara nilai 1 menandakan kesesuaian penuh dengan referensi. Tiga aspek utama dalam metrik ini adalah presisi *n*-gram, penalti untuk teks pendek (*brevity penalty*), dan rata-rata bobot (*weighted average*) (Papineni dkk., t.t.).

Presisi *n*-gram mengukur seberapa sering potongan kata dalam hasil teks model muncul dalam referensi. Nilai *n* menunjukkan tingkat kecocokan, seperti 1-gram untuk kata per kata,

2-gram untuk pasangan kata secara berurutan, dan seterusnya. Contohnya, 1-gram mencocokkan kata seperti "piring" secara langsung, sementara 2-gram mengevaluasi pasangan kata, seperti "piring merah".

Penalti untuk teks pendek atau *brevity penalty* diterapkan untuk mencegah model menghasilkan teks yang sangat pendek namun terlihat sesuai. Misalnya, jika model menghasilkan "piring", sementara referensi adalah "piring merah bermotif kotak", skor BLEU akan menurun karena penalti ini. Penalti dirancang untuk memastikan teks hasil model memiliki panjang yang proporsional dengan referensi.

Rata-rata bobot atau *weighted average* digunakan untuk menghitung rata-rata logaritmik dari presisi *n*-gram. Perhitungan ini bertujuan untuk memberikan evaluasi seimbang terhadap semua tingkat *n*-gram. Dengan menggabungkan tiga elemen ini, perhitungan BLEU dapat dilihat pada Persamaan **Error! Reference source not found.**.

$$BLEU = BP \cdot exp\left(\sum_{n=1}^{N} w_n \cdot log\, p_n\right) \tag{2.6}$$

Di mana:
BP adalah *Brevity Penalty*
w_n adalah bobot untuk *n*-gram ke-n
p_n adalah *precision* dari *n*-gram ke-n
N adalah urutan tertinggi dari *n*-gram (biasanya hingga 4-gram)

Metrik ini menjadi salah satu standar untuk menilai kualitas teks yang dihasilkan model, baik dalam tugas *Single-Aspect Captioning* (SAC) maupun *Unsupervised Text Summarization* (UTS).

b. Recall-Oriented Understudy for Gisting Evaluation (ROUGE)

ROUGE adalah metode evaluasi otomatis yang mengukur kualitas ringkasan dengan membandingkan teks hasil ringkasan dengan ringkasan ideal yang dibuat manusia. Metode ini mengukur kesamaan melalui *n*-gram, urutan kata, dan pasangan kata yang tumpang tindih. Berbagai varian ROUGE, seperti ROUGE-N, ROUGE-L, ROUGE-W, dan ROUGE-S, dirancang untuk menangkap aspek yang berbeda dari kesamaan teks. Metrik yang digunakan untuk evaluasi model pada Tugas Akhir ini adalah ROUGE-L, yang mengacu pada subsekuensi umum terpanjang (*Longest Common Subsequence* atau LCS) antara teks referensi dan kandidat. LCS mendeteksi urutan kata yang cocok dalam ringkasan tanpa perlu urutan kata yang bersebelahan. Pendekatan ini efektif untuk menangkap struktur dan urutan kata pada tingkat kalimat (C.-Y. Lin, t.t.). Persamaan metrik ROUGE-L LCS dapat dilihat pada Persamaan 2., Persamaan 2., dan Persamaan 2..

$$R_{lcs} = \frac{LCS(X,Y)}{m} \tag{2.7}$$

$$P_{lcs} = \frac{LCS(X,Y)}{n} \tag{2.8}$$

$$F_{lcs} = \frac{(1+b^2) \cdot R_{lcs} \cdot P_{lcs}}{R_{lcs} + b^2 \cdot P_{lcs}} \tag{2.9}$$

Di mana:
LCS(X,Y) adalah panjang subsekuensi umum terpanjang antara teks referensi (*X*) dan kandidat (*Y*)
m adalah panjang teks referensi
n adalah panjang teks kandidat
b adalah faktor penyeimbang antara presisi (P_{lcs}) dan *recall* (R_{lcs})

ROUGE-L memberikan nilai 1 jika teks kandidat sepenuhnya identik dengan referensi dan 0 jika tidak ada kesamaan sama sekali. Metrik ini sangat relevan untuk model SAC dan UTS, karena dapat mengevaluasi seberapa baik model menangkap struktur dan konteks semantik dalam deskripsi yang dihasilkan. ROUGE-L juga mampu menangani kasus di mana kata-kata muncul dalam urutan berbeda tetapi masih relevan secara semantic (C.-Y. Lin, t.t.).

c. Consensus-based Image Description Evaluation (CIDEr)

CIDEr adalah metrik evaluasi otomatis yang dirancang untuk menilai deskripsi gambar berdasarkan kesamaan konsensus dengan deskripsi manusia. CIDEr mengukur kualitas deskripsi dengan membandingkan kesamaan *n*-gram antara kalimat kandidat dan kumpulan kalimat referensi manusia yang menggambarkan gambar yang sama. Kesamaan ini dihitung dengan mempertimbangkan frekuensi *n*-gram dan menyesuaikan bobot menggunakan *Term Frequency-Inverse Document Frequency* (TF-IDF). TF-IDF memberikan bobot lebih besar pada *n*-gram yang unik atau informatif, serta mengurangi bobot pada *n*-gram yang sering muncul di seluruh *dataset***.** *Term Frequency* (TF) menghitung seberapa sering sebuah *n*-gram muncul dalam deskripsi tertentu, menunjukkan relevansinya dalam konteks deskripsi tersebut. Dan *Inverse Document Frequency* (IDF) mengurangi bobot *n*-gram yang umum muncul di banyak deskripsi, sehingga memberikan perhatian lebih pada *n*-gram yang unik atau spesifik konteks (Vedantam dkk., 2014). Perhitungan bobot TF-IDF $g_k(S_{ij})$ untuk setiap *n*-gram ω_k pada CIDEr dapat dilihat pada Persamaan 2..

$$g_k(S_{ij}) = \frac{h_k(S_{ij})}{\sum_{\omega_l \in \Omega} h_l(S_{ij})} log\left(\frac{|I|}{\sum_{I_p \in I} min(1, \sum_q h_k(S_{pq}))}\right) \quad (2.10)$$

Di mana:
$h_k(S_{ij})$ adalah frekuensi kemunculan *n*-gram ω_k dalam deskripsi/teks S_{ij}
$|I|$ adalah jumlah total gambar pada *dataset*
Ω adalah kumpulan semua *n*-gram

CIDEr menghitung kesamaan antara deskripsi kandidat dengan referensi berdasarkan *n*-gram menggunakan kesamaan kosinus. Perhitungannya dapat dilihat pada Persamaan 2..

$$CIDEr_n(c_i, S_i) = \frac{1}{m}\sum_j \frac{g^n(c_i) \cdot g^n(S_{ij})}{||g^n(c_i)|| \left|\left|g^n(S_{ij})\right|\right|} \quad (2.11)$$

Di mana:
c_i adalah deskripsi/kalimat kandidat
S_i adalah kumpulan deskripsi referensi untuk gambar *i*
m adalah jumlah *n*-gram yang dihitung
$g^n(c_i)$ adalah vektor TF-IDF dari *n*-gram berdasarkan deskripsi kandidat
$||g^n(c_i)||$ besarnya vektor $g^n(c_i)$, ini juga berlaku untuk $||g^n(S_{ij})||$

Hasil dari berbagai tingkat *n*-gram (biasanya dari 1-gram hingga 4-gram) kemudian digabungkan menggunakan rata-rata berbobot menggunakan Persamaan 2..

$$CIDEr(c_i, S_i) = \sum_{n=1}^{N} w_n CIDEr_n(c_i, S_i) \quad (2.12)$$

Di mana *uniform weights* (w_n) umumnya diatur seragam ($w_n = 1/N$), dan nilai $N = 4$ sering digunakan untuk menghasilkan evaluasi yang optimal.

CIDEr menilai seberapa baik deskripsi kandidat sesuai dengan referensi untuk aspek-aspek tertentu seperti warna, komposisi, atau pencahayaan pada tiap model CNN-LSTM di modul *Single-Aspect Captioning*. CIDEr juga memastikan bahwa deskripsi hasil ringkasan tetap mempertahankan informasi yang relevan dan sesuai dengan standar manusia pada modul *Unsupervised Text Summarization*.

d. Metric for Evaluation of Translation with Explicit ORdering (METEOR)

METEOR adalah metrik yang dirancang untuk mengevaluasi kualitas hasil terjemahan mesin (*Machine Translation* atau MT) dengan tingkat korelasi tinggi terhadap penilaian manusia. Berbeda dengan metrik seperti BLEU, METEOR menggunakan pencocokan berbasis *unigram* (kata tunggal) dengan mempertimbangkan bentuk permukaan, bentuk dasar, dan sinonim kata. Proses evaluasi ini dilakukan dengan menyelaraskan *unigram* antara terjemahan mesin dan referensi manusia, menghasilkan skor berbasis presisi (*precision*), *recall*, dan penalti fragmentasi untuk mengukur keakuratan dan kelancaran urutan kata dalam terjemahan (Banerjee & Lavie, t.t.). METEOR menghitung *Fmean* sebagai kombinasi harmonik dari presisi (*precision*) dan *recall*, dengan *recall* lebih banyak diberi bobot. Perhitungan tersebut dapat dilihat pada Persamaan 2.7.

$$F_{mean} = \frac{10 \cdot P \cdot R}{9 \cdot P + R} \quad (2.7)$$

Di mana:
P adalah presisi: proporsi *unigram* yang cocok terhadap total *unigram* dalam terjemahan mesin
R adalah *recall*: proporsi unigram yang cocok terhadap total *unigram* dalam referensi

Penalti dihitung berdasarkan fragmentasi atau tingkat ketidaksesuaian urutan kata menggunakan Persamaan 2.8.

$$Penalty = 0.5 \cdot \left(\frac{\#chunks}{\#unigrams\_matched}\right)^3 \quad (2.8)$$

Kemudian, skor akhir METEOR dihitung dengan mengurangi penalti dari F_{mean}, menghasilkan Persamaan 2.9.

$$Score = F_{mean} \cdot (1 - Penalty) \quad (2.9)$$

METEOR sangat relevan untuk mengevaluasi model SAC dan UTS karena mampu menangkap korelasi semantik dan keakuratan urutan kata, yang penting dalam menghasilkan deskripsi teks berbasis estetika makanan. Hal ini memungkinkan evaluasi yang lebih menyeluruh dibandingkan metrik berbasis *n*-gram lainnya.

BAB 3
METODOLOGI

3.1 Metode yang Digunakan

Pada metode yang dirancang dalam penelitian ini, langkah awal yang diperlukan adalah pemilihan dan penyesuaian metodologi terhadap tujuan pengembangan model *Food Image Aesthetic Captioning* (*Food* IAC) untuk menghasilkan deskripsi estetika citra makanan yang lebih detail dan relevan. Metode yang akan dilakukan adalah menambahkan aspek subjek makanan dan penggunaan kamera pada model *Single-Aspect Captioning* (SAC). Gambar 3.1 menggambarkan keseluruhan alur pengembangan sistem mulai dari tahap awal pra-pemrosesan hingga menghasilkan deskripsi akhir yang merepresentasikan aspek estetika pada citra makanan secara utuh.

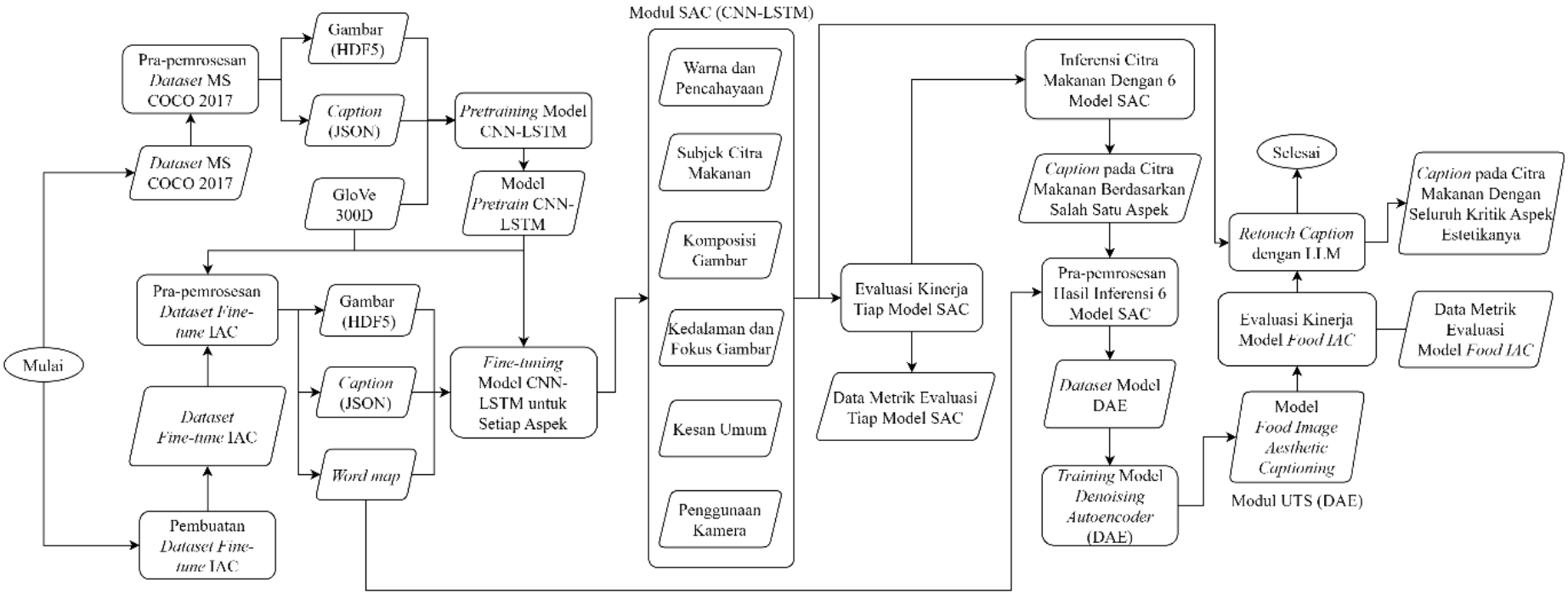
Pra-pemrosesan *dataset* MS COCO 2017 menghasilkan format citra dan *caption* yang sesuai dengan kebutuhan *training*, yaitu file citra dalam format HDF5 dan *caption* dalam format JSON. Mengingat *dataset* tersebut memiliki banyak kategori gambar, digunakan hanya citra makanan saja dalam proses ini. *Dataset* MS COCO 2017 digunakan dalam proses *pretraining* model CNN-LSTM, dengan bantuan *embedding* GloVe 300D sebagai representasi semantik kata. Hasil *pretraining* ini menjadi fondasi awal sebelum model digunakan lebih lanjut untuk tugas yang lebih spesifik.

Dataset *fine-tune* berisi citra makanan dan *caption* dengan label berdasarkan enam aspek estetika, yaitu warna dan pencahayaan, subjek citra, komposisi gambar, kedalaman dan fokus, kesan umum, serta penggunaan kamera. *Dataset* ini diproses ke dalam bentuk akhir beserta *wordmap* untuk kemudian digunakan dalam proses *fine-tuning* model CNN-LSTM secara terpisah untuk masing-masing aspek. Setiap model CNN-LSTM yang telah dilatih tersebut akan membentuk model *Single-Aspect Captioning* (SAC).

Setelah model SAC per aspek selesai dilatih, dilakukan proses evaluasi terhadap setiap model dengan menggunakan metrik seperti BLEU (1-4), ROUGE, METEOR, dan CIDEr. Selanjutnya, model-model ini digunakan untuk melakukan inferensi pada citra makanan, di mana setiap gambar akan diberikan *caption* untuk masing-masing aspek visual.

Keenam *caption* dari aspek tersebut kemudian dikumpulkan dan diproses menjadi satu *input* teks gabungan yang selanjutnya digunakan sebagai bahan pelatihan model *Denoising Autoencoder* (DAE). *Caption* hasil inferensi SAC yang mengandung *noise* (token <unk>/unknown) digunakan sebagai input *training* untuk DAE dengan target *caption* koheren yang disusun berdasarkan *caption* referensi pada *dataset*.

Setelah model DAE berhasil di-*training*, dilakukan proses inferensi untuk menghasilkan *caption* terpadu dari enam aspek. *Caption* yang dihasilkan ini masih dapat mengandung ketidaksempurnaan struktural atau semantik, sehingga pada tahap berikutnya dilakukan proses *retouch caption* menggunakan API GPT (LLM). Pada tahap ini, sistem mengirimkan *caption* hasil inferensi enam model SAC, *caption* hasil DAE, serta citra makanan yang akan diinferensi sebagai konteks ke model GPT melalui API. LLM akan menyusun ulang dan menyempurnakan *caption* sehingga hasil akhirnya menjadi lebih alami, padu, dan mencerminkan keseluruhan kritik estetika dengan baik. Model DAE dievaluasi menggunakan metrik evaluasi yang sama. Hasil evaluasi ini digunakan untuk membandingkan kualitas *caption* dari model SAC individual dan model DAE.



Gambar 3.1 Diagram Alir Penelitian

3.1.1 *Dataset* MS COCO 2017

Seperti yang ditunjukkan pada Gambar 3.1, penelitian ini menggunakan *dataset Microsoft Common Objects in Context* (MS COCO) 2017 sebagai sumber data utama untuk tahap *pretraining* model CNN-LSTM. MS COCO merupakan salah satu *dataset benchmark* dalam bidang visi komputer yang secara luas digunakan dalam berbagai penelitian *image captioning* karena menyediakan pasangan data berupa gambar dan *caption* yang kaya konteks dan bervariasi secara visual.

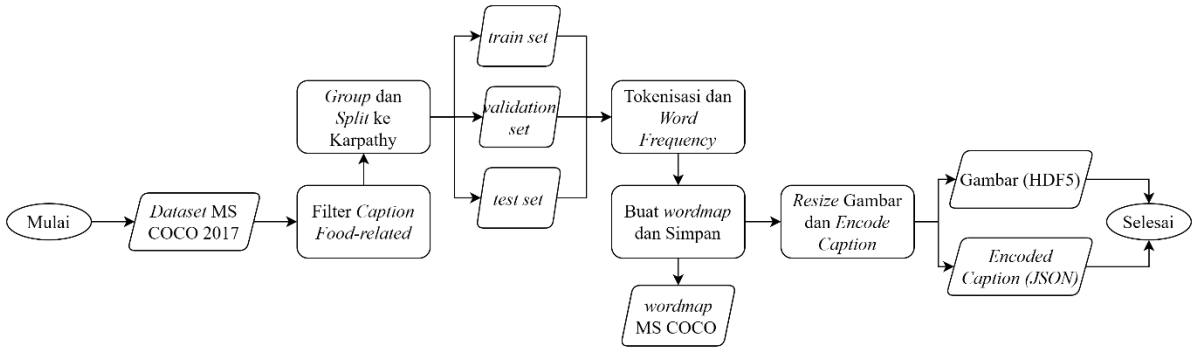
Dataset ini terdiri dari lebih dari 118.000 gambar untuk split *training* dan lebih dari 40.000 gambar untuk split *validation*, masing-masing dilengkapi dengan minimal 5 *caption* deskriptif per gambar (T.-Y. Lin dkk., 2014). *Caption-caption* ini ditulis oleh *annotator* manusia dan menggambarkan konten visual dalam gambar secara eksplisit, sehingga sangat cocok untuk melatih model yang bertugas menghubungkan representasi visual dan linguistik.

Perlu dicatat bahwa MS COCO adalah *dataset* yang bersifat *domain-general*, yang artinya mencakup berbagai kategori objek dan situasi mulai dari aktivitas manusia, hewan, kendaraan, hingga objek makanan. Karena fokus penelitian ini adalah menghasilkan *caption* yang menggambarkan estetika citra makanan, maka diperlukan proses seleksi dan penyaringan data agar hanya mencakup gambar yang relevan dengan domain makanan.

Untuk mendapatkan *subset* gambar makanan dari *dataset* MS COCO, dilakukan filtering berbasis kata kunci tematik pada *caption*. Setiap *caption* dari seluruh gambar diperiksa keberadaan kata-kata yang secara eksplisit mengindikasikan konteks makanan. Dengan pendekatan ini, dihasilkan total sebanyak 23.251 gambar yang dianggap merepresentasikan konteks makanan secara cukup kuat dan layak digunakan sebagai *data pretraining* untuk model *captioning* dalam *domain* kuliner. Gambar-gambar ini beserta *caption* yang relevan kemudian diproses lebih lanjut untuk membentuk *dataset* khusus pelatihan awal (*pretraining dataset*) pada tahap 3.1.2.

3.1.2 Pra-pemrosesan *Dataset* MS COCO 2017

Pra-pemrosesan *dataset* MS COCO 2017 dilakukan untuk mempersiapkan data citra makanan dan deskripsinya dalam format yang terstruktur agar dapat digunakan dalam pelatihan model CNN-LSTM untuk kebutuhan *pretraining* model SAC. Rangkaian alur dari tahapan ini dapat dilihat pada Gambar 3.2.



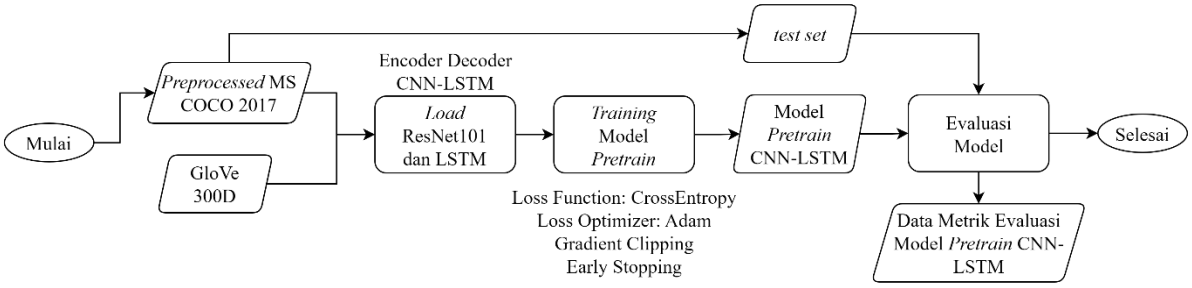
Gambar 3.2 Diagram Alir Pra-pemrosesan *Dataset* MS COCO 2017

Gambar 3.2 menunjukkan proses lengkap pra-pemrosesan *dataset* MS COCO 2017. Proses dimulai dengan memuat *raw* data dari *caption* dan *metadata* gambar MS COCO. Selanjutnya, dilakukan *filtering* terhadap *caption-caption* yang mengandung kata kunci yang relevan dengan makanan. Penyaringan ini bertujuan untuk memastikan bahwa hanya gambar dengan konteks

makanan yang digunakan dalam pelatihan model. Gambar yang telah difilter kemudian dikelompokkan dan dibagi menjadi tiga *subset* berdasarkan format Karpathy split; *train set*, *validation set*, dan *test set*, masing-masing dibagi dengan rasio 80%, 10%, dan 10% secara berurutan. Seluruh caption yang tersisa diolah melalui proses tokenisasi dan dihitung frekuensi katanya untuk membangun *wordmap*. *Wordmap* ini menyusun daftar kosakata yang digunakan dalam pelatihan dan menetapkan indeks unik untuk setiap kata, termasuk token khusus seperti <start>, <end>, <unk>, dan <pad>. *Caption* yang telah ditokenisasi kemudian dikonversi ke dalam format numerik sesuai dengan *wordmap*. Di sisi lain, gambar-gambar pada setiap *subset* di-*resize* ke dimensi 256×256 piksel dan disimpan dalam format HDF5. Sementara itu, *caption* yang telah di-*encode* dan informasi panjang *caption* disimpan dalam format JSON. Hasil akhir dari proses ini berupa kumpulan *file* HDF5 dan JSON yang digunakan sebagai input pada proses 3.1.3.

3.1.3 *Pretraining* Model CNN-LSTM

Proses *pretraining* bertujuan untuk memberikan pemahaman awal kepada model CNN-LSTM terhadap pola umum dalam deskripsi citra makanan. Tahapan ini dilaksanakan sebelum proses *fine-tuning* per aspek estetika agar model memiliki fondasi yang kuat terhadap struktur bahasa dan representasi visual dari citra makanan secara umum. Alur *pretraining* model CNN-LSTM dapat dilihat pada Gambar 3.3.



Gambar 3.3 Diagram Alir *Pretraining* Model CNN-LSTM *Dataset* MS COCO 2017

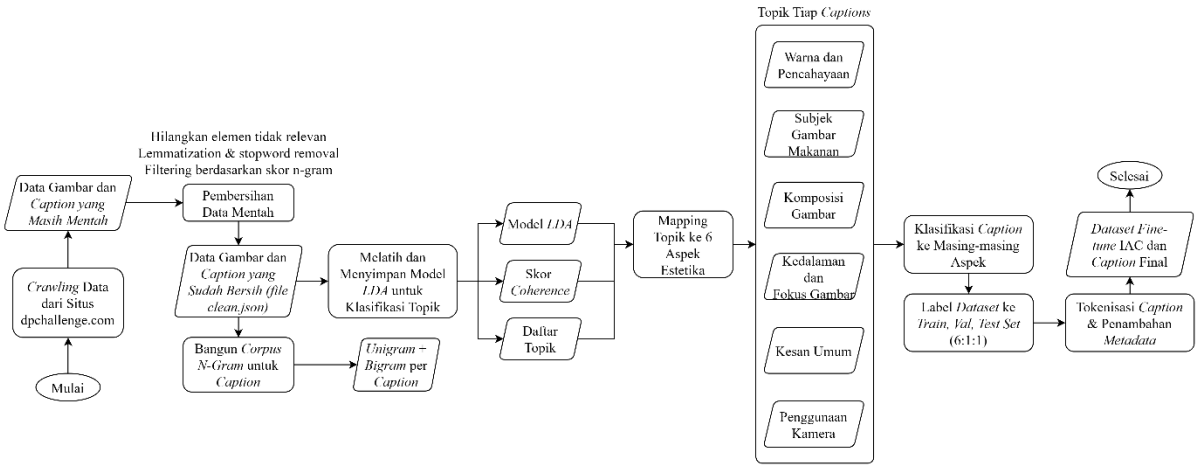
Gambar 3.3 memperlihatkan keseluruhan alur *pretraining* model CNN-LSTM. Proses dimulai dengan pemuatan *dataset* hasil pra-pemrosesan dari tahap 3.1.2. Vektor representasi kata dari GloVe 300D dimuat ke dalam *layer embedding* pada model *decoder* untuk memberikan pengetahuan semantik awal terhadap representasi kata-kata yang digunakan dalam *caption*. Model CNN-LSTM yang digunakan terdiri dari dua bagian yaitu; EncoderCNN yang berbasis arsitektur ResNet101 untuk mengekstraksi fitur visual, dan DecoderRNN (LSTM) yang berperan menyusun urutan kata dari fitur yang diekstrak.

Dataset dilatih menggunakan *loss function CrossEntropyLoss*, serta optimisasi dilakukan menggunakan algoritma Adam. Untuk menjaga stabilitas pelatihan, diterapkan teknik *gradient clipping*. Validasi dilakukan secara berkala terhadap *validation set*, dan parameter model terbaik disimpan berdasarkan nilai *loss* terendah pada data validasi. *Early stopping* diterapkan untuk menghindari *overfitting* terhadap data *training set*.

Setelah proses *training* selesai, model diuji menggunakan *subset test* dari *dataset*. *Caption* yang dihasilkan dievaluasi menggunakan metrik BLEU untuk mengukur kesesuaian antara *caption* prediksi dengan *caption* referensi yang tersedia. Evaluasi ini menjadi tolok ukur awal dari kemampuan model sebelum dilakukan penyesuaian lanjutan melalui proses *fine-tuning* per aspek estetika.

3.1.4 Pembuatan *Dataset Fine-tune* IAC

Pada penelitian ini dilakukan pembuatan *dataset*, karena tidak adanya *dataset* spesifik yang memuat data *caption*/deskripsi aspek estetika pada citra makanan. *Dataset* ini akan memuat sekitar 7.100 data citra makanan dan 46.000 *caption* bersih yang diolah dari galeri makanan dan minuman pada platform dpchallenge.com. Alur pembuatan *dataset* dapat dilihat pada Gambar 3.4.



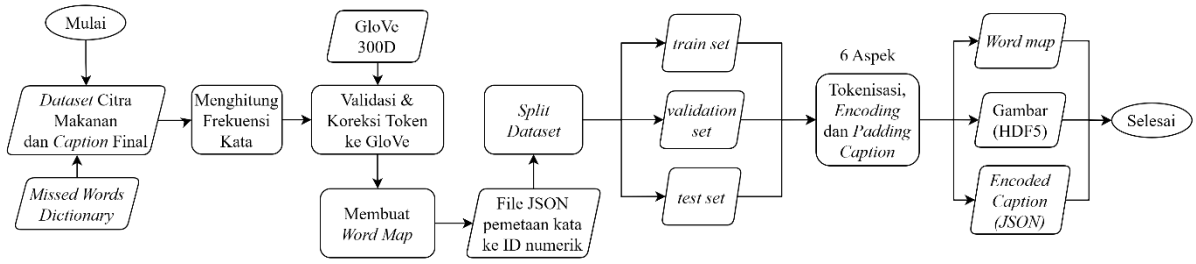
Gambar 3.4 Diagram Alir Pembuatan *Dataset* Citra Makanan

Gambar 3.4 menggambarkan alur pembuatan *dataset fine-tune* yang dimulai dari proses *crawling* data mentah untuk mengumpulkan gambar dan *caption* berupa komentar dari pengguna situs dpchallenge.com. *Caption* yang diperoleh masih bersifat mentah, sehingga perlu melalui proses pembersihan untuk menghilangkan komentar tidak relevan, *noise* linguistik, maupun kata-kata yang bersifat non-estetis. *Corpus* untuk setiap *caption* dibangun dalam bentuk representasi *n*-gram berupa *unigram* dan *bigram*, yang kemudian digunakan sebagai dasar pelatihan model topik menggunakan pendekatan *Latent Dirichlet Allocation* (LDA). Model LDA ini dilatih untuk mengidentifikasi topik-topik dominan pada *caption*, dan selanjutnya dievaluasi menggunakan *coherence score* untuk memilih model dengan kualitas pemetaan topik terbaik.

Topik-topik yang dihasilkan dari LDA kemudian di-*mapping* secara manual ke dalam enam aspek estetika yang disebutkan pada subbab 3.1. Mapping ini menjadi acuan dalam proses pembagian *caption* ke masing-masing aspek. Setelah semua *caption* dipetakan ke dalam enam aspek estetika, *dataset* yang dihasilkan selanjutnya dibagi ke dalam tiga *subset* yaitu *train, validation, dan test set* dengan rasio 6:1:1. Selama proses ini, *caption* juga ditokenisasi dan dilengkapi dengan *metadata* seperti informasi aspek dan pembagian *split dataset*. Hasil akhir dari keseluruhan proses ini adalah *dataset Fine-tune* IAC.

3.1.5 Pra-pemrosesan *Dataset Fine-tune* IAC

Pra-pemrosesan dilakukan untuk mempersiapkan *dataset* citra makanan dan *caption* dalam format yang terstruktur sehingga dapat digunakan secara langsung dalam pelatihan model CNN-LSTM pada model SAC. Proses ini bertujuan untuk memastikan bahwa setiap *caption* memiliki struktur token yang seragam, dapat dipetakan ke dalam ID numerik yang konsisten, serta memiliki representasi citra yang homogen baik dari segi resolusi maupun format penyimpanan. Alur pra-pemrosesan ini ditunjukkan pada Gambar 3.5.



Gambar 3.5 Diagram Alir Pra-pemrosesan *Dataset* Gambar dan *Caption*

Gambar 3.5 menggambarkan proses yang dimulai dari pemuatan *dataset* citra dan *caption* final hasil pemetaan ke enam aspek estetika. Langkah pertama adalah menghitung frekuensi kata dari seluruh token *caption* untuk membentuk sebuah *word frequency dictionary*. Token-token yang frekuensinya melebihi ambang batas minimal akan dipertahankan, sementara token yang tidak ditemukan dalam *embedding* GloVe 300D akan dicoba dikoreksi menggunakan pustaka *pyspellchecker* dan kamus koreksi manual (*missed words dictionary*). Kata-kata hasil seleksi kemudian digunakan untuk membentuk *word map*, yaitu pemetaan setiap token ke ID numerik, dilengkapi dengan token khusus seperti <pad>, <start>, <end>, dan <unk>. *Dataset* kemudian dibagi menjadi tiga *subset* berdasarkan label split yang telah ditentukan sebelumnya, yakni train, validation, dan test dengan rasio 6:1:1.

Setiap *caption* pada *subset* tersebut kemudian ditokenisasi dan di-*encode* menggunakan *word map*. Untuk menjaga panjang urutan tetap seragam, padding ditambahkan hingga mencapai panjang maksimum yang ditentukan. Citra makanan juga diubah ukurannya menjadi 256×256 piksel, dikonversi ke dalam tensor *array* (3, 256, 256), dan disimpan dalam *file* berformat HDF5. Sementara itu, caption yang telah di-*encode* dan panjang urutannya disimpan dalam *file* JSON. Proses ini diulang untuk setiap aspek estetika, namun *word map* yang digunakan disatukan dari keseluruhan *dataset* (*wordmap_all.json*) agar konsistensi representasi token tetap terjaga selama pelatihan model. *Dataset* yang telah diproses inilah yang akan digunakan pada tahapan pelatihan model CNN-LSTM dalam model SAC, sebagaimana akan dibahas pada subbab 3.1.6.

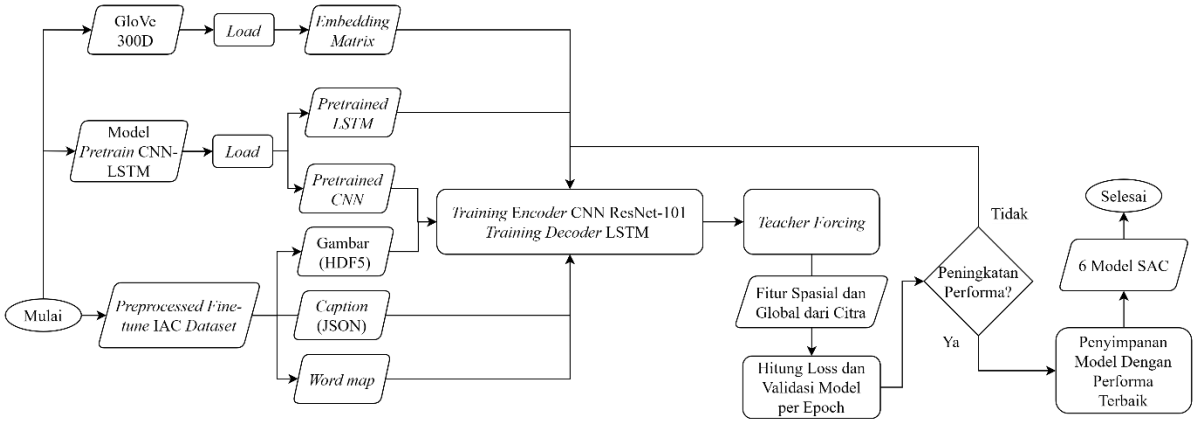
3.1.6 *Fine-tuning* Model CNN-LSTM untuk Setiap Aspek

Model pembangkit deskripsi citra makanan berdasarkan aspek estetika (Single-Aspect Captioning/SAC) bertujuan untuk menghasilkan *caption* deskriptif dari satu aspek visual tertentu. Setiap model SAC dirancang untuk menangani satu dari enam aspek estetika, yaitu warna dan pencahayaan (*color and lighting*), subjek gambar makanan (*subject*), komposisi (*composition*), kedalaman dan fokus gambar (*depth of field and focus*), kesan umum (*general impression*), dan penggunaan kamera (*use of camera*). Proses pelatihan model CNN-LSTM untuk masing-masing aspek tersebut dilakukan secara terpisah dan mandiri.

Gambar 3.6 menjelaskan proses pelatihan model CNN-LSTM untuk masing-masing aspek estetika. Proses dimulai dari pemuatan *dataset* hasil pra-pemrosesan yang telah disusun untuk setiap aspek pada tahap 3.1.5. *Dataset* ini terdiri dari gambar beresolusi seragam (256×256 piksel) dalam format tensor serta *caption* yang telah di-tokenisasi, di-*encode*, dan diproses dengan *padding*.

Model CNN-LSTM terdiri dari dua bagian utama yaitu *Encoder* dan *Decoder*. Bagian *encoder* menggunakan arsitektur ResNet-101 yang akan dimodifikasi dengan menghilangkan lapisan *fully-connected* dan menggantinya dengan *adaptive average pooling* untuk menghasilkan fitur spasial dari citra input. Bobot dari ResNet-101 dimuat dari versi *pretrained*

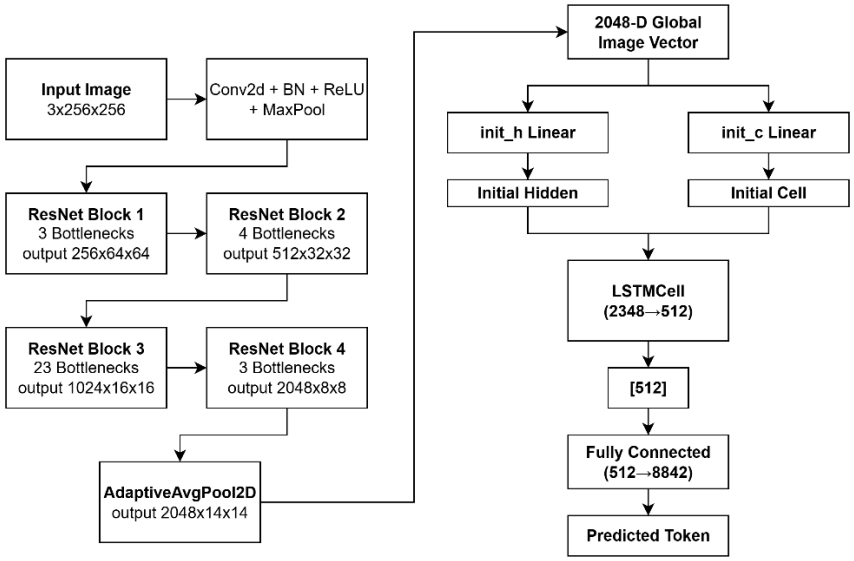
dan hanya beberapa *layer* akhir yang di-*fine-tune* selama pelatihan. *Decoder* adalah sebuah LSTMCell yang menerima *input* dari hasil *embedding* token *caption* dan fitur gambar yang dirata-rata secara spasial (*mean pooling*). *Embedding* token dimuat dari hasil pelatihan *embedding* GloVe 300D dan sebagian besar ditransfer dari model *pretraining*. Hanya *layer* akhir *decoder* seperti LSTM dan *fully connected layer* yang diperbarui selama *fine-tuning*.



Gambar 3.6 Diagram Alir Pembangunan Model SAC

Proses pelatihan dilakukan menggunakan pendekatan *teacher forcing*, di mana *caption* asli digunakan sebagai target urutan untuk mempercepat konvergensi. Fungsi loss yang digunakan adalah CrossEntropyLoss dan optimasi dilakukan menggunakan algoritma Adam. Untuk menjaga stabilitas digunakan teknik *gradient clipping*.

Evaluasi model dilakukan setiap *epoch* terhadap data validasi menggunakan nilai *loss*. Jika model menunjukkan peningkatan, parameter model disimpan dalam *checkpoint*. Namun jika tidak menunjukkan perbaikan dalam tiga *epoch* berturut-turut, proses pelatihan untuk aspek tersebut dihentikan secara otomatis (*early stopping*). Hasil akhirnya adalah enam model CNN-LSTM terpisah yang masing-masing mampu menghasilkan deskripsi teks otomatis berdasarkan aspek visual tertentu dari sebuah citra makanan.



Gambar 3.7 Arsitektur CNN-LSTM yang Digunakan

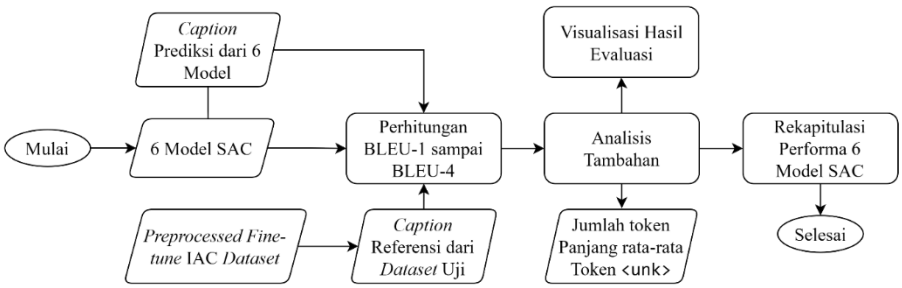
Arsitektur model yang digunakan dalam Tugas Akhir ini akan menggabungkan dua komponen utama, yaitu CNN sebagai *encoder* dan LSTM sebagai *decoder*. Model *encoder* menggunakan jaringan ResNet-101 yang telah terlatih sebelumnya untuk mengekstraksi fitur visual dari citra input berukuran 3×256×256 piksel. Proses ekstraksi dimulai dari serangkaian lapisan konvolusi awal yang diikuti oleh empat blok residual utama, masing-masing terdiri dari *bottleneck blocks*. Blok-blok ini secara bertahap mereduksi dimensi spasial dan meningkatkan kedalaman representasi fitur. Blok terakhir dari ResNet menghasilkan tensor fitur berukuran 2048×8×8, yang kemudian diubah menjadi representasi spasial berukuran 2048×14×14 melalui operasi *adaptive average pooling*. Vektor global berdimensi 2048 yang dihasilkan dari proses ini digunakan sebagai representasi semantik visual dari gambar.

Komponen *decoder* dirancang untuk menghasilkan *caption* berbasis teks dengan menggunakan jaringan LSTM. Sebelum memulai proses *decoding*, vektor global dari *encoder* terlebih dahulu ditransformasikan melalui dua *layer* linear untuk menghasilkan inisialisasi *hidden state* dan *cell state* dari LSTM. Pada setiap langkah *decoding*, LSTM menerima input berupa konkatenasi antara *embedding* token kata sebelumnya (300 dimensi) dan vektor global citra (2048 dimensi), menghasilkan vektor gabungan berdimensi 2348. *Output* dari LSTM berupa *hidden state* berukuran 512 akan diteruskan ke lapisan *fully-connected* yang bertugas mengklasifikasikan token keluaran ke dalam salah satu dari sekian kosakata yang tersedia. Model ini diimplementasikan dalam bentuk step-wise *decoder* tanpa *attention*, di mana token *caption* dihasilkan secara berurutan hingga mencapai token akhir atau batas panjang maksimum.

Secara keseluruhan, integrasi antara CNN dan LSTM dalam arsitektur ini bertujuan untuk memanfaatkan kekuatan CNN dalam memahami representasi spasial dari citra, serta kapabilitas LSTM dalam memodelkan urutan kata yang logis dan terstruktur. Gambar arsitektur model secara lengkap dapat dilihat pada Gambar 3.7 yang menunjukkan alur proses dari input citra hingga keluaran token *caption* melalui blok-blok jaringan yang telah dirancang. Lampiran 3 menjelaskan detail CNN dan LSTM yang digunakan.

3.1.7 Evaluasi Kinerja Tiap Model SAC

Evaluasi dilakukan untuk mengukur seberapa baik model SAC dalam menghasilkan *caption* yang relevan dan sesuai terhadap *caption* referensi pada setiap aspek estetika. Evaluasi dilakukan terhadap hasil prediksi *caption* pada *test set* dari masing-masing model, dan menggunakan metrik *Bilingual Evaluation Understudy* (BLEU) untuk mengukur kesamaan *n*-gram antara *caption* prediksi dengan *caption* referensi. Diagram alur proses evaluasi kinerja model SAC dapat dilihat pada Gambar 3.8.



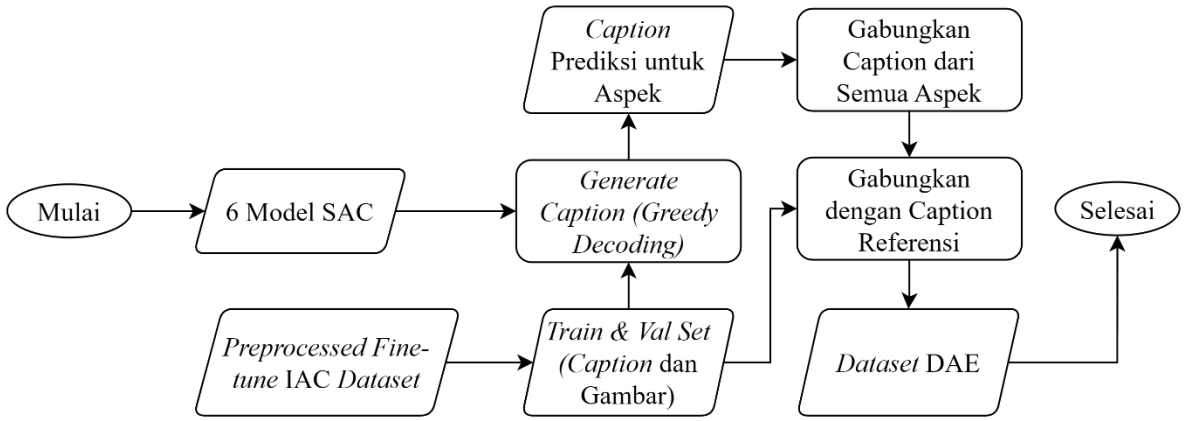
Gambar 3.8 Diagram Alir Evaluasi Model SAC

Gambar 3.8 memperlihatkan alur evaluasi kinerja model. *Caption* hasil inferensi dari model akan dibandingkan dengan *caption* referensi menggunakan BLEU-1 sampai BLEU-4. Selain

BLEU, evaluasi juga mencakup analisis distribusi kata dan panjang caption, serta identifikasi token yang muncul sebagai indikator kekurangan pada representasi *embedding* atau *vocabulary*. Hasil evaluasi akan digunakan untuk menentukan kualitas masing-masing model dan efektivitas proses *fine-tuning* terhadap setiap aspek estetika.

3.1.8 Inferensi Citra Makanan Dengan 6 Model SAC

Setelah keenam model SAC selesai dilatih dan disimpan dalam bentuk *checkpoint* terbaik, proses inferensi dilakukan untuk menghasilkan deskripsi otomatis dari citra makanan berdasarkan masing-masing aspek estetika. Proses ini tidak hanya bertujuan untuk menguji kemampuan generalisasi model terhadap data uji, namun juga berfungsi sebagai tahap dalam pembuatan *dataset* untuk pelatihan model DAE.



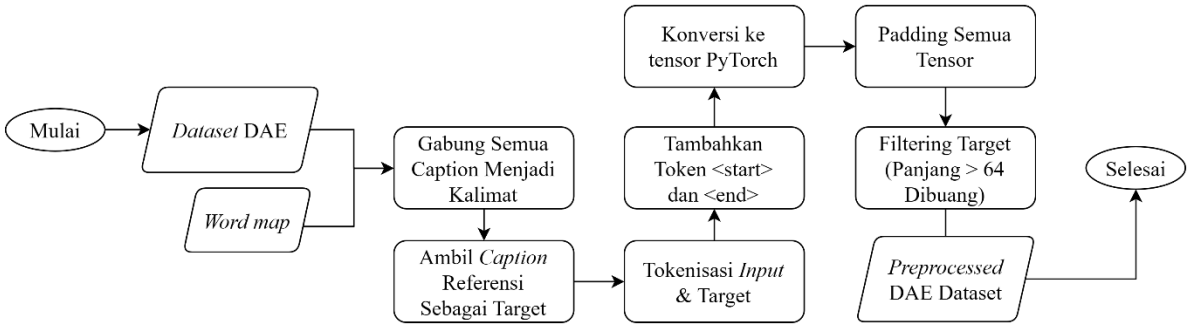
Gambar 3.9 Diagram Alir Inferensi dan Pembuatan *Dataset* DAE

Gambar 3.9 memperlihatkan alur inferensi dan pembuatan *dataset* DAE. Inferensi dilakukan dengan memuat model SAC per aspek yang terdiri dari pasangan *encoder* CNN berbasis ResNet-101 dan *decoder* LSTM, masing-masing diambil dari *checkpoint* pelatihan terbaik. Untuk setiap gambar pada *dataset* akhir, gambar dibaca dan dipra-proses ke bentuk *tensor*, kemudian dilewatkan melalui *encoder* untuk mengekstrak fitur spasial. *Decoder* akan menghasilkan *caption* berdasarkan aspek terkait menggunakan metode *greedy decoding* secara autoregresif, dimulai dari token <start> dan berhenti pada token <end>.

Setiap gambar akan diproses secara berulang sebanyak enam kali, masing-masing dengan model berbeda untuk aspek warna dan pencahayaan, subjek makanan, komposisi gambar, kedalaman dan fokus, kesan umum, dan penggunaan kamera. Hasil *caption* dari keenam model akan digabungkan dan disimpan dalam *file* terstruktur JSON. Setiap entri dalam *file* tersebut mencakup nama *file* gambar, *caption* referensi dari pengguna, serta *caption* hasil prediksi dari setiap aspek. *Dataset* ini kemudian digunakan dalam pelatihan model DAE melalui proses pra-pemrosesan terlebih dahulu yang dibahas pada subbab 3.1.9.

3.1.9 Pra-pemrosesan Hasil Inferensi 6 Model SAC

Setelah proses inferensi menghasilkan *caption* prediksi dari enam model SAC untuk setiap gambar dalam *dataset*, langkah berikutnya adalah melakukan pra-pemrosesan terhadap data tersebut agar dapat digunakan sebagai *input* dan target pada proses pelatihan model DAE. Tujuan utama dari tahap ini adalah mengubah kumpulan *caption* prediksi dan referensi menjadi representasi *tensor* yang terstruktur dan dapat digunakan langsung dalam pelatihan model sekuens. Alur pra-pemrosesan hasil inferensi (*dataset* DAE) dapat dilihat pada Gambar 3.10.



Gambar 3.10 Diagram Alir Pra-pemrosesan *Dataset* DAE

Gambar 3.10 menggambarkan alur pra-pemrosesan hasil inferensi dari model SAC. Proses dimulai dengan memuat berkas hasil inferensi dalam format .json, yang berisi *caption* prediksi dari enam aspek estetika serta *caption* referensi dari *dataset* awal. *Caption* dari enam aspek tersebut digabungkan menjadi satu kalimat panjang yang menyertakan *tag* eksplisit seperti [general_impression], [subject], dan seterusnya. *Caption* referensi dipilih sebagai target yang akan dipelajari oleh model.

Proses tokenisasi dilakukan terhadap *input* gabungan dan *caption* referensi menggunakan kamus *word_map* yang telah dibentuk pada subbab 3.1.5. Token yang tidak ditemukan akan dikonversi menjadi token khusus <unk> (*unknown*). Tokenisasi menghasilkan urutan ID numerik yang dilengkapi dengan token pembuka <start> dan penutup <end>.

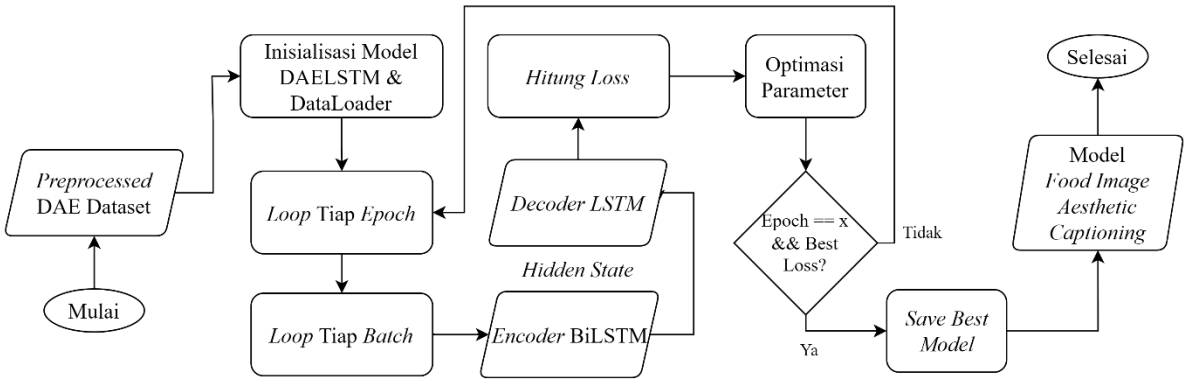
Setelah tokenisasi, seluruh pasangan *input* dan target diubah menjadi *tensor* PyTorch dan disamakan panjangnya dengan *padding* menggunakan token <pad>. Untuk menjaga kualitas data dan efisiensi pelatihan, dilakukan penyaringan terhadap pasangan yang memiliki panjang *caption* target melebihi 64 token. Hanya pasangan data yang memenuhi syarat panjang inilah yang disimpan ke dalam berkas hasil pra-proses dalam *file* berformat .pt sebagai hasil akhir pra-pemrosesan. File tersebut akan memuat tiga komponen utama, yaitu tensor *input*, tensor target, dan *word map* yang akan digunakan dalam pelatihan DAE pada subbab 3.1.10.

3.1.10 *Training* Model *Denoising Autoencoder* (DAE)

Model *Denoising Autoencode*r (DAE) dilatih untuk menghasilkan *caption* estetika yang terintegrasi dengan menggabungkan enam aspek *caption* prediksi menjadi satu representasi yang lebih utuh dan estetis. Model DAE yang digunakan pada penelitian ini merupakan model berbasis LSTM dengan arsitektur *encoder-decoder*, di mana *encoder* menerima input *caption* gabungan dari enam aspek dan *decoder* menghasilkan caption target referensi.

Gambar 3.11 memperlihatkan ilustrasi pemanduan *caption* memuat alur pelatihan model DAE. Proses pelatihan dimulai dengan memuat dataset hasil preprocessing pada subbab 3.1.9, yang berisi tensor *input* dan target hasil tokenisasi dan *padding* dari tahap sebelumnya. *Dataset* ini dikemas menggunakan DataLoader dan dilatih dengan *loss* function CrossEntropyLoss yang mengabaikan token <pad>.

Model DAE terdiri dari tiga bagian utama, yaitu layer *embedding*, encoder BiLSTM untuk memproyeksikan input ke representasi laten, dan decoder LSTM *unidirectional* yang menghasilkan urutan *caption*. Decoder mengambil state akhir dari *encoder* sebagai kondisi awal dan menghasilkan *caption* secara autoregresif.



Gambar 3.11 Diagram Alir Pemanduan *Caption* dengan DAE

Pelatihan dilakukan selama beberapa *epoch* menggunakan optimasi Adam. Model terbaik disimpan berdasarkan nilai *loss* terendah selama pelatihan. Setelah pelatihan selesai, model dapat digunakan untuk menghasilkan *caption* baru dari input yang belum pernah dilihat sebelumnya, dengan proses *decoding* autoregresif dimulai dari token <start> hingga mencapai token <end> atau panjang maksimum.

Model yang telah dilatih akan digunakan untuk menghasilkan deskripsi estetika yang lebih konsisten dan bermakna dalam tahap evaluasi akhir sistem *captioning*.

3.1.11 Evaluasi Kinerja Model *Food IAC*

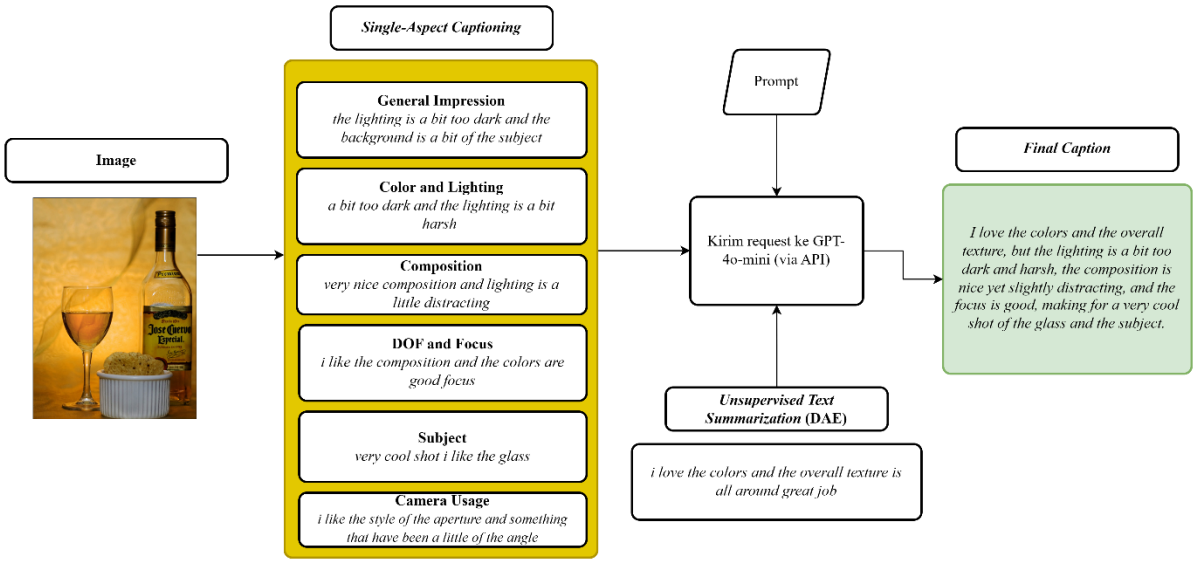
Setelah model DAE berhasil dilatih, langkah selanjutnya adalah mengevaluasi kualitas *caption* yang dihasilkan oleh model tersebut. Evaluasi ini bertujuan untuk mengukur seberapa baik *caption* yang dihasilkan dapat merepresentasikan deskripsi estetika makanan dibandingkan dengan referensi yang tersedia dalam *dataset ground truth*.

Evaluasi dilakukan menggunakan berbagai metrik yang umum digunakan dalam tugas *image captioning*, yaitu BLEU (1 sampai 4), ROUGE-L, METEOR, CIDEr. Proses evaluasi diawali dengan memuat data referensi dari *file* all.json dan hasil prediksi *caption* dari model DAE yang disimpan dalam dae_lstm_outputs.json. Setiap *caption* kemudian dipasangkan berdasarkan nama file gambar, sehingga hasil dan referensi dapat dibandingkan secara langsung. Seluruh *caption* kemudian ditokenisasi menggunakan *tokenizer* Penn Treebank dari pustaka pycocoevalcap agar struktur token antar *caption* seragam dan dapat dibandingkan secara adil. Evaluasi metrik dilakukan satu per satu terhadap pasangan *caption* hasil dan *caption* referensi.

Hasil evaluasi ini menjadi landasan dalam menilai apakah model DAE sudah mampu mengintegrasikan aspek-aspek estetika menjadi satu deskripsi terpadu secara natural dan bermakna. Selain itu, metrik seperti METEOR juga memberikan indikasi terhadap kualitas semantik dan koherensi antar elemen dalam caption hasil.

3.1.12 *Retouch Caption* dengan LLM dan Evaluasi

Setelah menghasilkan *caption* terpadu dari model DAE, tahap selanjutnya adalah melakukan *retouch* terhadap *caption* tersebut menggunakan *Large Language Model* (LLM), dalam hal ini GPT-4o-mini. Tujuan dari proses *retouch* ini adalah untuk memperhalus struktur kalimat, memperbaiki kesalahan token seperti <unk>, serta memastikan bahwa seluruh aspek estetika terwakili secara eksplisit dan alami dalam satu kalimat terpadu tanpa kehilangan makna aslinya. Ilustrasi proses *retouch* dapat dilihat pada Gambar 3.12.



Gambar 3.12 Ilustrasi *Retouch Caption* Dengan LLM

Proses *retouch* diawali dengan menggabungkan enam *caption* dari model CNN-LSTM berdasarkan masing-masing aspek ke dalam satu string input, yang kemudian dikodekan menggunakan *word map* dan digunakan sebagai masukan ke model DAE. Hasil *output* dari DAE kemudian dikirimkan ke GPT-4o-mini beserta *input* aspek gabungan dalam bentuk *prompt* instruksi.LLM akan menerima informasi teks tersebut beserta gambar dan mengembalikan *caption* versi *retouch* yang telah disesuaikan secara linguistik dan semantik, tanpa menambahkan informasi baru yang tidak ada dalam *caption* asli. *Prompt* yang digunakan bersifat terstruktur dan ketat, bertujuan untuk membatasi kreativitas GPT agar tetap fokus pada penyempurnaan, bukan imajinasi bebas. *Caption* hasil *retouch* ini kemudian disimpan sebagai *output* akhir untuk ditampilkan kepada *end user*.

Evaluasi terhadap *caption* hasil *retouch* dilakukan dengan membandingkan *caption* yang telah diperhalus menggunakan model GPT dengan *ground truth caption* dari *subset test dataset*, menggunakan metrik evaluasi teks yang umum pada tugas *image captioning*. Proses ini dimulai dengan menyamakan nama file gambar antara *caption* hasil *retouch* dan anotasi *ground truth*, lalu dilakukan tokenisasi terhadap kedua set kalimat menggunakan tokenizer standar Penn Treebank (PTBTokenizer). Setelah itu, *caption* hasil *retouch* dinilai dengan lima metrik utama, yaitu BLEU (1-gram hingga 4-gram), ROUGE-L, METEOR, dan CIDEr, yang masing-masing mengukur aspek berbeda dari kemiripan semantik, leksikal, dan struktural antara *caption* prediksi dan referensi. Implementasi metrik dilakukan menggunakan library pycocoevalcap, yang telah banyak digunakan dalam kompetisi dan *benchmark captioning*. Nilai-nilai metrik tersebut kemudian dianalisis untuk menilai sejauh mana *caption* hasil *retouch* berhasil menyampaikan informasi yang sesuai dengan *ground truth* dan meningkatkan keterbacaan dibandingkan *caption* DAE *original*.

3.2 Bahan dan Peralatan yang Digunakan

Berikut adalah peralatan pendukung yang akan digunakan untuk mendukung pengerjaan Tugas Akhir ini:

3.2.1 Perangkat Keras

Beberapa perangkat keras yang menunjang pengerjaan Tugas Akhir ini dapat dilihat pada Tabel 3.1.

Tabel 3.1 Perangkat Keras Pendukung

Atribut	Detil Spesifikasi
Komputer	Acer Swift 3 SF314-54G
Processor/CPU	Intel Core 15-8250U
GPU	NVIDIA GeForce MX150
Memori/RAM	8 GB DDR4-2400 MHz
Disk	SSD 512 GB

3.2.2 Perangkat Lunak

Beberapa perangkat lunak yang menunjang pengerjaan Tugas Akhir ini dapat dilihat pada Tabel 3.2.

Tabel 3.2 Perangkat Lunak Pendukung

Jenis	Perangkat Lunak
Sistem Operasi	Windows 11 Home Single Language 64-bit (10.0, Build 22631)
Bahasa Pemrograman	Python 3.10.12 Beserta Library Pendukung Penelitian
IDE	Visual Studio Code
Notebook	Kaggle

3.3 Rencana Implementasi dan Uji Coba

Pada Tugas Akhir ini seluruh proses akan dilakukan menggunakan perangkat lunak pendukung yang disebutkan pada Tabel 3.2.

3.3.1 Implementasi Praproses *Dataset* MS COCO 2017

Sebelum proses pelatihan model dilakukan, *dataset* MS COCO 2017 perlu dipersiapkan agar sesuai dengan kebutuhan *pretraining* CNN-LSTM. *Dataset* ini menyediakan ribuan citra dan lima *caption* per citra yang bersifat umum. Namun, karena fokus penelitian ini adalah citra makanan, hanya *subset* data yang relevan dengan topik tersebut yang diambil melalui proses penyaringan berbasis kata kunci. Selain itu, dilakukan pula proses normalisasi teks, tokenisasi, pembuatan *word map*, serta *encoding caption* menjadi representasi numerik.

Langkah-langkah praproses ini bertujuan untuk menyusun *dataset* makanan dari COCO dalam format terstruktur agar dapat digunakan secara langsung oleh model CNN-LSTM. File hasil praproses disimpan dalam format `.hdf5` untuk citra dan `.json` untuk *metadata* dan *word map*, mengikuti praktik standar dalam *image captioning*.

```
INPUT: Folder MS COCO (train2017, val2017), file annotation caption
OUTPUT: File preprocessed_dataset.hdf5, captions.json, dan word_map.json

1: BEGIN
2: raw_captions ← LOAD_JSON("annotations/captions_train2017.json")
3: filtered_data ← []
4: FOR each image_id, captions IN raw_captions DO
5:       IF CONTAINS_KEYWORD(captions, ["food", "meal", "dish", "plate"])
THEN
6:          filtered_data.APPEND((image_id, captions))
```

```
7: END FOR
8:
9: word_freq ← COUNT_WORDS(filtered_data)
10: word_map ← CREATE_WORD_MAP(word_freq, min_freq=5)
11:
12: processed_data ← []
13: FOR each image_id, captions IN filtered_data DO
14:     encoded_caps ← []
15:     FOR each cap IN captions DO
16:         IF LENGTH(cap) ≤ 50 THEN
17:             encoded ← ENCODE(cap, word_map)
18:             encoded_caps.APPEND(encoded)
19:         END IF
20:     END FOR
21:     processed_data.APPEND((image_id, encoded_caps))
22: END FOR
23:
24: SAVE_TO_HDF5(processed_data, "preprocessed_dataset.hdf5")
25: SAVE_JSON(word_map, "word_map.json")
26: SAVE_JSON(processed_data, "captions.json")
27: END
```

Kode Semu 3.1 Pra-pemrosesan *Dataset* MS COCO 2017

Proses diawali dengan memuat seluruh *caption* dari berkas anotasi MS COCO menggunakan `LOAD_JSON` pada baris 2. *Caption* disaring menggunakan fungsi `CONTAINS_KEYWORD` agar hanya menyertakan deskripsi yang memiliki kata kunci terkait makanan, seperti "*food*" atau "*dish*". Data hasil seleksi disimpan ke dalam `filtered_data`. Selanjutnya, dilakukan perhitungan frekuensi kata dari seluruh *caption* untuk menyusun `word_map`, yaitu kamus yang memetakan kata ke indeks numerik. Kata yang muncul kurang dari lima kali diabaikan agar mengurangi *noise* dalam pelatihan. Setelah kamus tersedia, setiap *caption* di-*encode* menggunakan `ENCODE` dan dipastikan panjangnya tidak melebihi 50 token. *Caption* yang valid disimpan dalam `encoded_caps` dan ditautkan ke ID gambar. Semua hasil akhirnya disimpan dalam tiga file output: `.hdf5` untuk menyimpan array numerik citra dan *caption*, serta dua `.json` file untuk *word map* dan *metadata caption*.

3.3.2 Implementasi *Pretraining* CNN-LSTM

Setelah *dataset* MS COCO 2017 berhasil diproses dan disimpan dalam format terstruktur, langkah selanjutnya adalah melakukan *pretraining* model CNN-LSTM untuk tugas *image captioning* umum. *Pretraining* ini bertujuan untuk memberikan pengetahuan awal kepada model dalam memahami korelasi antara fitur visual dan deskripsi tekstual, sebelum nantinya dilakukan *fine-tuning* pada domain estetika makanan. Model yang digunakan terdiri dari *encoder* CNN (ResNet-101) untuk mengekstrak fitur citra dan decoder LSTM untuk menghasilkan urutan kata. *Embedding* kata menggunakan GloVe 300 dimensi untuk mempercepat konvergensi dan meningkatkan representasi semantik. Proses pelatihan dilakukan secara *supervised* dengan perhitungan *cross-entropy loss* antara *caption* yang dihasilkan dan *caption* referensi.

```
INPUT: preprocessed_dataset.hdf5, word_map.json
OUTPUT: model_pretrained.pth, training_logs.json

1: BEGIN
2: word_map ← LOAD_JSON("word_map.json")
3: dataset ← LOAD_HDF5("preprocessed_dataset.hdf5")
4:
```

```
5: model ← INIT_CNN_LSTM(encoder="ResNet101", decoder="LSTM",
embedding_dim=300, hidden_size=512)
6: model.LOAD_GLOVE_EMBEDDINGS("glove.6B.300d.txt", word_map)
7:
8: optimizer ← INIT_ADAM(model.parameters(), lr=4e-4)
9: criterion ← INIT_CROSS_ENTROPY_LOSS()
10:
11: FOR epoch IN range(1, max_epochs + 1) DO
12:     FOR images, captions IN DATALOADER(dataset, batch_size=64) DO
13:         encoded_images ← model.encoder(images)
14:         outputs ← model.decoder(encoded_images, captions)
15:         loss ← criterion(outputs, captions)
16:
17:         optimizer.zero_grad()
18:         loss.backward()
19:         CLIP_GRAD_NORM(model.parameters(), max_norm=5.0)
20:         optimizer.step()
21:     END FOR
22:
23:     SAVE_CHECKPOINT(model, epoch, loss,
filename="model_pretrained.pth")
24:     LOG_METRICS(epoch, loss, filename="training_logs.json")
25: END FOR
26: END
```

Kode Semu 3.2 *Pretraining* Model CNN-LSTM

Langkah awal adalah memuat kembali `word_map` dan *dataset* hasil praproses dari subbab sebelumnya. Model CNN-LSTM diinisialisasi pada baris ke-5, dengan ResNet-101 sebagai *encoder* dan LSTM sebagai *decoder*. *Embedding* kata dimuat dari file GloVe 300D dan dipetakan ke setiap token dalam `word_map`. Proses pelatihan dilakukan dalam *loop epoch* (baris 11), di mana setiap batch terdiri dari gambar dan *caption*. Fitur visual diekstrak oleh *encoder*, lalu diteruskan ke *decoder* untuk menghasilkan urutan prediksi. Prediksi ini dibandingkan dengan *caption ground-truth* menggunakan fungsi *loss CrossEntropyLoss*. Setelah menghitung *loss*, dilakukan proses *backpropagation*, normalisasi *gradien* (`CLIP_GRAD_NORM`) untuk mencegah *exploding gradients*, dan optimisasi parameter. Model disimpan setiap *epoch* ke file `.pth` sebagai checkpoint, dan *loss* dicatat dalam *file log* untuk keperluan visualisasi performa pelatihan.

3.3.3 Implementasi Pembuatan *Dataset Fine-tune* IAC

Langkah awal dalam membangun *dataset fine-tune* IAC adalah melakukan pembersihan data komentar dari *file* `raw.json`, yang memuat komentar-komentar estetika dari pengguna terhadap citra makanan. Komentar-komentar ini bersifat bebas dan tidak terstruktur, sehingga proses pembersihan dilakukan untuk memastikan bahwa hanya komentar yang informatif, relevan, dan bersih secara linguistik yang disertakan dalam *dataset* akhir. Tujuan utama dari tahap ini adalah menyaring komentar yang mengandung n-gram bermakna, tidak repetitif, dan tidak mengandung kata yang tidak relevan atau bersifat *noise*.

Proses pembersihan dimulai dengan memuat file `raw.json` yang berisi *metadata* dan komentar untuk setiap gambar. *Tokenizer* berbasis ekspresi reguler dan *lemmatizer* WordNet diinisialisasi, bersama dengan daftar *stopword* berbahasa Inggris. Setiap komentar pada gambar diproses dengan beberapa tahapan yaitu normalisasi teks dasar, pengurangan karakter berulang, tokenisasi, *part-of-speech tagging* (POS), dan *lemmatization*. Setelah itu, dilakukan pengecekan terhadap kehadiran kata-kata terlarang seperti "*congrats*" atau kata asing, serta diperiksa apakah *caption* tersebut memiliki *n*-gram yang relevan untuk digunakan. *N*-gram ini

terbagi menjadi *unigram* (kata benda dominan) dan *bigram* (gabungan kata benda dengan kata sifat atau keterangan).

Caption yang tidak lolos proses validasi tersebut akan dicatat alasannya, seperti terlalu repetitif, tidak memiliki *n*-gram, mengandung kata tidak diinginkan, atau terlalu pendek. *Caption* yang lolos disimpan sementara dalam daftar yang kemudian akan digunakan untuk menghitung skor *informativeness* berdasarkan distribusi frekuensi *n*-gram global. Setiap *caption* yang tersisa akan diberikan skor *informativeness* dan hanya *caption* dengan skor di atas ambang batas yang disimpan ke dalam *dataset* akhir. *Caption* dengan panjang di bawah lima token juga disaring. *Dataset* hasil pembersihan disimpan dalam *file* `clean.json`.

```
INPUT: raw.json
OUTPUT: clean.json

BEGIN
raw_data ← LOAD_JSON("raw.json")
tokenizer ← INIT_REGEXP_TOKENIZER()
lemmatizer ← INIT_WORDNET_LEMMATIZER()
stopwords ← LOAD_STOPWORDS("english")

INIT unigram_dict, bigram_dict, reason_counter
INIT image_list ← []

FOR image_id, metadata IN raw_data DO
    image ← INIT_IMAGE_DICT(image_id, metadata["image_url"])

    FOR comment IN metadata["comments"] DO
        clean ← BASIC_CLEAN(comment)
        reduced ← REDUCE_REPEAT_CHAR(clean)
        tokens ← TOKENIZE(reduced)
        pos_tags ← POS_TAG(tokens)
        lemmatized ← LEMMATIZE(pos_tags)

        IF CONTAINS_BAD_WORD(lemmatized) THEN
            reason_counter["bad_word"]++
            CONTINUE
        END IF

        unigrams, bigrams ← EXTRACT_NGRAMS(lemmatized, stopwords)
        IF unigrams IS_EMPTY OR bigrams IS_EMPTY THEN
            reason_counter["no_ngram"]++
            CONTINUE
        END IF

        token_counts ← COUNT_TOKENS(tokens)
        IF MAX_FREQ(token_counts) / LEN(tokens) > 0.2 THEN
            reason_counter["repetitive_token"]++
            CONTINUE
        END IF

        caption ← FORMATTED_CAPTION(reduced, tokens, unigrams,
bigrams)
        APPEND caption TO image["sentences"]
    END FOR

    IF image["sentences"] NOT EMPTY THEN
        APPEND image TO image_list
    END IF
END FOR

```

```
46: COMPUTE unigram_score_list FROM unigram_dict
47: COMPUTE bigram_score_list FROM bigram_dict
48:
49: clean_data ← {"images": []}
50: FOR image IN image_list DO
51:     valid_captions ← []
52:     FOR caption IN image["sentences"] DO
53:         score ← COMPUTE_TFIDF_SCORE(caption, unigram_score_list,
bigram_score_list)
54:         IF score < 10 THEN
55:             reason_counter["too_generic"]++
56:             CONTINUE
57:         END IF
58:
59:         IF LEN(caption["tokens"]) < 5 THEN
60:             reason_counter["too_short"]++
61:             CONTINUE
62:         END IF
63:
64:         APPEND caption TO valid_captions
65:     END FOR
66:
67:     IF valid_captions NOT EMPTY THEN
68:         image["sentences"] ← valid_captions
69:         APPEND image TO clean_data["images"]
70:     END IF
71: END FOR
72:
73: SAVE_JSON(clean_data, "clean.json")
74: PRINT_STATS(reason_counter)
75: END
```

Kode Semu 3.3 *Cleaning Raw Captions*

Untuk mengelompokkan caption berdasarkan aspek estetika, dilakukan proses pemetaan topik menggunakan *Latent Dirichlet Allocation* (LDA). Proses ini dimulai dari data hasil pembersihan `clean.json`, yang berisi *caption* yang telah di-tokenisasi, dilematisasi, dan difilter berdasarkan kualitas *n*-gram. Setiap *caption* kemudian dikonversi menjadi daftar *n*-gram (gabungan *unigram* dan *bigram*) yang dianggap merepresentasikan fitur semantik dari *caption* tersebut. Dengan pendekatan ini, *corpus* LDA dibentuk dari *n*-gram *caption* sebagai representasi dokumen.

Setelah corpus disusun, dibentuklah *dictionary* dan *document-term matrix* menggunakan pustaka gensim. Filtering dilakukan agar hanya token yang muncul lebih dari 30 dokumen dan kurang dari 10% dokumen yang disertakan, untuk menjaga kualitas fitur. Model LDA kemudian dilatih secara bertahap dengan jumlah topik bervariasi dari 6 hingga 66, dengan langkah 6. Setiap model yang dihasilkan dievaluasi menggunakan *coherence score* c_v untuk mengukur kesesuaian topik yang terbentuk terhadap koherensi semantik antar kata. Model dengan *coherence* tertinggi akan dipilih dan disimpan sebagai model LDA utama untuk proses pemetaan *caption* berdasarkan aspek.

Visualisasi model LDA terbaik dilakukan dengan pustaka pyLDAvis untuk mendapatkan gambaran distribusi topik dan dominasi kata. Topik-topik hasil LDA kemudian akan dipetakan secara manual ke enam aspek estetika yang telah didefinisikan sebelumnya, seperti *lighting, color, composition, subject, use of camera,* dan *general impression*.

```
INPUT:                                                          clean.json
OUTPUT:    lda_topics_60.model,    lda_dictionary.dict,    lda_corpus.mm,
lda_coherence_summary.txt

BEGIN
image_list ← LOAD_JSON("clean.json")["images"]
text_corpus ← []

FOR img IN image_list DO
    FOR caption IN img["sentences"] DO
        ngrams ← GET_UNIGRAMS_AND_BIGRAMS(caption)
            formatted ← FORMAT_NGRAMS(ngrams)   // contoh: ["soft",
"warm_color", "shallow_depth"]
        text_corpus.APPEND(formatted)
    END FOR
END FOR

dictionary ← CREATE_DICTIONARY(text_corpus)
dictionary.FILTER_EXTREMES(no_below=30, no_above=0.10)
doc_term_matrix ← CONVERT_TO_BOW(text_corpus, dictionary)
SAVE(dictionary, "lda_dictionary.dict")
SAVE(doc_term_matrix, "lda_corpus.mm")

best_score ← -inf; best_topics ← NULL
FOR num_topics IN RANGE(6, 66, 6) DO
        model ← TRAIN_LDA(doc_term_matrix, dictionary, num_topics,
passes=200, iterations=5000)
        score ← COMPUTE_COHERENCE(model, text_corpus, dictionary,
method="c_v")
    LOG_COHERENCE(num_topics, score, "lda_coherence_summary.txt")
    SAVE(model, f"lda_topics_{num_topics}.model")
    IF score > best_score THEN
        best_score ← score
        best_topics ← num_topics
    END IF
END FOR

best_model ← LOAD_MODEL(f"lda_topics_{best_topics}.model")
                VISUALIZE_LDA(best_model,                text_corpus,
save_path=f"lda_topics_{best_topics}_visualization.html")
END
```

Kode Semu 3.4 Implementasi LDA *Topic Modeling*

Setelah proses pelatihan LDA menghasilkan model topik terbaik, langkah berikutnya adalah mengelompokkan setiap *caption* dari `clean.json` ke dalam salah satu dari enam aspek estetika, yaitu *color_light*, *composition*, *subject*, *dof_and_focus*, *general_impression*, dan *use_of_camera*. Proses ini dilakukan dengan mengimplementasikan *topic-to-aspect mapping* yang telah ditentukan sebelumnya berdasarkan analisis topik hasil LDA. Setiap topik dari model LDA dipetakan ke aspek estetika tertentu berdasarkan karakteristik dominan kata-katanya. *Caption* yang telah dibersihkan sebelumnya diubah menjadi representasi *bag-of-words* (BoW) berdasarkan *n*-gram-nya. *N*-gram tersebut dicocokkan terhadap kamus (`dictionary.token2id`) dari model LDA agar hanya token yang dikenali yang digunakan.

Setiap *caption* kemudian dievaluasi dengan fungsi `get_document_topics()` dari model LDA untuk memperoleh distribusi probabilitas topik. Probabilitas tertinggi dari topik yang valid (sesuai mapping) menentukan aspek estetika dari *caption* tersebut. Hasil pemetaan *caption* disimpan dalam enam file `JSON` terpisah sesuai aspeknya, dan statistik ringkasan seperti jumlah *caption* per aspek disimpan ke dalam file `aspect_summary.json`. Proses ini juga menangani *caption* yang tidak dapat di-*mapping*-kan akibat seluruh token-nya tidak dikenali di kamus jumlah *caption* ini dilaporkan sebagai *skipped*.

```
INPUT:  clean.json,  lda_topics_60.model,  lda_topics_60.model.id2word,
topic_map
OUTPUT:      color_light.json,      composition.json,      subject.json,
dof_and_focus.json,     general_impression.json,     use_of_camera.json,
aspect_summary.json

1: BEGIN
2: topic_map ← DICTIONARY_MAPPING_TOPIC_TO_ASPECT()
3: lda_model ← LOAD_LDA_MODEL("lda_topics_60.model")
4: dictionary ← LOAD_DICTIONARY("lda_topics_60.model.id2word")
5: data ← LOAD_JSON("clean.json")["images"]
6:
7: INIT aspect_data ← {aspect: {"images": []} for each of 6 aspects}
8:
9: FOR img IN data DO
10:       INIT aspect_sentences ← {aspect: {"filename": img["filename"],
"url": img["url"], "sentences": []}}
11:
12:     FOR cap IN img["sentences"] DO
13:         ngrams ← EXTRACT_NGRAMS(cap)  // gabungan unigram dan bigram
14:         filtered ← FILTER_NGRAMS_IN_DICTIONARY(ngrams, dictionary)
15:         bow ← dictionary.DOC2BOW(filtered)
16:         IF bow IS EMPTY THEN CONTINUE
17:
18:         topic_distribution ← lda_model.GET_DOCUMENT_TOPICS(bow)
19:           best_topic ← SELECT_HIGHEST_VALID_TOPIC(topic_distribution,
topic_map)
20:         assigned_aspect ← topic_map[best_topic]
21:
aspect_sentences[assigned_aspect]["sentences"].APPEND(cap["clean"])
22:     END FOR
23:
24:     FOR aspect, content IN aspect_sentences DO
25:         IF content["sentences"] IS NOT EMPTY THEN
26:             aspect_data[aspect]["images"].APPEND(content)
27:         END IF
28:     END FOR
29: END FOR
30:
31: FOR aspect, content IN aspect_data DO
32:     SAVE_JSON(content, f"{aspect}.json")
33: END FOR
34:
35: summary ← {aspect: COUNT_TOTAL_CAPTIONS(content) for each aspect}
36: SAVE_JSON(summary, "aspect_summary.json")
37: END
```

Kode Semu 3.5 *Map Topics to Aspects*

sistem memuat model LDA dan kamus *n*-gram dari proses sebelumnya. *Caption* dalam `clean.json` kemudian diproses satu per satu. Untuk setiap *caption*, fungsi `EXTRACT_NGRAMS`

digunakan untuk mengambil *unigram* dan *bigram*, lalu disaring agar hanya *n*-gram yang terdapat dalam kamus yang digunakan. *Caption* yang memiliki representasi BoW valid akan diproses oleh model LDA untuk mendapatkan distribusi topik. Dari distribusi tersebut, topik dengan probabilitas tertinggi yang telah dipetakan ke aspek estetika digunakan untuk menentukan aspek *caption* tersebut. *Caption* kemudian ditambahkan ke dalam struktur data berdasarkan aspek tersebut. Pada akhir proses, *caption* yang telah dikelompokkan disimpan ke dalam file `JSON` sesuai aspek. Statistik total *caption* per aspek juga dihitung dan disimpan dalam *file* ringkasan. *Caption* yang tidak dapat dipetakan dilaporkan sebagai bagian dari analisis kualitas pemetaan, misalnya karena semua *n*-gram tidak dikenali dalam kamus.

Setelah *caption* berhasil dipetakan ke dalam enam aspek estetika, tahap selanjutnya adalah membagi *dataset* ke dalam tiga subset: *train*, *validation*, dan *test*. Tujuannya adalah untuk memastikan bahwa proses pelatihan dan evaluasi model dilakukan secara adil dan tidak terjadi kebocoran data antar tahap. Pembagian ini mengikuti rasio umum 6:1:1 atau 75% untuk pelatihan, 12.5% untuk validasi, dan 12.5% untuk pengujian. *Dataset* dari masing-masing aspek (`aspect.json`) serta versi gabungan keseluruhan (`clean.json`) dibagi menggunakan metode `split_indice`, yang mengacak indeks data secara acak kemudian membaginya ke dalam tiga subset berdasarkan jumlah data. Untuk setiap caption dalam gambar, dilakukan tokenisasi menggunakan *regular expression tokenizer* dari pustaka nltk, lalu ditambahkan metadata seperti *split* dan nama aspek. Output dari proses ini adalah file `JSON` dengan struktur sesuai standar Karpathy split, dan disimpan di direktori /final untuk digunakan dalam pelatihan model selanjutnya.

```
INPUT: aspect_outputs/*.json, clean.json
OUTPUT: final/*.json dengan format split: train, val, test

BEGIN
DEFINE tokenizer ← REGEXP_TOKENIZER(r'\w+\S*\w*')

FOR each path IN aspect_outputs DO
    image_list ← LOAD_JSON(path)["images"]
     train_ids, val_ids, test_ids ← SPLIT_INDICES(len(image_list))  //
rasio 6:1:1

    INIT split_image_list ← []
    FOR imgID, img IN ENUMERATE(image_list) DO
        IF imgID IN train_ids THEN img["split"] ← "train"
        ELSE IF imgID IN val_ids THEN img["split"] ← "val"
        ELSE img["split"] ← "test"

        FOR cap IN img["sentences"] DO
            tokens ← tokenizer.TOKENIZE(cap)
            REPLACE cap WITH {"raw": cap, "tokens": tokens}
        END FOR

        img["aspect"] ← EXTRACT_ASPECT_FROM_FILENAME(path)
        split_image_list.APPEND(img)
    END FOR

    SAVE_JSON({"images": split_image_list}, f"final/{path}")
END FOR

// Proses file clean.json sebagai dataset gabungan
image_list ← LOAD_JSON("clean.json")["images"]
train_ids, val_ids, test_ids ← SPLIT_INDICES(len(image_list))

```

```
FOR imgID, img IN ENUMERATE(image_list) DO
    IF imgID IN train_ids THEN img["split"] ← "train"
    ELSE IF imgID IN val_ids THEN img["split"] ← "val"
    ELSE img["split"] ← "test"

    FOR cap IN img["sentences"] DO
        tokens ← tokenizer.TOKENIZE(cap["clean"])
        REPLACE cap WITH {"raw": cap["clean"], "tokens": tokens}
    END FOR

    split_image_list.APPEND(img)
END FOR

SAVE_JSON({"images": split_image_list}, "final/all.json")
END
```

Kode Semu 3.6 *Split Caption Dataset per Aspect*

3.3.4 Implementasi Praproses *Dataset Fine-tune* IAC

Proses praproses *dataset fine-tune* IAC bertujuan untuk mengubah data mentah hasil *splitting* dan pemetaan menjadi format siap pakai untuk pelatihan model CNN-LSTM. Tahapan ini dilakukan terhadap tujuh subset data yaitu `all`, `color_light`, `composition`, `dof_and_focus`, `general_impression`, `subject`, dan `use_of_camera`. Langkah pertama dalam proses ini adalah memuat seluruh token *caption* dan membentuk *word map*. Untuk memastikan kompatibilitas dengan *pretrained* GloVe, setiap token dikoreksi menggunakan kamus `missed_words_all.json` jika tersedia, atau dikoreksi ejaannya menggunakan modul pyspellchecker, selama hasil koreksi tersebut ada dalam kosakata GloVe. Token yang tidak dapat dikenali akan dimasukkan sebagai `<unk>`, sementara token spesial seperti `<start>`, `<end>`, dan `<pad>` juga ditambahkan secara eksplisit ke dalam *word map*. Setelah word map terbentuk, setiap *caption* diubah menjadi urutan indeks berdasarkan token yang telah dikoreksi dan dipetakan. *Caption* ini kemudian di-*encode* dan dipotong atau dipadatkan ke panjang maksimum yang ditentukan, dengan padding `<pad>` di akhir jika perlu. Satu gambar dapat memiliki beberapa *caption*, dan jumlah *caption* ini diatur agar konsisten per gambar (default: 5 *caption*). Gambar-gambar yang terkait kemudian dimuat, diubah ukurannya menjadi 256×256 piksel, dan disimpan ke dalam file `.hdf5` per split (train/val/test) dan per aspek. Setiap gambar akan disimpan dalam format tensor 3×256×256. Bersamaan dengan gambar, *caption* terencode dan panjang masing-masing *caption* disimpan dalam file `.json`. Setiap aspek diproses secara independen, dan aspek `all` akan membentuk *word map* baru dari seluruh korpus, sementara aspek lainnya menggunakan *word map* dari `all` agar konsisten.

```
INPUT: final/all.json, glove.6B.300d.txt, missed_words_all.json
OUTPUT:          preprocessed_dataset/{split}_{images,          captions,
caplength}_{aspect}.hdf5/.json, wordmap_all.json

BEGIN
Load missed_word_dict ← JSON("missed_words_all.json")
Load glove_vocab ← LOAD_GLOVE_VOCAB("glove.6B.300d.txt")
FOR each aspect IN ["all", "color_light", "composition", ...,
"use_of_camera"] DO
    data ← LOAD_JSON(f"final/{aspect}.json")["images"]

    IF aspect == "all" THEN
        all_tokens ← FLATTEN([cap["tokens"] FOR img IN data FOR cap IN
img["sentences"]])
```

```
            word_map, missed ← BUILD_WORD_MAP(all_tokens, glove_vocab,
missed_word_dict)
        SAVE_JSON(word_map, f"preprocessed_dataset/wordmap_all.json")
       SAVE_JSON(missed, f"preprocessed_dataset/missed_words_all.json")
    ELSE
        word_map ← LOAD_JSON("preprocessed_dataset/wordmap_all.json")

    FOR split IN ["train", "val", "test"] DO
        split_data ← FILTER(data, img → img["split"] == split)
        image_paths ← []
        all_captions ← []
        FOR each img IN split_data DO
            valid_caps ← [cap["tokens"] FOR cap IN img["sentences"] IF
5 ≤ len(cap["tokens"]) ≤ 100]
            IF valid_caps IS EMPTY THEN CONTINUE
            image_paths.APPEND(f"images/{img['filename']}")
            all_captions.APPEND(valid_caps)

        CREATE hdf5_file ← f"{split}_images_{aspect}.hdf5"
        FOR i IN 0 TO len(image_paths) - 1 DO
            image ← LOAD_AND_RESIZE_IMAGE(image_paths[i], size=256)
            STORE hdf5_file[i] ← image

            captions ← all_captions[i]
            IF len(captions) < 5 THEN
                captions ← PAD_TO_5(captions)
            ELSE
                captions ← RANDOM_SAMPLE(captions, 5)

            FOR cap IN captions DO
                enc ← ENCODE_CAPTION(cap, word_map, max_len=100)
                encoded_captions.APPEND(enc)
                cap_lengths.APPEND(len(cap) + 2)
        END FOR

        SAVE_JSON(encoded_captions, f"{split}_captions_{aspect}.json")
        SAVE_JSON(cap_lengths, f"{split}_caplength_{aspect}.json")
    END FOR
END FOR
END
```

Kode Semu 3.7 Praproses *Dataset Fine-tune*

3.3.5 Implementasi *Fine-tuning* Model CNN-LSTM

Setelah tahap *pretraining*, model CNN-LSTM diadaptasi secara spesifik untuk menghasilkan *caption* estetika berdasarkan enam aspek visual yang telah dipetakan sebelumnya. Proses fine-tuning ini dilakukan secara terpisah untuk setiap aspek menggunakan *dataset* hasil *split* format Karpathy. Model yang digunakan mencakup arsitektur *encoder* CNN berbasis ResNet101 dan *decoder* LSTM dengan *pretrained* GloVe *embedding* sebesar 300 dimensi. Strategi *fine-tuning* dilakukan dengan *selective weight loading*: *encoder* diinisialisasi penuh, sementara *decoder* hanya memuat parameter yang kompatibel dari hasil *pretraining*. *Early stopping* digunakan untuk mencegah *overfitting*, dengan evaluasi menggunakan *loss* validasi pada setiap *epoch*.

```
INPUT:       preprocessed_dataset       (folder),       wordmap_all.json,
pretrained_checkpoint.pth
OUTPUT: fine-tuned_{aspect}.pth, training_log.json

1: BEGIN
```

```
word_map ← LOAD_JSON("wordmap_all.json")
embedding_matrix ← LOAD_GLOVE("glove.6B.300d.txt", word_map)
aspects ← ["general_impression", "subject", "use_of_camera", ...]

FOR aspect IN aspects DO
    train_loader ← LOAD_DATASET(split="train", aspect, transform)
    val_loader ← LOAD_DATASET(split="val", aspect, transform)

        encoder, decoder ← LOAD_PRETRAINED_MODEL(embedding_matrix,
pretrained_checkpoint)
    criterion ← CrossEntropyLoss()
    optimizer_enc ← Adam(encoder.parameters(), lr)
    optimizer_dec ← Adam(decoder.parameters(), lr)

    best_val_loss ← ∞; patience ← 3; no_improve ← 0
    FOR epoch IN range(1, max_epoch + 1) DO
        train_loss ← TRAIN_EPOCH(train_loader, encoder, decoder, ...)
        val_loss ← EVAL_EPOCH(val_loader, encoder, decoder, ...)

        IF val_loss < best_val_loss THEN
            best_val_loss ← val_loss
                    SAVE_CHECKPOINT(encoder, decoder, optimizer_enc,
optimizer_dec, epoch, val_loss)
            no_improve ← 0
        ELSE
            no_improve ← no_improve + 1
            IF no_improve ≥ patience THEN
                BREAK  // Early stopping
            END IF
        END IF
    END FOR
END FOR
END
```

Kode Semu 3.8 *Fine-tuning* Model IAC

Proses pelatihan berlangsung selama maksimal 20 *epoch*, dengan pemantauan terhadap performa validasi untuk setiap aspek. Optimisasi dilakukan menggunakan Adam optimizer dan *loss function* Cross Entropy. Fungsi `train_epoch()` dan `evaluate()` bertanggung jawab untuk menjalankan proses *forward* dan *backward pass*, serta penghitungan *loss* pada data pelatihan dan validasi. *Checkpoint* model terbaik disimpan menggunakan fungsi `save_checkpoint()` yang secara otomatis menghapus *file checkpoint* lama untuk efisiensi penyimpanan. Dengan pendekatan ini, model dapat menghasilkan *caption* estetika yang terfokus dan relevan terhadap masing-masing aspek visual yang ditentukan, sekaligus menjaga efisiensi pelatihan melalui *reuse* parameter dan strategi *early stopping*.

3.3.6 Implementasi Pembuatan *Dataset* DAE

Setelah seluruh model CNN-LSTM telah selesai di-*fine-tune* untuk masing-masing dari enam aspek estetika, langkah selanjutnya adalah membuat *dataset* untuk pelatihan model DAE. *Dataset* ini dirancang untuk menggabungkan *caption* yang dihasilkan oleh keenam model aspek menjadi satu representasi terstruktur, sehingga DAE dapat belajar menyintesis *caption* estetika yang holistik berdasarkan masukan multi-aspek. Proses ini dilakukan dengan cara melakukan inferensi *caption* dari setiap gambar menggunakan keenam model aspek tersebut. Setiap gambar diproses secara individual, dimulai dengan pemuatan dan *preprocessing* citra, lalu masing-masing model menghasilkan *caption* untuk aspek tertentu dengan strategi *decoding greedy*. *Caption* hasil prediksi ini dikumpulkan dan disimpan dalam bentuk *dictionary captions*

per aspek. Sebagai referensi *ground truth*, digunakan caption asli dari dataset `all.json` yang telah melalui proses *cleaning*.

```
INPUT: all.json, folder images/, 6 model CNN-LSTM hasil fine-tune
OUTPUT: dae_dataset.json

BEGIN
word_map ← LOAD_JSON("wordmap_all.json")
data_json ← LOAD_JSON("all.json")
model_dict ← EMPTY_DICTIONARY

 FOR aspect IN ["general_impression", "subject", ..., "dof_and_focus"]
DO
    checkpoint ← LOAD_CHECKPOINT(aspect + "_best.pth")
    encoder ← LOAD_ENCODER_FROM_CHECKPOINT(checkpoint)
    decoder ← LOAD_DECODER_FROM_CHECKPOINT(checkpoint)
     model_dict[aspect] ← (encoder, decoder)
 END FOR

 transform ← DEFINE_TRANSFORM(resize=256, normalize=True)
 all_samples ← EMPTY_LIST

 FOR image_info IN data_json["images"] DO
    filename ← image_info["filename"]
    IF NOT EXISTS(image_path):
        CONTINUE
    image ← LOAD_IMAGE(filename)
    image_tensor ← APPLY_TRANSFORM(image)

    aspect_captions ← EMPTY_DICTIONARY
    FOR aspect, (encoder, decoder) IN model_dict DO
          caption ← GENERATE_CAPTION(encoder, decoder, image_tensor,
word_map)
        aspect_captions[aspect] ← caption
    END FOR

              reference_captions ← [sent["raw"] FOR sent IN
image_info["sentences"]]
    sample ← {
        "image_id": filename,
        "captions": aspect_captions,
        "reference": reference_captions
    }
    all_samples.APPEND(sample)
 END FOR

 SAVE_JSON("dae_dataset.json", all_samples)
 END
```

Kode Semu 3.9 Pembuatan *Dataset* DAE

Hasil akhir dari proses ini adalah file `dae_dataset.json`, yang berisi daftar gambar beserta caption per aspek dan daftar caption referensi. Dataset ini digunakan untuk melatih model DAE yang bertugas menghasilkan caption estetika akhir yang menggabungkan informasi dari seluruh aspek secara harmonis.

3.3.7 Implementasi Praproses *Dataset* DAE

Dataset `dae_dataset.json` yang sebelumnya dibuat berisi daftar gambar dengan *caption* hasil prediksi dari keenam aspek estetika serta *caption* referensi sebagai target. Untuk

mempersiapkannya menjadi *input* bagi model *sequence-to-sequence* (seperti DAE), dilakukan proses tokenisasi dan *padding* agar dapat dibentuk menjadi tensor dengan ukuran yang seragam.

Pertama, seluruh *caption* per aspek dalam satu gambar digabung menjadi satu string panjang, dengan penanda aspek seperti `general_impression` dan sebagainya untuk mempertahankan konteks. *Caption* referensi digunakan sebagai *target*—biasanya dipilih satu dari sekian banyak *caption ground truth* yang tersedia, misalnya yang pertama.

Seluruh kalimat input dan target kemudian dikonversi menjadi urutan ID token menggunakan kamus `word_map`, lalu ditambahkan token khusus `<start>` dan `<end>`. Karena panjang setiap urutan bisa berbeda, digunakan fungsi `pad_sequence` dari PyTorch untuk melakukan *padding* sehingga semua memiliki dimensi yang sama. Token `<pad>` digunakan sebagai nilai padding. Agar dataset lebih bersih dan efisien untuk pelatihan, dilakukan filtering terhadap sampel yang memiliki target caption terlalu panjang. Dalam implementasi ini, hanya caption target dengan panjang ≤ 64 token yang disimpan. Dataset akhir disimpan dalam file PyTorch (`.pt`) dengan nama `dae_preprocessed_filtered.pt`.

```
INPUT: dae_dataset.json, wordmap_all.json
OUTPUT: dae_preprocessed_filtered.pt

BEGIN
dae_data ← LOAD_JSON("dae_dataset.json")
word_map ← LOAD_JSON("wordmap_all.json")
input_seqs ← EMPTY_LIST
target_seqs ← EMPTY_LIST

FUNCTION tokenize(text, word_map):
    tokens ← text.lower().split()
    RETURN [word_map.get(token, word_map['<unk>']) for token in
tokens]

FOR sample IN dae_data DO
    input_txt ← CONCATENATE([
        "[general_impression] " +
sample["captions"]["general_impression"],
        "[subject] " + sample["captions"]["subject"],
        "[use_of_camera] " + sample["captions"]["use_of_camera"],
        "[color_light] " + sample["captions"]["color_light"],
        "[composition] " + sample["captions"]["composition"],
        "[dof_and_focus] " + sample["captions"]["dof_and_focus"]
    ])

    target_txt ← sample["reference"][0]  # ambil 1 caption ground
truth

    input_ids ← [<start>] + tokenize(input_txt, word_map) + [<end>]
    target_ids ← [<start>] + tokenize(target_txt, word_map) + [<end>]

    input_seqs.APPEND(torch.tensor(input_ids))
    target_seqs.APPEND(torch.tensor(target_ids))
END FOR

input_pad ← PAD_SEQUENCE(input_seqs, padding_value=word_map['<pad>'])
target_pad ← PAD_SEQUENCE(target_seqs,
padding_value=word_map['<pad>'])

max_len ← 64
new_input, new_target ← EMPTY_LIST, EMPTY_LIST
FOR inp, tgt IN ZIP(input_pad, target_pad) DO
```

```
36:     IF NON_PAD_LENGTH(tgt) ≤ max_len THEN
37:         new_input.APPEND(inp)
38:         new_target.APPEND(tgt)
39:     END IF
40: END FOR
41:
42: new_input_pad ← PAD_SEQUENCE(new_input,
padding_value=word_map['<pad>'])
43: new_target_pad ← PAD_SEQUENCE(new_target,
padding_value=word_map['<pad>'])
44:
45: SAVE_PT("dae_preprocessed_filtered.pt", {
46:     "input_seqs": new_input_pad,
47:     "target_seqs": new_target_pad,
48:     "word_map": word_map
49: })
50: END
```

Kode Semu 3.10 Praproses *Dataset* DAE

3.3.8 Implementasi *Training* Model DAE

Model *Denoising Autoencoder* pada Tugas Akhir ini bertujuan untuk merekonstruksi *caption* yang merepresentasikan semua aspek estetika dari enam *caption* parsial. Arsitektur yang digunakan terdiri atas dua bagian utama: *encoder* BiLSTM (*bidirectional* LSTM) dan *decoder* LSTM biasa. *Embedding* layer mengubah token *input* menjadi representasi vektor, kemudian *encoder* menghasilkan *hidden state* gabungan dari arah *forward* dan *backward* yang digunakan untuk menginisialisasi *decoder*. *Decoder* akan menghasilkan urutan kata yang mendekati *caption ground truth* penuh.

Dataset yang digunakan berupa pasangan *input-output* yang telah diproses dalam `dae_preprocessed_filtered.pt`. Data dimuat menggunakan `DAEDataset` dan dibungkus dalam `DataLoader` dengan `batch_size = 32`. *Optimizer* yang digunakan adalah Adam, dan fungsi loss-nya adalah `CrossEntropyLoss`, dengan token `<pad>` diabaikan dalam perhitungan *loss*.

Proses pelatihan dilakukan selama beberapa *epoch*. Pada setiap batch, *input* dan target *caption* diproses, dan *loss* dihitung berdasarkan hasil prediksi model dibandingkan dengan target. *Caption* target digeser satu langkah ke kanan agar model memprediksi kata berikutnya. Jika *loss* pada *epoch* saat ini lebih baik dari sebelumnya, model disimpan sebagai *checkpoint* terbaik.

```
INPUT: dae_preprocessed_filtered.pt, wordmap_all.json
OUTPUT: dae_lstm_best.pt

1: BEGIN
2: word_map ← LOAD_JSON("wordmap_all.json")
3: vocab_size ← LENGTH(word_map)
4: pad_idx ← word_map["<pad>"]
5:
6: dataset ← LOAD_TORCH_DATASET("dae_preprocessed_filtered.pt")
7: train_loader ← DATALOADER(dataset, batch_size=32, shuffle=True)
8:
9:    model    ←    INIT_DAE_LSTM(vocab_size=vocab_size,    embed_dim=300,
hidden_dim=512, pad_idx=pad_idx)
10: optimizer ← INIT_ADAM(model.parameters(), lr=1e-3)
11: criterion ← CrossEntropyLoss(ignore_index=pad_idx)
12:
13: best_loss ← ∞
```

```
FOR epoch IN range(1, n_epochs + 1) DO
    model.train()
    total_loss ← 0

    FOR inputs, targets IN train_loader DO
        inputs ← inputs.to(device)
        targets ← targets.to(device)
        optimizer.zero_grad()

        outputs ← model(inputs, targets)  # (B, T, V)
        outputs ← RESHAPE(outputs, [-1, vocab_size])
        gold ← RESHAPE(targets[:, 1:], [-1])

        loss ← criterion(outputs, gold)
        loss.backward()
        optimizer.step()
        total_loss += loss.item()
    END FOR

    avg_loss ← total_loss / LENGTH(train_loader)
    PRINT("Epoch", epoch, "Loss:", avg_loss)

    IF avg_loss < best_loss THEN
        best_loss ← avg_loss
        SAVE_MODEL(model.state_dict(), "dae_lstm_best.pt")
        PRINT("Best model saved.")
    END IF
END FOR
PRINT("Training selesai. Best loss:", best_loss)
END
```

Kode Semu 3.11 *Training* Model DAE

3.3.9 Implementasi *Retouch Caption* dengan LLM dan Evaluasi

Setelah caption dari masing-masing aspek estetika berhasil dihasilkan oleh model CNN-LSTM dan digabungkan oleh model DAE, tahap akhir dari pembuatan *Image Aesthetic Captioning* adalah proses *retouching*. Tujuan *retouch* ini adalah untuk menyempurnakan *caption* hasil DAE menjadi kalimat utuh yang alami, namun tetap mempertahankan informasi eksplisit dari keenam aspek estetika. Proses ini dilakukan dengan bantuan *Large Language Model* (LLM), yaitu GPT-4o-mini melalui API OpenAI. Untuk memulai proses *retouching*, dilakukan inferensi terlebih dahulu terhadap gambar input menggunakan enam model CNN-LSTM yang telah di-*fine-tune* secara khusus untuk masing-masing aspek*: general impression, subject, use of camera, color/light, composition,* dan *depth of field (dof)/focus*. Tiap model menghasilkan *caption* pendek yang relevan untuk satu aspek. *Caption-caption* ini kemudian digabung secara linear ke dalam sebuah string input DAE dengan menyisipkan tag aspek di setiap bagian, seperti `[general_impression] caption`, `[subject] caption`, dan seterusnya. Input gabungan ini dikodekan ke dalam urutan token menggunakan `word_map`, dan disusun dalam format yang sesuai untuk model DAE. Model DAE kemudian melakukan *decoding* untuk menghasilkan satu *caption* tunggal. *Caption* ini sudah mencerminkan gabungan aspek, namun struktur bahasanya belum optimal secara linguistik karena dibentuk melalui model autoregresif sederhana yang dilatih terbatas.

Untuk memperhalus *caption*, digunakan pendekatan *retouching* berbasis LLM. Dalam hal ini, GPT-4o-mini digunakan dengan *prompt* yang sangat spesifik dan bersifat terbatas. *Prompt* menjelaskan bahwa caption DAE perlu disusun ulang agar lebih alami dan mengalir, namun tidak boleh mengubah struktur utama atau menambahkan informasi baru. GPT hanya boleh

memperbaiki kealamian bahasa, menyatukan gaya penyampaian, dan menghapus token `<unk>` jika ada. *Prompt* juga mewajibkan agar seluruh informasi dari keenam aspek tetap tercakup dalam *caption* akhir. Proses *retouch* ini dapat dilakukan dengan atau tanpa gambar. Jika gambar tersedia, maka dikodekan dalam format `base64` dan dikirim sebagai bagian dari *payload* OpenAI API. Namun, pada dasarnya model akan tetap mengandalkan *textual alignment* antara *input* DAE dan *caption* DAE sebagai dasar *retouching*. *Output* akhir dari proses ini adalah sebuah kalimat yang secara eksplisit mencerminkan keenam aspek estetika dalam satu kalimat utuh, alami, dan terstruktur dengan baik. *Caption* tersebut kemudian ditampilkan bersama *caption* per aspek, input gabungan, hasil DAE, serta *caption* akhir dari GPT untuk keperluan evaluasi.

```
BEGIN
word_map ← LOAD_JSON("wordmap_all.json")
inv_word_map ← INVERT_DICT(word_map)
device ← SELECT_DEVICE("cuda" if available else "cpu")

# Load semua model CNN-LSTM untuk tiap aspek
aspect_models ← {}
FOR aspect, ckpt_path IN ASPECT_MODEL_PATHS DO
    encoder ← LOAD_ENCODER_MODEL(ckpt_path).to(device)
     decoder ← LOAD_DECODER_MODEL(ckpt_path).to(device)
     encoder.eval(); decoder.eval()
     aspect_models[aspect] ← (encoder, decoder)
END FOR

# Load model DAE
dae_model ← LOAD_DAE_MODEL("dae_lstm_best.pt", word_map).to(device)
dae_model.eval()

# Step 1: Generate caption per aspek
aspect_captions ← {}
FOR aspect, (encoder, decoder) IN aspect_models DO
      caption ← GENERATE_CAPTION_ASPECT(image_path, encoder, decoder,
word_map, device)
    aspect_captions[aspect] ← caption
END FOR

# Step 2: Gabungkan input untuk DAE
dae_input ← ""
FOR aspect, caption IN aspect_captions DO
    dae_input ← CONCAT(dae_input, "[", aspect, "] ", caption, " ")
END FOR
 dae_ids ← [word_map["<start>"]] + TOKENIZE(dae_input, word_map) +
[word_map["<end>"]]

# Step 3: Decode caption dari DAE
caption_dae ← DECODE_DAE_CAPTION(dae_model, dae_ids, word_map, device)

# Step 4: Retouch menggunakan GPT API
caption_retouch ← CALL_GPT_API(
    caption_dae=caption_dae,
    dae_input=dae_input,
    img_path=image_path,
    api_key=OPENAI_API_KEY
)

# Step 5: Simpan hasil akhir
SAVE_TEXT(caption_retouch, "caption_final_retouch.txt")
```

```
46: PRINT("Retouch caption selesai:", caption_retouch)
47: END
```

Kode Semu 3.12 *Retouch Caption* dengan LLM

Proses inferensi dan *retouch caption* terdiri dari beberapa tahap utama yang divisualisasikan dalam Kode Semu 3.13. Langkah pertama diawali dengan memuat kamus `word_map` dari file JSON, yang berisi pemetaan kata ke indeks numerik sesuai dengan hasil *preprocessing dataset*. Kamus ini kemudian dibalik menggunakan fungsi `invert_dict` untuk memungkinkan proses *decoding* dari indeks menjadi teks. Selanjutnya, sistem secara otomatis memilih perangkat eksekusi antara CPU atau GPU (baris 2–4), untuk memastikan efisiensi selama proses inferensi berlangsung.

Tahap berikutnya adalah memuat seluruh model CNN-LSTM yang telah dilatih berdasarkan aspek-aspek estetika seperti *composition*, *color and lighting*, *subject*, *depth of field*, dan lainnya. Masing-masing aspek memiliki pasangan *encoder* dan *decoder* tersendiri, yang dimuat dari *checkpoint* dan disimpan dalam struktur *dictionary* bernama `aspect_models`. Setiap model dievaluasi dalam mode `eval()` untuk memastikan bahwa parameter tidak berubah selama inferensi. Proses pemuatan model ini dilakukan secara iteratif pada baris 7–13 dalam Kode Semu 3.13.

Setelah semua model aspek dimuat, proses inferensi dimulai dengan melewatkan citra input ke masing-masing model CNN-LSTM untuk menghasilkan caption per aspek (baris 19–24). Hasil *caption* dari tiap aspek ini kemudian digabungkan ke dalam satu *string* panjang sebagai input bagi model *Denoising Autoencoder* (DAE). Gabungan ini menggunakan format spesifik `[aspect]` *caption*, sehingga model DAE dapat mengenali batas antar aspek saat membentuk kalimat koheren. String input tersebut ditokenisasi dan dibungkus oleh token khusus `<start>` dan `<end>`, lalu diberikan ke model DAE (baris 26–31).

Setelah *caption* dikodekan oleh DAE (baris 34), hasilnya dikirimkan bersama input awal ke GPT-4o-mini melalui OpenAI API. Proses ini bertujuan untuk *retouch* caption agar lebih natural dan bebas dari token seperti `<unk>`, tanpa mengubah makna dan cakupan aspek informasi yang dikandung. Proses ini dijalankan menggunakan *prompt* yang eksplisit untuk menjaga struktur dan semantik kalimat. *Output* yang dihasilkan oleh GPT disimpan sebagai *caption* akhir (baris 36–46).

```
1: BEGIN
2: word_map ← LOAD_JSON("wordmap_all.json")
3: inv_word_map ← INVERT_DICT(word_map)
4: device ← SELECT_DEVICE("cuda" if available else "cpu")
5:
6: # Load semua model CNN-LSTM untuk tiap aspek
7: aspect_models ← {}
8: FOR aspect, ckpt_path IN ASPECT_MODEL_PATHS DO
9:     encoder ← LOAD_ENCODER_MODEL(ckpt_path).to(device)
10:     decoder ← LOAD_DECODER_MODEL(ckpt_path).to(device)
11:     encoder.eval(); decoder.eval()
12:     aspect_models[aspect] ← (encoder, decoder)
13: END FOR
14:
15: # Load model DAE
16: dae_model ← LOAD_DAE_MODEL("dae_lstm_best.pt", word_map).to(device)
17: dae_model.eval()
```

```
18:
19: # Step 1: Generate caption per aspek
20: aspect_captions ← {}
21: FOR aspect, (encoder, decoder) IN aspect_models DO
22:      caption ← GENERATE_CAPTION_ASPECT(image_path, encoder, decoder,
word_map, device)
23:     aspect_captions[aspect] ← caption
24: END FOR
25:
26: # Step 2: Gabungkan input untuk DAE
27: dae_input ← ""
28: FOR aspect, caption IN aspect_captions DO
29:     dae_input ← CONCAT(dae_input, "[", aspect, "] ", caption, " ")
30: END FOR
31:  dae_ids ← [word_map["<start>"]] + TOKENIZE(dae_input, word_map) +
[word_map["<end>"]]
32:
33: # Step 3: Decode caption dari DAE
34: caption_dae ← DECODE_DAE_CAPTION(dae_model, dae_ids, word_map, device)
35:
36: # Step 4: Retouch menggunakan GPT API
37: caption_retouch ← CALL_GPT_API(
38:     caption_dae=caption_dae,
39:     dae_input=dae_input,
40:     img_path=image_path,
41:     api_key=OPENAI_API_KEY
42: )
43:
44: # Step 5: Simpan hasil akhir
45: SAVE_TEXT(caption_retouch, "caption_final_retouch.txt")
46: PRINT("Retouch caption selesai:", caption_retouch)
47: END
```

Kode Semu 3.13 Evaluasi *Caption Retouch*

BAB 4
HASIL DAN PEMBAHASAN

4.1 Hasil Penelitian

Bab ini berisikan pemaparan hasil dan pembahasan dari setiap langkah implementasi yang telah dilakukan pada Tugas Akhir ini. Hasil akhir menampilkan keseluruhan alur sistem *Image Aesthetic Caption Generator* yang dibangun menggunakan pendekatan kombinasi model CNN-LSTM, *mapping* topik dengan LDA, model *autoencoder* DAE, dan penyempurnaan kalimat dengan bantuan *Large Language Model* (LLM). Penelitian ini mencakup tahapan mulai dari pra-pemrosesan *dataset* MS COCO 2017 untuk *pretraining* model dasar, dilanjutkan dengan proses *fine-tuning* model CNN-LSTM berdasarkan enam aspek estetika dari *dataset* IAC yang telah dipetakan sebelumnya. Setelahnya, dilakukan inferensi *caption* untuk setiap gambar menggunakan keenam model tersebut, dan hasilnya digabung untuk membentuk input bagi model *Denoising Autoencoder*. Caption dari DAE kemudian disempurnakan melalui proses *retouch* menggunakan GPT agar menghasilkan kalimat yang lebih natural tanpa kehilangan konteks estetika dari seluruh aspek. Setiap tahapan diuraikan secara rinci pada subbab-subbab berikut, dimulai dari proses awal pra-pemrosesan data hingga pada tahap akhir berupa generasi *caption* estetika yang telah *diretouch* oleh LLM.

4.1.1 Pra-pemrosesan *Dataset* MS COCO 2017

Tahap awal dalam proses pelatihan model adalah melakukan pra-pemrosesan terhadap *dataset* MS COCO 2017, yang secara *default* berisi lebih dari 118.000 gambar dengan total 591.753 anotasi *caption*. Karena fokus dari sistem ini adalah pada estetika makanan, maka diperlukan seleksi *subset* gambar yang relevan, pembersihan *caption*, dan pembentukan struktur data khusus untuk pelatihan model *captioning*. *Dataset* ini terdiri dari anotasi *caption* yang sudah dibagi menjadi *subset train*, *val*, dan *test*.

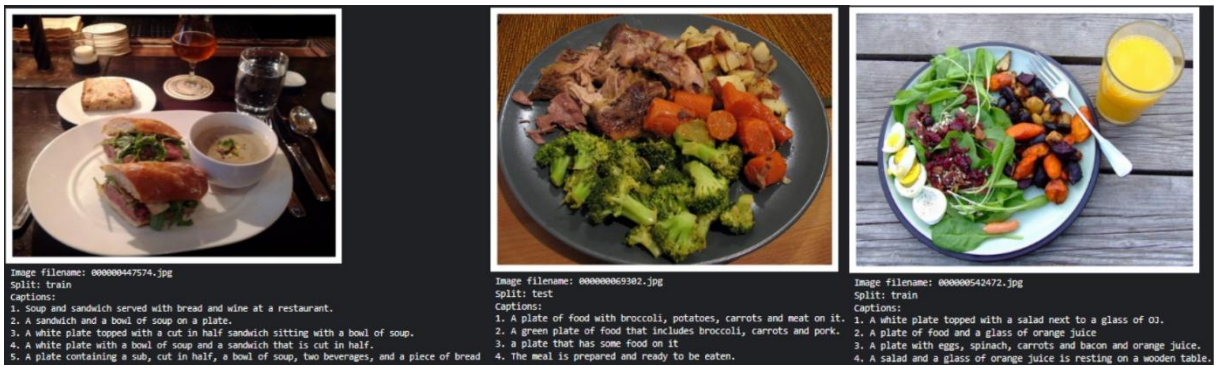
Langkah pertama adalah menyaring *caption-caption* yang mengandung kata kunci makanan, seperti *"food", "pizza", "plate", "bowl", "table", "kitchen", "dining", "meal", "cake", "bread", "sandwich", "lunch", "dinner", "breakfast", "cupcake", "cheese", "pasta", "dish", "eating", "cook", "burger", "noodle", "chopstick", "restaurant", "cooking", "doughnut", "fork", "spoon", "knife"*. *Caption* yang mengandung setidaknya satu kata dari daftar kata kunci ini dianggap relevan. Proses ini dilakukan secara efisien terhadap seluruh *caption* dalam *file* captions_train2017.json yang memuat seluruh anotasi caption gambar pada *subset train*. Sebagai hasilnya, diperoleh sebanyak 23.251 gambar makanan, yang kemudian digunakan sebagai dasar untuk pelatihan model. *Dataset* hasil seleksi kemudian dibagi menjadi tiga bagian, mengikuti format Karpathy split yang umum digunakan dalam pelatihan model *captioning*. Masing-masing gambar memiliki minimal dua *caption* dan secara struktural disimpan dalam format Karpathy di *file* coco_food_karpathy.json. Distribusi *dataset* yang telah difilter dapat dilihat pada Tabel 4.1.

Tabel 4.1 Distribusi *Dataset* MS COCO Setelah Difilter

Nama *Subset*	Rasio *Split*	Total Gambar
Train	80%	18.600
Validation	10%	2.325
Test	10%	2.326
Total		**23.251**

Setiap *caption* ditokenisasi dengan fungsi sederhana berbasis pemisah spasi (split()) dan dikonversi menjadi huruf kecil. Kemudian dilakukan; perhitungan frekuensi kata, penyaringan berdasarkan ambang batas minimal frekuensi (min_word_freq=5), dan pembuatan word_map untuk meng-*encode* setiap token menjadi ID *integer*. Sebagai hasilnya, *word map* akhir berisi 4.718 token unik, ditambah token khusus: <start>, <end>, <pad>, dan <unk>. Semua gambar yang dipilih kemudian diubah ke format RGB dan di-*resize* menjadi 256 × 256 piksel, disimpan dalam format .hdf5 menggunakan Tensor dari PyTorch. Proses ini dilakukan untuk efisiensi memori saat pelatihan.

Setiap gambar dilengkapi dengan 5 *caption* acak dari *caption* yang tersedia. *Caption* tersebut di-*encode* menggunakan word_map, dan disesuaikan dengan panjang maksimum (max_len=50) menggunakan *padding*.



Gambar 4.1 Contoh Gambar dan *Caption* dari *Dataset* COCO Bertema Makanan

Pra-pemrosesan ini menghasilkan *subset dataset* yang kaya akan variasi visual makanan dan *caption* deskriptif manusiawi. Setiap *caption* memberikan informasi kontekstual yang beragam, mencakup komposisi makanan, warna, hingga aktivitas konsumsi. *Dataset* inilah yang akan digunakan untuk tahap *pretraining* model CNN-LSTM agar dapat memahami struktur bahasa dan hubungan semantik antara gambar makanan dan deskripsi naratifnya.

Tabel 4.2 Statistik Dataset COCO Setelah Pra-pemrosesan

Nama *Subset*	**Jumlah Gambar**	**Jumlah *Caption***	***Caption* per Gambar**
Train	18.600	93.000	5
Validation	2.325	11.625	5
Test	2.326	11.630	5
Total	**23.251**	**116.255**	5

Pada Tabel 4.2 seluruh gambar pada *subset* yang disebutkan memiliki ukuran 256 × 256 piksel. Hal ini tentunya sesuai dengan pernyataan sebelumnya yang menjelaskan format gambar yang dirubah saat pra-pemrosesan datanya.

4.1.2 *Pretraining* Model CNN-LSTM

Setelah proses pra-pemrosesan *dataset* COCO selesai, tahap selanjutnya adalah melakukan *pretraining* model CNN-LSTM pada data makanan tersebut. Tujuan dari *pretraining* ini adalah agar model dapat memahami relasi visual dan linguistik secara umum, sebelum dilakukan *fine-tuning* terhadap aspek estetika tertentu pada tahap selanjutnya.

```
DecoderRNN(
  (embedding): Embedding(4718, 300)
  (dropout): Dropout(p=0.5, inplace=False)
  (init_h): Linear(in_features=2048, out_features=512, bias=True)
  (init_c): Linear(in_features=2048, out_features=512, bias=True)
  (lstm): LSTMCell(2348, 512)
  (f_beta): Linear(in_features=512, out_features=2048, bias=True)
  (fc): Linear(in_features=512, out_features=4718, bias=True)
)
```

Gambar 4.2 *Summary* RNN (LSTM)

Model CNN-LSTM terdiri dari dua komponen utama yaitu; *encoder* menggunakan arsitektur ResNet-101 yang telah dilatih sebelumnya (*pretrained*) pada ImageNet. Layer terakhir dihapus, dan *output* berupa fitur spasial disimpan untuk dilanjutkan ke *decoder*. *Decoder* merupakan LSTM dengan dimensi tersembunyi 512 dan embedding GloVe 300D. *Decoder* ini dilengkapi komponen berikut; *embedding layer* untuk mengubah token ID menjadi vektor kata (dengan 4.718 entri dari word_map), dua layer linear untuk inisialisasi h dan c, *attention mechanism* menggunakan gating vector f_beta, *output layer* fc untuk memetakan *hidden state* menjadi distribusi token *output*.

Pelatihan dilakukan selama maksimal 20 *epoch* dengan *early stopping* jika validasi tidak membaik dalam 3 *epoch* berturut-turut. Optimasi dilakukan menggunakan Adam, dengan parameter *learning rate* = 4e-4, *batch size* = 32, *dropout* = 0.5, dan *loss* menggunakan CrossEntropyLoss (dengan *masking* <pad> token). Hasil *training* menggunakan parameter tersebut dapat dilihat pada Tabel 4.3.

Tabel 4.3 Hasil *Pretraining* Model CNN-LSTM

Epoch	***Train Loss***	***Val Loss***	**Nama Model**
1	3.1473	2.7093	pretrain_coco_food_epoch1.pth
2	2.6012	2.5696	pretrain_coco_food_epoch2.pth
3	2.3902	2.5197	pretrain_coco_food_epoch3.pth
4	**2.2320**	**2.5086**	**pretrain_coco_food_epoch4.pth**
5	2.0992	2.5347	*No improvement*
6	1.9774	2.5529	*No improvement*
7	1.8676	2.5786	*No improvement*
—	—	—	*Early stopping*

Tabel 4.3 Menunjukan hasil evaluasi per *epoch* saat melakukan proses *training* model CNN-LSTM yang digunakan untuk *pretraining* pada tahap selanjutnya. Model *pretraining* yang digunakan pada penelitian ini berada pada model dengan *checkpoint epoch* ke-4. *Checkpoint* tersebut digunakan, karena pada *epoch* selanjutnya *validation loss* terus naik yang dapat menjadi indikasi model mengalami *overfitting* ringan. Karena tidak ada improvisasi *validation loss* pada 3 *epoch* selanjutnya, maka dilakukan *early stopping* untuk mendapatkan model terbaiknya.

Setelah model selesai dilatih, dilakukan proses inferensi terhadap seluruh *test set* sebanyak 11.630 *captions* untuk menghasilkan deskripsi otomatis dari gambar makanan. *Caption* hasil prediksi dievaluasi menggunakan metrik BLEU terhadap *ground truth caption*. Hasil evaluasi model *pretrain* CNN-LSTM dapat dilihat pada Tabel 4.4.

Tabel 4.4 Hasil Evaluasi *Caption* (BLEU *Score*)

Metrik	Skor
BLEU-1	0.6251
BLEU-2	0.4557
BLEU-3	0.3291
BLEU-4	0.2346

Nilai BLEU-1 hingga BLEU-4 menunjukkan performa yang baik untuk tahap *pretraining*. Model mampu mengenali pola bahasa deskriptif makanan dari gambar secara umum, menghasilkan *caption* dengan struktur yang mendekati ground truth. Skor BLEU-4 sebesar 0.2346 sudah cukup representatif untuk *baseline* awal, dan akan ditingkatkan lebih lanjut melalui proses *fine-tuning* berdasarkan aspek estetika di tahap berikutnya.

Setelah dilakukan proses evaluasi pada model *pretraining*, dilakukan percobaan inferensi dengan menggunakan gambar yang ada pada *subset test*. *Subset* tersebut tidak digunakan dalam proses *training* model *pretrain* ini, sehingga cocok untuk mengukur kemampuan inferensi model hingga proses penelitian ini selesai. Hasil inferensi dapat dilihat pada Tabel 4.5.

Tabel 4.5 Hasil Inferensi Model *Pretraining*

Gambar	***Caption***
000000540508.jpg 	*a sandwich with meat and vegetables on a plate*
000000438788.jpg 	*a cake with a slice of cake on a plate*
000000153909.jpg 	*a table with a plate of food and a cup of coffee*

Gambar	*Caption*
000000277998.jpg 	*a plate of food with a fork and a cup of coffee*
000000552352.jpg 	*a piece of cake is sitting on a plate*

Hasil inferensi yang dilakukan oleh model *pretraining* sudah sesuai dan mudah dimengerti oleh manusia. Hasil inferensi tersebut juga relevan dengan topik utama yang mengenal kata kunci makanan. Hal ini menunjukan model sudah cukup relevan untuk dilakukan *fine-tuning* dengan *dataset* yang akan dibuat pada subbab 4.1.3. Nama-nama gambar tersebut kedepannya akan diguakan sebagai hasil pada subbab 4.1.6 dan 4.1.9.

4.1.3 Pembuatan *Dataset Fine-tune* IAC

Dalam tahap awal pembuatan *dataset fine-tune* IAC, komentar-komentar mentah yang berasal dari situs dpchallenge.com diekstraksi sebagai *caption* awal untuk masing-masing gambar makanan. Komentar-komentar ini mencerminkan penilaian estetika pengguna, tetapi sering kali memiliki format yang tidak seragam seperti penggunaan tanda baca berlebihan, singkatan informal, dan frasa tidak relevan terhadap kualitas visual gambar. Untuk meningkatkan konsistensi dan keterbacaan data, dilakukan proses *cleaning* yang menghapus karakter non-alfabetik, menormalisasi huruf kapital, serta menghilangkan kata-kata yang tidak informatif. Proses ini menghasilkan *caption* bersih yang siap digunakan dalam pelatihan model estetika *caption generator*. Perbandingan antara *caption* mentah dan *caption* bersih dapat dilihat pada Tabel 4.6.

Tabel 4.6 Contoh *Caption* Mentah dan *Caption* Bersih

***Image* ID**	***Image***	***Caption* Mentah (Contoh)**	***Caption* Bersih (Hasil)**
440464.jpg		*The way the wrapper has been torn nicely draws the eye back and I like the shallow DOF. Highlights are too blown out, however.*	*the way the wrapper has been torn nicely draws the eye back and i like the shallow dof highlights are too blown out however*

Image ID	*Image*	*Caption* Mentah (Contoh)	*Caption* Bersih (Hasil)
124525.jpg		*I love the exposure, and the color you chose for the duotone. Beautiful!*	*electrical yes electronicdoesnt seem so much low techdefinitely sepia toning greatly enhances the aging of this image and it is wellframed but seems too busy...*
343630.jpg		*The idea is there but the focus isn't... if you put your camera on a tripod or something the image would be more clear.*	*the idea is there but the focus isnt i though if you put your camera on a tripod or somthing the image wougld be more clear*
525067.jpg		*Interesting to see the differences inside the pepper slices – another play on "same but different"...*	*interesting to see the differences inside the pepper slices another play on same but different the focus is not entirely sharp...*
755550.jpg		*Thanks everyone, for your comments. As someone once said: "I suck at Photoshop!"*	*thanks everyone for your comments as someone once said i suck at photoshop i tried to correct the color issue in ps elements...*

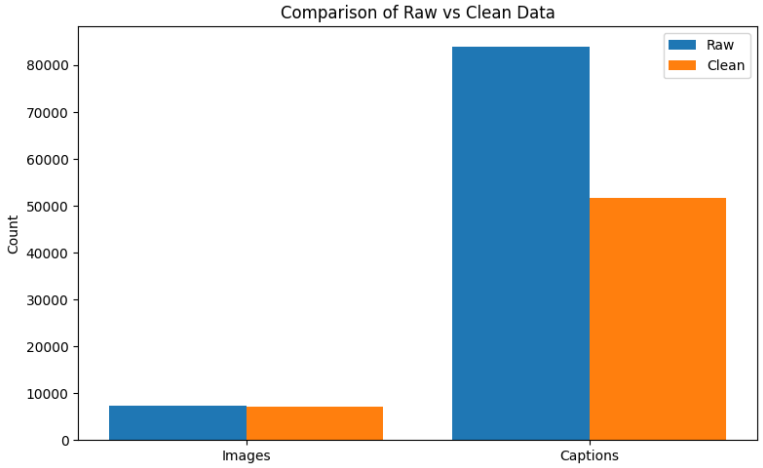
Caption mentah yang diambil dari komentar pengguna di situs dpchallenge.com sering kali mengandung opini pribadi yang beragam dan tidak selalu relevan secara langsung terhadap aspek visual. *Caption* bersih memberikan struktur linguistik yang lebih konsisten untuk pelatihan LDA dan *fine-tuning* CNN-LSTM. Pemrosesan ini tidak menghilangkan makna utama, melainkan menormalisasi konten untuk tujuan NLP dan pembelajaran mesin. Seluruh *caption* yang telah dibersihkan akan disimpan di dalam *file* clean.json.

Sebelum dilakukan proses pembersihan data, *dataset* awal (*raw dataset*) terdiri atas 7.318 gambar dan 84.050 komentar mentah dari pengguna. Setiap komentar ini ditulis dalam gaya bahasa bebas yang tidak seragam, termasuk ejaan yang keliru, tanda baca yang tidak konsisten, dan berbagai simbol tidak relevan. Setelah melalui proses *cleaning* yang mencakup normalisasi teks, penghapusan karakter non-alfabetik, serta filtrasi berdasarkan kualitas informatif *caption*,

jumlah data mengalami penyusutan. Berikut adalah perbandingan kuantitatif antara data mentah dan data bersih yang ditampilkan pada Tabel 4.7.

Tabel 4.7 Perbandingan Jumlah Data *Caption* Mentah dan Bersih

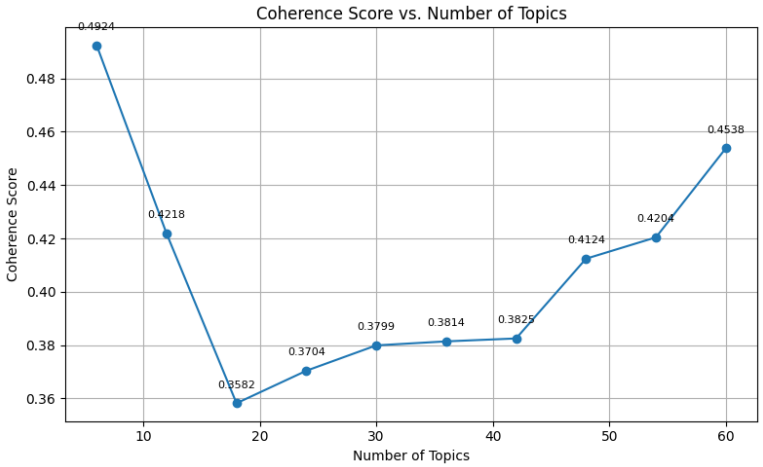
Jenis Data	Jumlah Data Mentah	Jumlah Data Bersih
Jumlah Gambar	7.318	7.221
Jumlah Caption	84.050	51.703



Gambar 4.3 Perbandingan Jumlah Data *Caption* Mentah dan Bersih

Berdasarkan Tabel 4.7 dan Gambar 4.3, sebanyak 96 gambar tidak lolos dalam proses pembersihan karena *caption* yang menyertainya tidak memenuhi kriteria kualitas, seperti *caption* kosong atau tidak relevan. Sedangkan untuk *caption*, hanya sekitar 61.5% dari total *caption* awal yang dianggap layak untuk diproses lebih lanjut. Hal ini menunjukkan bahwa proses pembersihan sangat penting untuk meningkatkan kualitas data dan mencegah model belajar dari input yang ambigu atau tidak informatif.

Setelah *dataset caption* bersih berhasil disiapkan, langkah berikutnya adalah melakukan ekstraksi topik menggunakan model *Latent Dirichlet Allocation* (LDA). Tujuan dari proses ini adalah untuk menemukan struktur tematik tersembunyi dari kumpulan *caption* yang telah dinormalisasi, sehingga setiap *caption* dapat dipetakan ke dalam salah satu dari enam aspek estetika utama yang relevan dengan kualitas visual gambar. Seluruh *caption* bersih yang terdapat dalam clean.json dibaca dan diproses menggunakan metode tokenisasi berbasis *n*-gram. Setiap *caption* dikonversi menjadi kombinasi *unigram* dan *bigram*, lalu digabungkan ke dalam format teks final untuk membentuk korpus dokumen. Proses ini dilakukan melalui fungsi get_all_ngrams() dan create_corpus() yang memastikan bahwa struktur bahasa dari *caption* tetap dipertahankan dalam bentuk token yang informatif. Setelah korpus teks terbentuk, dilakukan representasi numerik menggunakan *dictionary* dan *bag-of-words matrix* dengan pustaka Gensim. Kata-kata yang terlalu jarang (muncul di kurang dari 30 dokumen) atau terlalu umum (muncul di lebih dari 10% dokumen) dihapus untuk meningkatkan kualitas topik yang akan diekstraksi. Objek dictionary dan matriks dokumen-kata ini kemudian disimpan sebagai file lda_dictionary.dict dan lda_corpus.mm di direktori lda_output.



Gambar 4.4 Grafik *Coherence Score Training* LDA Model

Model LDA dilatih menggunakan LdaMulticore dari Gensim, dengan variasi jumlah topik dari 6 hingga 60 dalam interval kelipatan 6. Setiap model dievaluasi menggunakan *coherence score* bertipe c_v, yang lebih sesuai untuk teks pendek seperti *caption*. Skor koherensi ini dicatat pada *file* lda_coherence_summary.txt untuk dibandingkan. Hasil pelatihan menunjukkan bahwa model dengan 60 topik memberikan nilai koherensi terbaik, yaitu sebesar 0.4538, sebagaimana ditampilkan pada Gambar 4.4.

```
0
 0.303*"detail" + 0.084*"place" + 0.071*"beer" + 0.057*"really nice" + 0.055*"product" + 0.039*"backdrop" + 0.034*"shot nice" + 0.027*"nice detail" + 0.027*"cookie"
+ 0.026*"brown" + 0.026*"nice shot" + 0.025*"great detail" + 0.018*"board" + 0.018*"product shot" + 0.012*"summer" + 0.011*"add" + 0.011*"fridge" + 0.011*"close" +
0.010*"direction" + 0.010*"great setup"

1
 0.082*"youve" + 0.081*"drink" + 0.080*"reason" + 0.077*"stock" + 0.077*"bit harsh" + 0.056*"photography" + 0.051*"topic" + 0.050*"lighting bit" + 0.043*"framing" +
0.039*"honey" + 0.028*"variety" + 0.024*"stock photo" + 0.024*"tree" + 0.022*"youd" + 0.021*"treatment" + 0.020*"simple effective" + 0.017*"pencil" + 0.016*"entry c
hallenge" + 0.014*"cap" + 0.014*"meaning"

2
 0.285*"lot" + 0.099*"entry" + 0.088*"interest" + 0.074*"element" + 0.059*"effort" + 0.050*"thought" + 0.040*"point" + 0.037*"others" + 0.033*"story" + 0.020*"chops
tick" + 0.020*"extra point" + 0.019*"mug" + 0.018*"nice contrast" + 0.015*"etc" + 0.013*"darkness" + 0.013*"container" + 0.013*"bet" + 0.012*"date" + 0.011*"idea sh
ot" + 0.011*"color detail"

3
 0.151*"clarity" + 0.120*"piece" + 0.090*"processing" + 0.081*"placement" + 0.071*"flower" + 0.067*"post" + 0.060*"paper" + 0.042*"window" + 0.031*"post processing"
+ 0.027*"softness" + 0.023*"ingredient" + 0.020*"hole" + 0.016*"positioning" + 0.014*"nice clarity" + 0.013*"trading" + 0.013*"trading post" + 0.013*"beach" + 0.012
*"editing" + 0.012*"game" + 0.012*"cloud"

4
 0.555*"challenge" + 0.060*"take" + 0.058*"quality" + 0.031*"take challenge" + 0.030*"great image" + 0.027*"color great" + 0.024*"cheese" + 0.020*"approach" + 0.014
*"screen" + 0.012*"image challenge" + 0.011*"nice take" + 0.011*"fish" + 0.010*"cheer" + 0.010*"beautiful shot" + 0.009*"challenge good" + 0.009*"challenge nice" +
0.009*"challenge great" + 0.008*"well challenge" + 0.007*"spice" + 0.007*"creative take"

5
 0.285*"crop" + 0.093*"macro" + 0.064*"black background" + 0.060*"tad" + 0.057*"feeling" + 0.057*"abstract" + 0.040*"wish" + 0.038*"tight crop" + 0.032*"interpretat
ion" + 0.024*"cropping" + 0.022*"rock" + 0.016*"hehe" + 0.015*"perhaps little" + 0.014*"crisp clean" + 0.013*"macro shot" + 0.013*"foot" + 0.012*"lettuce" + 0.011
*"reduction" + 0.010*"tc" + 0.010*"good macro"

6
 0.542*"composition" + 0.075*"nice composition" + 0.051*"good composition" + 0.024*"color composition" + 0.023*"appeal" + 0.023*"composition good" + 0.021*"mark" +
0.021*"composition color" + 0.016*"composition nice" + 0.016*"composition lighting" + 0.014*"composition great" + 0.013*"shade" + 0.009*"imagination" + 0.008*"color
lighting" + 0.008*"excellent composition" + 0.008*"average" + 0.007*"exposure" + 0.007*"woman" + 0.006*"simple composition" + 0.006*"high mark"

7
 0.402*"job" + 0.178*"good job" + 0.149*"great job" + 0.079*"sense" + 0.044*"humor" + 0.019*"mine" + 0.016*"excellent job" + 0.014*"wow great" + 0.014*"picture goo
d" + 0.011*"laugh" + 0.011*"way" + 0.010*"sense humor" + 0.010*"picture great" + 0.008*"job lighting" + 0.006*"shot good" + 0.006*"well good" + 0.004*"thing" + 0.00
3*"shot great" + 0.003*"eye" + 0.003*"expression"
```

Gambar 4.5 *Output* Distribusi Topik Pelatihan Model LDA

Setiap topik yang dihasilkan memuat 20 kata dominan yang merepresentasikan tema semantik dari kelompok caption tertentu seperti yang ditampilkan pada Gambar 4.5. Daftar topik ini disimpan dalam file lda_topics_60.txt, sementara model terbaik disimpan sebagai lda_topics_60.model. Model ini kemudian dimuat ulang dan divisualisasikan untuk mendukung proses *mapping caption* ke dalam enam aspek utama yaitu; *color & lighting*, *composition*, *depth of field & focus*, *subject*, *use of camera*, dan *general impression*. Proses mapping dilakukan secara manual dengan mengaitkan masing-masing ID topik ke aspek estetika tertentu

berdasarkan kata-kata dominan yang muncul. Hasil dari proses *mapping* dapat dilihat lebih jelas dan detil pada **Lampiran 2**.

Setelah model LDA selesai dilatih, langkah selanjutnya adalah melakukan *mapping* atau pemetaan seluruh *caption* yang telah dibersihkan ke dalam enam aspek estetika utama. Proses ini dimulai dengan memuat model LDA hasil pelatihan sebelumnya beserta kamus token (*Dictionary*) yang telah difilter. Setiap *caption* kemudian ditransformasikan menjadi representasi *bag-of-words* berdasarkan *unigram* dan *bigram* yang sesuai dengan indeks kamus.

Caption yang berhasil dipetakan ke kamus akan dievaluasi menggunakan distribusi topik dari model LDA untuk menentukan topik dengan probabilitas tertinggi. Setiap topik kemudian dipetakan ke salah satu dari enam aspek estetika berdasarkan topic_map yaitu; color_light, subject, composition, dof_and_focus, general_impression, dan use_of_camera. Jika sebuah *caption* tidak memiliki *n*-gram yang cocok dalam kamus (misalnya karena tokenisasi buruk atau kata yang terlalu langka), maka *caption* tersebut tidak dipetakan dan dikeluarkan dari proses lebih lanjut. Distribusi jumlah topik untuk tiap aspek ditampilkan pada Tabel 4.8.

Tabel 4.8 Distribusi Topik dalam Setiap Aspek Estetika

Aspek Estetika	Jumlah Topik	ID Topik
Subject	6	0, 9, 16, 26, 56, 59
Color & Lighting	16	1, 8, 12, 15, 24, 27, 30, 37, 38, 40, 44, 47, 50, 52, 55, 57
Composition	13	2, 6, 14, 18, 21, 28, 29, 32, 36, 43, 48, 51, 53
Depth of Field & Focus	11	3, 5, 17, 22, 23, 25, 33, 34, 35, 46, 54
General Impression	13	4, 7, 11, 13, 19, 20, 31, 39, 41, 42, 45, 49, 58
Use of Camera	1	10

Sebanyak 50.625 *caption* berhasil dipetakan dan disimpan dalam enam berkas JSON terpisah, masing-masing mewakili satu aspek estetika yaitu; color_light.json, subject.json, composition.json, dof_and_focus.json, general_impression.json, dan use_of_camera.json. Rangkuman jumlah *caption* yang ter-*mapping* dalam tiap aspek disimpan dalam berkas aspect_summary.json. Distribusi jumlah *caption* per aspek estetika ditunjukkan pada Tabel 4.9.

Tabel 4.9 Distribusi *Caption* per Aspek Estetika

Aspek Estetika	Jumlah Caption
Color & Lighting	10.660
Subject	4.813
Composition	10.588
DOF & Focus	9.119
General Impression	14.862
Use of Camera	583
Total	**50.625**

Untuk mendapatkan gambaran lebih mendalam terhadap hasil *mapping* yang disebutkan pada Tabel 4.9, dilakukan analisis terhadap lima gambar yang dipilih secara representatif, yaitu 440464.jpg, 124525.jpg, 343630.jpg, 525067.jpg, dan 755550.jpg. Setiap *caption* yang

menyertai gambar-gambar ini telah dipetakan ke dalam aspek estetika yang relevan berdasarkan distribusi probabilistik topik dari model LDA. Tabel 4.10 membantu mengilustrasikan bagaimana satu gambar dapat mengandung beragam komentar yang masing-masing merefleksikan dimensi estetika yang berbeda.

Tabel 4.10 Hasil *Mapping Caption* Per Gambar

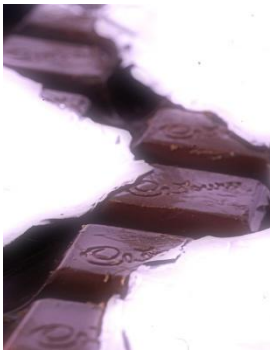
Image ID	***Image***	***Aspect***	***Captions***
440464.jpg		*Color & Lighting*	*my favorite in the chocolate shotsand cadbury to boot*
			seems like the highlights are oversaturated and some chromatic abberations are present
			the blown highlights ruin this one for me hard to look at
		Depth of Field & Focus	*the way the wrapper has been torn nicely draws the eye back and i like the shallow dof highlights are too blown out however*
		General Impression	*the white looks a little glarey to me but i like the idea*
			good idea but the foil just makes it too contrasty to shoot for minimal editing if you could layer it would work
			good idea i think the whole picture needs toned down to get some detail back and the focus may not be quite there still a good idea though
			not a bad idea but the lighting
			i like the creative portion of this photo just wish there was more of the photo that had a sharp focus
124525.jpg		*Color & Lighting*	*nice use of color a bit blown out in spots good job*
		Composition	*electrical yes electronic doesnt seem so much low tech definitely sepia toning greatly enhances the aging of this image and it is well framed but seems too busy the butter on the lower corner and bagel bread on the right center are way overexposed as is the upper left corner perhaps tho this is something akind to high key antique style compared to the dark rich tone in the toaster it seems a bit out of place clean out some of the distractions and simplify the whole to create a real impact but overall a good presentation of the challenge*

Image ID	*Image*	*Aspect*	*Captions*
			great choice of subject however this photo looks a little too busy for me also the upper left corner is very very bright
		Depth of Field & Focus	*excellent duotone but seems to suffer from compression and many areas are overexposed also it may have been mentioned a million times already about the toaster being electrical so i wont go there*
			this image is just wacky and funny
		General Impression	*nicely framed great example of low tech*
343630.jpg		*Subject*	*nice attempt however the background creases in the fabric take away from the image also it is not well focused should have left more of the top of the glass*
		Depth of Field & Focus	*it would be much better if the focus were more crisp*
		General Impression	*cool idea but think it could be executed a little better better lighting background composition etc*
			submission link this will help you get the best score you can and not lose points for having too small an image
			doesnt look to be in focus and there is a lot of noise grain in the photo better lighting might help
		Use of Camera	*the idea is there but the focus isnt i though if you put your camera on a tripod or something the image would be more clear*
525067.jpg		*Color & Lighting*	*interesting to see the differences inside the pepper slices another play on same but different the focus is not entirely sharp and the lighting is rather harsh with distracting shadows if your background were very bright white or entirely dark it would make the peppers stand out better*
			the lighting is a little bright from the front looks like a camera flash perhaps maybe using a side light as well would help
		Subject	*nice and sharp but the lighting is a bit harsh and the subject a little boring*
755550.jpg		*Color & Lighting*	*sounds yummy looks a bit unsharp and messy*

Image ID	*Image*	*Aspect*	*Captions*
			*a bit to beige in colora different colored plate looks good although*
		Subject	*im not sure i care for how this is lit at all bluish light from the frontright and the crop feels very tight i like the dof on the salad in the back though*
		Depth of Field & Focus	*seems underrated to me i guess i can see some color issues now but hey id still eat it*
		General Impression	*thanks everyone for your comments as someone once said i suck at photoshop i tried to correct the color issue in ps elements but it didnt turn out as well i thought the color of the background here at dpc made it look worse than what i saw in elements which is black plus the dnmc police were vicious oh well maybe ill get better at pp and also make sure it meets the description as well as the challenge title*
			the lighting on a lot of these photos this one included is making the food look very unappealing

Sebagai contoh yang disebutkan pada Tabel 4.10, gambar 440464.jpg memiliki total sembilan *caption* yang terbagi ke dalam tiga aspek utama*; Color & Lighting, DOF & Focus, dan General Impression*. *Caption-caption* ini menyoroti pencahayaan yang terlalu terang (highlight), ketajaman gambar, serta kesan keseluruhan dari komposisi. Hal serupa juga ditemukan pada gambar 124525.jpg, di mana *caption* ter-*mapping* ke empat *aspek Color & Lighting, Composition, DOF & Focus,* dan *General Impression*. Komentar-komentar tersebut mengangkat isu-isu estetika seperti kelebihan eksposur, tata letak visual yang terlalu padat, serta penggunaan warna yang tidak seimbang. Gambar 343630.jpg memiliki distribusi *caption* yang bahkan lebih luas, mencakup lima aspek estetika sekaligus termasuk *Use of Camera*, yang menunjukkan bahwa penilai juga memperhatikan aspek teknis seperti penggunaan *tripod* dan pengaturan fokus. Sementara itu, gambar 525067.jpg lebih banyak dikomentari dalam aspek *Color & Lighting dan Subject* yang menyoroti kualitas pencahayaan dan kejernihan objek utama dalam foto. Gambar 755550.jpg juga menunjukkan penyebaran *caption* yang merata ke berbagai aspek, dengan catatan tambahan pada kekurangan dari sisi pencahayaan yang dianggap mengurangi daya tarik makanan yang ditampilkan. Melalui analisis ini, dapat disimpulkan bahwa pendekatan *mapping* berbasis topik menggunakan model LDA berhasil mengelompokkan komentar-komentar pengguna ke dalam aspek estetika yang bermakna. Hasil ini menunjukkan bahwa persepsi estetika terhadap sebuah gambar bersifat multidimensional dan subjektif, serta bahwa setiap *caption* dapat mencerminkan fokus perhatian yang berbeda tergantung pada aspek yang diamati oleh penilai.

Setelah proses *mapping caption* ke dalam enam aspek estetika selesai dilakukan, langkah selanjutnya adalah menyusun *dataset* hasil *mapping* tersebut menjadi format yang sesuai untuk proses pelatihan model. Penyusunan ini mencakup dua tahap penting. Pertama, setiap *file* JSON

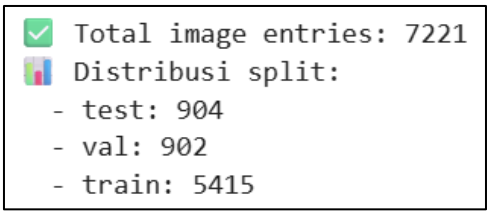
hasil pemetaan (misalnya composition.json, color_light.json, dst.) di-*split* menjadi tiga *subset* menggunakan rasio 6:1:1, yakni 60% data untuk pelatihan (*train*), 20% untuk validasi (*validation*), dan 20% untuk pengujian (*test*). Proses *splitting* dilakukan dengan membangkitkan indeks secara acak menggunakan numpy untuk memastikan distribusi data yang merata dan tidak bias terhadap urutan *input*. Kedua, *caption* yang telah dibersihkan di-*tokenisasi* menggunakan pendekatan berbasis *regular expression tokenizer*, yaitu RegexpTokenizer(r'\w+\S*\w*') dari pustaka NLTK. Setiap *caption* dikonversi menjadi representasi token, yang kemudian disimpan bersamaan dengan *caption raw*-nya. Setiap entri gambar memiliki atribut *filename, url, split*, dan *sentences*, di mana *sentences* berisi daftar *caption* yang telah di-*tokenize*. Hasil splitting penambahan indeks *split* dan token dapat dilihat pada contoh *entry* JSON di Tabel 4.11.

Tabel 4.11 Hasil *Splitting Dataset Fine-tune*


```
  {
    "filename": "755550.jpg",
    "split": "val",
    "url": "https://images.dpchallenge.com/images_challenge/0-
999/979/1200/Copyrighted_Image_Reuse_Prohibited_755550.jpg",
    "sentences": [
      {
        "raw": "thanks everyone for your comments as someone once said i
suck at photoshop i tried to correct the color issue in ps elements but
it didnt turn out as well i thought the color of the background here at
dpc made it look worse than what i saw in elements which is black plus
the dnmc police were vicious oh well maybe ill get better at pp and also
make sure it meets the description as well as the challenge title",
        "tokens": [
          "thanks", "everyone", "for", "your", "comments", "as", "..."
        ]
      },
      {
        "raw": "i can see this shot in any cooking mag should have
finished higher",
        "tokens": [
          "i", "can", "see", "this", "shot", "in","..."
        ]
      },
      {
        "raw": "seems underrated to me i guess i can see some color
issues now but hey id still eat it",
        "tokens": [
          "seems", "underrated", "to", "me", "i", "guess", "..."
        ]
      },
      {
        "raw": "sounds yummy looks a bit unsharp and messy",
        "tokens": [
          "sounds", "yummy", "looks", "a", "bit", "unsharp", "..."
        ]
      },
      {
        "raw": "a bit to beige in colora different colored plate looks
good although",
        "tokens": [
          "a", "bit", "to", "beige", "in", "colora", "..."
        ]
      },
```

```
      {
        "raw": "im not sure i care for how this is lit at all bluish
light from the frontright and the crop feels very tight i like the dof on
the salad in the back though",
        "tokens": [
          "im", "not", "sure", "i", "care", "for", "..."
        ]
      },
      {
        "raw": "the lighting on a lot of these photos this one included
is making the food look very unappealing",
        "tokens": [
          "the", "lighting", "on", "a", "lot", "of", "..."
        ]
      }
    ]
  }
```


Setelah proses pelabelan indeks *split* dan tokenisasi selesai, diperoleh *dataset* gabungan dengan struktur format Karpathy yang mencakup total 7.221 entri gambar, dengan rincian distribusi yang dapat dilihat pada Gambar 4.6.



Gambar 4.6 Hasil *Splitting Dataset Fine-tune*

Validasi dilakukan untuk memastikan integritas *dataset*. Diperoleh 0 gambar tanpa atribut *sentences*, 0 gambar tanpa atribut *split*, dan 0 *caption* dengan format tidak valid (tanpa *raw* atau *tokens*). Dengan demikian, seluruh *dataset* untuk melakukan *fine-tune* telah siap untuk memasuki tahap pra-pemrosesan dan dilanjutkan ke pelatihan model.

4.1.4 Pra-pemrosesan *Dataset Fine-tune* IAC

Setelah proses *labeling* dan tokenisasi *caption* dilakukan, tahap selanjutnya adalah melakukan pra-pemrosesan terhadap *dataset* untuk menyiapkan data dalam format numerik yang dapat digunakan oleh model CNN-LSTM. Proses ini mencakup validasi dan koreksi token terhadap kosakata GloVe, pembentukan *word map*, *encoding caption* ke format numerik, serta penyimpanan data ke dalam format .json dan .hdf5.

Untuk memastikan bahwa seluruh token *caption* berada dalam cakupan kosakata yang dikenali oleh model *embedding*, setiap *token* dicek terhadap GloVe 300D. Bila ditemukan token yang tidak terdapat dalam GloVe, maka token tersebut akan dicoba dikoreksi menggunakan pustaka pyspellchecker, dan dicocokkan dengan kamus koreksi manual pada *file* missed_words_all.json.

4.1.5 *Fine-tuning* dan Evaluasi Model CNN-LSTM untuk Setiap Aspek

Setelah seluruh dataset berhasil dipersiapkan dan dikelompokkan berdasarkan enam aspek estetika, tahap selanjutnya dalam penelitian ini adalah melakukan proses *fine-tuning* terhadap model CNN-LSTM secara terpisah untuk masing-masing aspek. Model yang digunakan terdiri atas dua komponen utama, yaitu *encoder* dan *decoder*. Komponen *encoder* memanfaatkan arsitektur ResNet101 yang telah dipra-latih menggunakan ImageNet. Seluruh lapisan akhir dari

ResNet dipotong, dan hasil representasi spasial dari citra dikompresi menggunakan adaptive average pooling. *Fine-tuning* hanya diaktifkan pada blok-blok akhir dari jaringan ResNet untuk menjaga efisiensi dan menghindari *overfitting*, sementara lapisan awal tetap dibekukan.

Sementara itu, komponen *decoder* berupa LSTM dikembangkan dengan mekanisme inisialisasi *hidden state* berdasarkan keluaran *encoder*. *Decoder* ini menerima *caption* dalam bentuk vektor *embedding* yang berasal dari *pretrained* GloVe 300D. Matriks *embedding* ini dimuat dari sumber eksternal dan hanya lapisan *embedding* yang sesuai yang diaktifkan untuk pelatihan lebih lanjut. *Caption* yang digunakan telah ditokenisasi dan dipetakan ke dalam indeks kosakata berdasarkan hasil *preprocessing* sebelumnya. *Decoder* juga dilengkapi dengan *dropout* untuk menghindari *overfitting* dan fungsi aktivasi linear untuk memetakan keluaran *hidden state* ke ruang dimensi kosakata.

Model CNN-LSTM ini tidak dilatih dari awal, melainkan dimuat dari *checkpoint* hasil *pretraining* pada *dataset* COCO yang sudah difokuskan pada citra makanan pada subbab 4.1.2. Strategi pelatihan dilakukan dengan cara memuat seluruh bobot dari *encoder*, sementara untuk *decoder* hanya lapisan-lapisan yang memiliki kecocokan dimensi dengan arsitektur yang digunakan saat ini yang dimuat, untuk menghindari kesalahan akibat perbedaan ukuran kosakata. Selanjutnya, *fine-tuning* dilakukan secara independen terhadap *subset* data yang telah dipisahkan berdasarkan label aspek estetika. Untuk setiap aspek, data pelatihan dan validasi disediakan secara terpisah, dan pelatihan dilakukan dalam *loop epoch* dengan maksimum 20 iterasi.

Proses *training* dikendalikan dengan fungsi *loss* CrossEntropyLoss, *optimizer* Adam dengan *learning rate* sebesar 1e-4, serta teknik *gradient clipping* sebesar 5.0 untuk menjaga kestabilan pelatihan. Di samping itu, digunakan juga mekanisme *early stopping* dengan batas toleransi stagnasi sebesar tiga *epoch* berturut-turut tanpa peningkatan performa validasi. Selama proses pelatihan, setiap model disimpan ke dalam berkas *checkpoint* dengan nama yang menunjukkan aspek estetika terkait. *Checkpoint* ini hanya diperbarui jika terjadi peningkatan performa validasi pada *epoch* berjalan. Dengan demikian, hasil akhir dari proses *fine-tuning* ini adalah enam buah model CNN-LSTM yang masing-masing telah dioptimalkan untuk menghasilkan *caption* sesuai dengan karakteristik gaya penilaian dari satu aspek estetika tertentu.

Pemilihan *best model* untuk setiap aspek dalam proses *fine-tuning* dilakukan berdasarkan nilai *validation loss* terendah yang dicapai selama pelatihan. Strategi ini mengacu pada pendekatan *early stopping*, di mana proses pelatihan dihentikan secara otomatis apabila dalam beberapa *epoch* berturut-turut tidak terdapat peningkatan performa validasi yang signifikan. Hal ini dilakukan untuk mencegah *overfitting*, yaitu kondisi ketika model terlalu menyesuaikan diri dengan data pelatihan dan kehilangan kemampuan generalisasi terhadap data baru. Hasilnya ditunjukan pada Tabel 4.12 dan Tabel 4.13.

Tabel 4.12 *Train* dan *Val Loss* Aspek *Color Light, Composition,* dan *DOF and Focus*

EPOCH	TRAIN (COLOR & LIGHT)	VAL (COLOR & LIGHT)	TRAIN (COMPOSITION)	VAL (COMPOSITION)	TRAIN (DOF AND FOCUS)	VAL (DOF AND FOCUS)
1	6.8168	5.6112	6.8546	5.6831	6.9238	5.7556
2	5.6089	5.3453	5.6914	5.4317	5.7347	5.5056
3	5.2996	5.1807	5.3943	5.2669	5.4134	5.3362
4	5.0537	5.0743	5.1426	5.1543	5.1458	5.2221

EPOCH	TRAIN (COLOR & LIGHT)	VAL (COLOR & LIGHT)	TRAIN (COMPOSITION)	VAL (COMPOSITION)	TRAIN (DOF AND FOCUS)	VAL (DOF AND FOCUS)
5	4.8385	5.001	4.9381	5.0746	4.9141	5.1502
6	4.6439	4.9546	4.7487	5.0236	4.7087	5.1076
7	4.4705	4.9285	4.5776	4.9948	4.5236	5.0821
8	4.3094	4.9126	4.4153	4.9767	4.3562	5.0674
9	4.1617	4.9106	4.2732	4.9786	4.1991	5.0716
10	**4.0185**	**4.896**	**4.1428**	**4.961**	**4.0545**	**5.0669**
11	3.888	4.9094	4.0109	4.9666	3.9206	5.0856
12	3.7641	4.9449	3.8974	4.997	3.7902	5.0992
13	3.6455	4.9348	3.7866	5.0021	3.6664	5.1

Pada aspek *Color & Light*, model terbaik diperoleh pada *epoch* ke-10 dengan *validation loss* sebesar 4.896. Nilai ini merupakan yang terendah sepanjang pelatihan, menunjukkan bahwa model paling mampu mengeneralisasi data validasi pada titik tersebut. Hal serupa juga terjadi pada aspek *Composition*, di mana *validation loss* terbaik sebesar 4.961 juga tercapai pada *epoch* ke-10. Aspek *DOF and Focus* menunjukkan performa optimal pada *epoch* ke-10 dengan nilai *validation loss* sebesar 5.0669, yang menandakan stabilitas performa model pada area persepsi kedalaman dan ketajaman gambar.

Tabel 4.13 *Train* dan *Val Loss* Aspek *General Impression*, *Subject*, dan *Use of Camera*

EPOCH	TRAIN (GENERAL IMPRESSION)	VAL (GENERAL IMPRESSION)	TRAIN (SUBJECT)	VAL (SUBJECT)	TRAIN (USE OF CAMERA)	VAL (USE OF CAMERA)
1	6.7729	5.596	7.3985	5.9497	9.0442	8.68
2	5.6441	5.3103	5.9123	5.7361	8.3101	7.71
3	5.3401	5.1401	5.6206	5.6022	7.1742	6.591
4	5.1172	5.0258	5.3723	5.5094	6.4245	6.1587
5	4.9266	4.9477	5.1399	5.4439	6.0964	6.004
6	4.7616	4.8924	4.9272	5.4093	5.9303	5.9179
7	4.6072	4.8504	4.7343	5.3854	5.7794	5.8607
8	4.4608	4.8441	4.5499	5.3744	5.6356	5.8153
9	4.3325	4.8148	4.3817	5.3706	5.4823	5.7819
10	4.2103	4.8026	**4.224**	**5.3612**	5.3261	5.7495
11	**4.0937**	**4.7985**	4.0739	5.3973	5.1742	5.7202
12	3.9807	4.8094	3.936	5.3943	5.026	5.7024
13	3.8742	4.8042	3.7987	5.4078	4.8766	5.6868
14	3.7719	4.8031	-	-	4.7259	5.6675
15	-	-	-	-	4.5857	5.6643
16	-	-	-	-	4.4362	5.6541
17	-	-	-	-	4.3023	5.6524
18	-	-	-	-	4.1682	5.6555
19	-	-	-	-	**4.0513**	**5.6507**
20	-	-	-	-	3.9318	5.6576

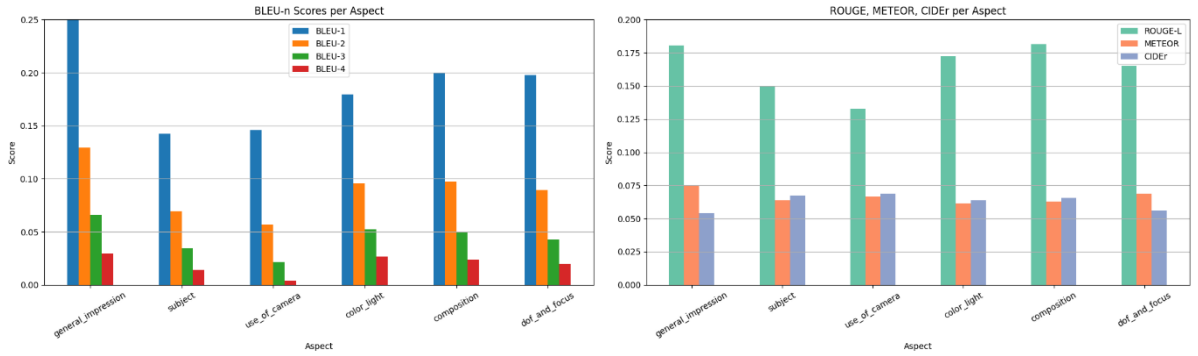
Pada aspek *General Impression*, model terbaik diperoleh pada *epoch* ke-11 dengan nilai *validation loss* sebesar 4.7985. Meskipun perbaikan performa terjadi secara bertahap, tidak ada

peningkatan signifikan setelah *epoch* ini, sehingga model disimpan sebelum memasuki fase *plateau*. Sementara itu, untuk aspek *Subject*, nilai *validation loss* terendah sebesar 5.3612 juga ditemukan pada *epoch* ke-10, menunjukkan bahwa representasi objek utama dalam gambar berhasil dipelajari secara efektif oleh model. Terakhir, aspek *Use of Camera* menunjukkan nilai terendah sebesar 5.6507 pada *epoch* ke-19. Meskipun proses pelatihan dilanjutkan hingga *epoch* ke-20, tidak ditemukan peningkatan performa yang lebih baik, sehingga model dari *epoch* ke-19 dijadikan model final.

Setelah proses *fine-tuning* selesai dilakukan untuk masing-masing aspek estetika, tahap berikutnya adalah evaluasi performa model dalam menghasilkan *caption* menggunakan berbagai metrik kuantitatif. Evaluasi ini bertujuan untuk mengukur seberapa baik model CNN-LSTM yang telah dilatih mampu menghasilkan deskripsi teks yang relevan dan koheren terhadap konten visual dari citra makanan. Pengujian dilakukan pada data *test set* menggunakan tujuh metrik evaluasi umum dalam *image captioning*, yaitu BLEU-1 hingga BLEU-4, ROUGE-L, METEOR, dan CIDEr. Hasil Evaluasi dapat dilihat pada Tabel 4.14 dan Gambar 4.7.

Tabel 4.14 Skor Evaluasi Setiap Model CNN-LSTM

Aspect	BLEU-1	BLEU-2	BLEU-3	BLEU-4	ROUGE-L	METEOR	CIDEr
general_impression	0.276820	0.129268	0.066158	0.029602	0.180745	0.075183	0.054211
subject	0.142690	0.069177	0.034411	0.014066	0.149570	0.063592	0.067344
use_of_camera	0.145745	0.057046	0.021318	0.003602	0.132602	0.066256	0.068833
color_light	0.179265	0.095733	0.052072	0.026829	0.172532	0.061312	0.063925
composition	0.199296	0.097510	0.049556	0.023630	0.181487	0.062752	0.065527
dof_and_focus	0.197832	0.089442	0.042408	0.019603	0.165059	0.068764	0.055871



Gambar 4.7 Grafik Evaluasi Setiap Model CNN-LSTM

Hasil evaluasi menunjukkan bahwa aspek *general_impression* secara konsisten mencetak skor tertinggi pada hampir semua metrik. Aspek ini mencapai skor BLEU-1 sebesar 0.2768, BLEU-2 sebesar 0.1293, dan CIDEr sebesar 0.0542, yang mencerminkan kemampuannya dalam menghasilkan *caption* dengan kesamaan leksikal yang lebih tinggi terhadap referensi manusia. Di sisi lain, aspek *subject* dan *use_of_camera* mencatat skor terendah, khususnya pada metrik BLEU-4 yang masing-masing hanya mencapai 0.0141 dan 0.0036. Hal ini menunjukkan bahwa model masih kesulitan dalam menghasilkan *n*-gram yang panjang dan koheren untuk deskripsi

yang lebih spesifik seperti teknik pengambilan gambar atau identifikasi subjek utama. Aspek *color_light*, *composition*, dan *dof_and_focus* menunjukkan performa yang relatif seimbang dengan skor BLEU-1 berkisar antara 0.1792 hingga 0.1993. Meskipun tidak sekuat *general_impression*, model-model pada aspek ini tetap mampu menghasilkan caption yang cukup relevan dan sesuai konteks visual. Skor METEOR dan ROUGE-L juga memperkuat temuan ini, dengan nilai tertinggi masih didominasi oleh aspek *general_impression* namun diikuti cukup dekat oleh aspek lain seperti *composition* dan *dof_and_focus*. Secara keseluruhan, hasil ini menunjukkan bahwa model CNN-LSTM yang telah ditingkatkan untuk masing-masing aspek estetika mampu menangkap nuansa visual secara bervariasi. Performanya paling optimal pada aspek yang bersifat umum atau subjektif (*general impression*), sementara aspek yang lebih teknis seperti *use of camera* dan *subject* membutuhkan pendekatan tambahan atau arsitektur yang lebih kompleks untuk meningkatkan kualitas *caption* yang dihasilkan. Temuan ini menjadi dasar penting untuk pengembangan tahap selanjutnya, yaitu penggabungan *caption* dari berbagai aspek ke dalam sistem akhir yang lebih holistik.

4.1.6 Inferensi Citra Makanan Dengan 6 Model SAC

Tujuan dari proses ini adalah untuk menghasilkan deskripsi otomatis dari masing-masing aspek; *general impression*, *subject*, *use of camera*, *color & light*, *composition*, serta *depth of field and focus* yang akan digunakan sebagai *input* untuk pelatihan model DAE. Proses inferensi dimulai dengan memuat ulang model CNN-LSTM terbaik dari setiap aspek. Model ini terdiri dari dua komponen utama: *encoder* CNN berbasis *ResNet-101* untuk mengekstraksi fitur visual dari citra, dan *decoder* LSTM yang bertugas menghasilkan *caption* berbasis representasi visual tersebut. Setiap model diinisialisasi dengan bobot terbaik yang diperoleh dari proses pelatihan sebelumnya. *Dataset* yang digunakan untuk proses inferensi berasal dari himpunan data *all.json*, yang terdiri dari 7.221 entri gambar. Masing-masing gambar diproses dengan pipeline transformasi standar (*resize*, *normalize*) dan dikonversi ke format tensor. Setiap gambar kemudian diberi *forward pass* melalui seluruh enam model CNN-LSTM, menghasilkan satu caption untuk tiap aspek.

Tabel 4.15 Contoh Hasil Inferensi Pada *Test Set*

	**Color & Light**	*this is a bit too dark*
	Composition	*nice composition and i like the colors and the background is a little distracting*
	DOF and Focus	*i like the colors and the light is a little soft*
	General Impression	*i like the colors in this photo great job*
	Subject	*this is a nice shot*
	Use of Camera	*all looks on the lighting*

Caption-caption hasil inferensi pada Tabel 4.15 tersebut merupakan hasil dari proses *greedy decoding*, yaitu metode inferensi dengan memilih token dengan probabilitas tertinggi pada setiap langkah waktu tanpa mempertimbangkan reranking. Seluruh hasil *caption* yang dihasilkan dari proses inferensi ini kemudian disimpan dalam sebuah *file* JSON. Format *file* ini terdiri dari entri-entri yang mencakup nama gambar, *caption* per aspek, dan kumpulan *caption* referensi yang tersedia dari anotasi manusia. *Dataset* hasil inferensi ini menjadi dasar bagi tahap selanjutnya dalam pengembangan sistem, yaitu pelatihan model DAE yang bertugas

menyatukan berbagai sudut pandang estetika ke dalam *caption* akhir yang lebih menyeluruh dan bernilai artistik. Keseluruhan *dataset* berjumlah 7.221 gambar, memastikan efisiensi pipeline dan kesiapan data untuk tahapan lanjutan.

4.1.7 Pra-pemrosesan Hasil Inferensi 6 Model SAC

Setelah seluruh gambar berhasil diproses oleh keenam model SAC, setiap gambar menghasilkan enam *caption* dari masing-masing aspek estetika. Tahap pra-pemrosesan dimulai dengan menggabungkan keenam *caption* tersebut menjadi satu kalimat panjang, dengan menyisipkan penanda khusus seperti *tag* "[general_impression]" atau "[composition]" di awal setiap segmen. Hal ini bertujuan untuk memberikan konteks eksplisit terhadap isi *caption* yang dihasilkan, sehingga model DAE dapat mengenali peran masing-masing aspek dalam satu baris *input*. Sebagai target dari model, digunakan salah satu *caption* referensi yang sebelumnya telah diberikan oleh anotator manusia. Baik input maupun target kemudian dikonversi menjadi urutan indeks berdasarkan peta kata yang telah dibentuk pada tahap awal. Setiap kalimat diberi tanda khusus `<start>` dan `<end>` untuk menandai awal dan akhir urutan, serta di-*tokenisasi* menggunakan metode *split* sederhana berdasarkan spasi. Setelah proses tokenisasi selesai, seluruh urutan input dan target dipadatkan ke dalam bentuk tensor dan disamakan panjangnya menggunakan padding berbasis token `<pad>`. Untuk menjaga efisiensi pemrosesan selama pelatihan, dilakukan penyaringan terhadap data dengan cara membatasi panjang maksimum *caption* target. Hanya data dengan panjang target yang tidak melebihi ambang batas tertentu yang dipertahankan, sementara sisanya dikeluarkan dari *dataset*. Langkah ini memastikan bahwa model DAE akan dilatih hanya pada data yang kompak dan relevan, tanpa kehilangan konteks semantik dari *caption-caption* yang telah dihasilkan oleh model SAC.

4.1.8 *Training* dan Evaluasi Model *Denoising Autoencoder* (DAE)

Setelah proses pra-pemrosesan selesai, model *Denoising Autoencoder* dilatih untuk melakukan rekonstruksi *caption* manusia berdasarkan enam *caption* hasil prediksi model SAC dari setiap aspek estetika. Model DAE yang digunakan terdiri dari *embedding layer*, *bidirectional LSTM encoder*, dan *unidirectional LSTM decoder* yang kemudian diakhiri dengan *linear layer* untuk menghasilkan distribusi probabilitas atas seluruh kosakata.

Tabel 4.16 *Training Loss* Model DAE

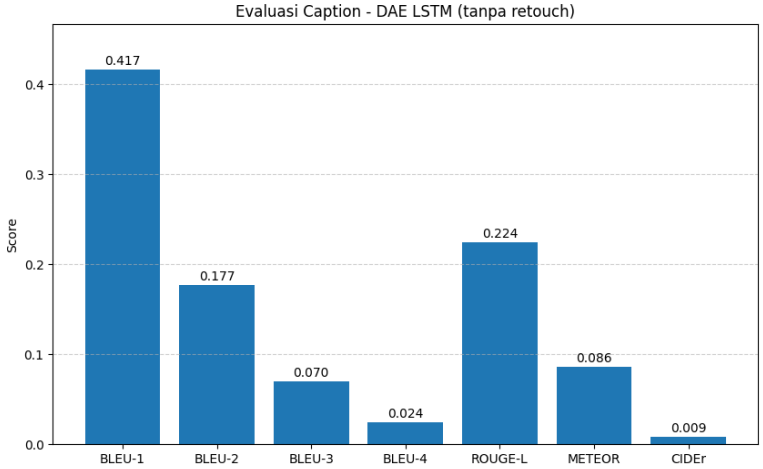
Epoch	Training Loss
1	5.5120
2	4.6987
3	4.2932
4	3.9147
5	3.5086
6	3.0540
7	2.5720
8	2.0944
9	1.6568
10	**1.2930**

Selama pelatihan, pasangan input dan target di-*batch* menggunakan *DataLoader*, dan proses optimisasi dilakukan dengan *Adam optimizer* serta fungsi kerugian *CrossEntropyLoss* yang mengabaikan token *padding*. Model dilatih selama 10 *epoch*, dengan penurunan nilai *loss* yang

konsisten dari awal hingga akhir pelatihan. Model terbaik disimpan berdasarkan nilai *loss* terendah yang diperoleh sebesar 1.2930.

Tabel 4.17 Hasil Evaluasi Metrik Model DAE

Metrik	Skor
BLEU-1	0.4167
BLEU-2	0.1771
BLEU-3	0.0699
BLEU-4	0.0244
ROUGE-L	0.2244
METEOR	0.0862
CIDEr	0.0087



Gambar 4.8 Hasil Evaluasi Metrik Model DAE

Setelah pelatihan selesai, model digunakan untuk menggenerasi *caption* hasil rekonstruksi berdasarkan input dari *caption* multi-aspek. Evaluasi awal dilakukan dengan memilih lima sampel secara acak dari *dataset*, yang memperlihatkan kemampuan model dalam menggabungkan informasi dari seluruh aspek ke dalam satu kalimat utuh yang terdengar natural, meskipun dalam beberapa kasus masih terdapat token `<unk>` akibat keterbatasan kosakata. Evaluasi kuantitatif dilakukan menggunakan metrik standar dalam penilaian kualitas *caption* seperti BLEU, ROUGE-L, METEOR, dan CIDEr. Hasil evaluasi menunjukkan bahwa model mencapai skor BLEU-1 sebesar 0.4167, namun skor menurun secara progresif pada BLEU-4 yang hanya mencapai 0.0244. Hal ini mengindikasikan bahwa meskipun model mampu menangkap sebagian besar kata penting (*unigram*), model tersebut masih menghadapi kesulitan dalam membentuk struktur kalimat yang lengkap dan presisi (*n*-gram lebih tinggi). Skor ROUGE-L dan METEOR masing-masing sebesar 0.2244 dan 0.0862 menunjukkan bahwa model mampu mempertahankan sebagian struktur dan kesamaan semantik dengan caption referensi, meskipun dengan kualitas yang masih dapat ditingkatkan. Skor CIDEr yang rendah (0.0087) menunjukkan bahwa caption yang dihasilkan masih kurang informatif dibandingkan standar komunitas. Metrik SPICE tidak tersedia pada tahap ini.

4.1.9 *Retouch Caption* dengan LLM dan Evaluasi

Setelah proses penggabungan *caption* dari keenam model CNN-LSTM untuk masing-masing aspek, sistem menghasilkan satu *caption* gabungan menggunakan pendekatan *Denoising Autoencoder*. Akan tetapi, karena keterbatasan model dalam menangani generalisasi

linguistik dan kehadiran token <unk> yang belum dikenali, hasil *caption* dari DAE ini sering kali belum cukup alami ataupun utuh dalam menyampaikan keseluruhan informasi yang telah dihasilkan oleh model-model per-aspek. Untuk mengatasi hal tersebut, dilakukan proses penyempurnaan kalimat (*retouch*) dengan bantuan *Large Language Model* (LLM), dalam hal ini menggunakan model GPT-4o-mini melalui antarmuka *chat completion* API.

Proses *retouch* ini diawali dengan menyiapkan kombinasi *caption* dari enam aspek, kemudian dijadikan satu *input sequence* untuk DAE. *Output* dari DAE berupa *caption* awal yang menggabungkan informasi dari seluruh aspek, namun masih dalam bentuk mentah. *Caption* ini kemudian dikirimkan ke LLM bersama dengan input gabungan dari CNN-LSTM, serta gambar opsional, guna mendapatkan versi kalimat yang lebih alami namun tetap menjaga makna inti dan cakupan enam aspek yang telah ditentukan, yakni *general impression*, *subject*, *use of camera*, *color/light*, *composition*, dan *dof/focus*.

Tabel 4.18 Hasil *Caption* Sebelum *Retouch*

	Color & Light	*a bit too dark and the lighting is a bit harsh*
	Composition	*very nice composition and lighting is a little distracting*
	DOF and Focus	*i like the composition and the colors are good focus and the colors are good focus*
	General Impression	*the lighting is a bit too dark and the background is a bit of the subject and the lighting is a bit too dark and the background is a bit more of*
	Subject	*very cool shot i like the glass and the glass is a little more of the glass*
	Use of Camera	*i like the <unk> of the <unk> and something that have been a little of the <unk>*

Sebagai contoh, untuk sebuah gambar makanan, model CNN-LSTM menghasilkan enam *caption* yang terpisah berdasarkan aspek. *Caption-caption* ini kemudian digabung menjadi satu input teks panjang yang menjelaskan keseluruhan kesan dari gambar. DAE menghasilkan kalimat awal seperti "*i love the colors and the* <unk> *texture is all around great job*", yang selanjutnya diperhalus oleh LLM menjadi "*I love the colors and the overall texture, but the lighting is a bit too dark and harsh, the composition is nice yet slightly distracting, and the focus is good, making for a very cool shot of the glass and the subject.*". Kalimat ini berhasil mempertahankan cakupan enam aspek secara eksplisit, memperbaiki kekurangan sintaksis, dan menghilangkan token <unk>. Namun, *caption* awal dari DAE sering kali memiliki keterbatasan dalam hal kelancaran bahasa atau mencakup token <unk> yang tidak terdefinisi. Untuk mengatasi hal ini, sistem menggunakan LLM (dalam hal ini GPT-4o-mini) untuk melakukan *post-editing* berdasarkan prompt instruksi yang ketat. Prompt tersebut secara eksplisit menyatakan

Tabel 4.19 *Prompt Retouch*

```
"Here are combined aspect captions from the CNN-LSTM model (may still
contain '<unk>'):\n
```

```
"{input_dae}"\n\n
And this is the DAE-generated caption (may still contain <unk>):\n
"{caption_dae}"\n\n
Your task:\n

Rewrite the DAE caption to be more natural and fluent, but do NOT change
its structure or core words.\n

IMPORTANT: Ensure the final caption explicitly covers all the key
information/aspects mentioned in the combined CNN-LSTM caption above
(general impression, subject, use of camera, color/light, composition,
dof/focus)—even if briefly, and not necessarily in order.\n

Do not add new information that is not present in the input/DAE captions.
Do not guess objects or details from the image.\n

Fix <unk> if possible, and merge all aspects into one cohesive sentence.\n
Return only the final caption as one simple sentence containing all aspects.
No extra words, no extra creativity."
```

Instruksi (*prompt*) ini memastikan bahwa LLM tidak akan menambahkan interpretasi subjektif atau informasi dari luar konteks input, melainkan hanya memperhalus kalimat yang ada dan memastikan bahwa semua aspek estetika tetap tercakup secara eksplisit. Sebagai contoh, untuk sebuah gambar makanan, model CNN-LSTM menghasilkan enam *caption* yang terpisah berdasarkan aspek. *Caption-caption* ini kemudian digabung menjadi satu input teks panjang yang menjelaskan keseluruhan kesan dari gambar. Tabel 4.20 menunjukan kalimat setelah proses *retouch* berhasil mempertahankan cakupan enam aspek secara eksplisit, memperbaiki kekurangan sintaksis, dan menghilangkan token <unk>.

Tabel 4.20 Perbandingan Hasil DAE Sebelum dan Sesudah *Retouch*

DAE Original	*i love the colors and the <unk> texture is all around great job*
DAE Retouch	*I love the colors and the overall texture, but the lighting is a bit too dark and harsh, the composition is nice yet slightly distracting, and the focus is good, making for a very cool shot of the glass and the subject.*

Proses *testing* hasil *retouch* dimulai dengan menyiapkan *caption DAE* dan *caption gabungan enam aspek*. Masing-masing pasangan *input* digunakan untuk menyusun permintaan (*prompt*) ke API GPT-4o-mini. Dengan menerapkan *prompt engineering* dengan format instruksi yang konsisten, model GPT diharapkan menyempurnakan *caption* menjadi kalimat tunggal yang tetap mencakup seluruh aspek yang dimaksud. Hasil dari proses *retouch* disimpan dalam file .json dengan struktur data berisi ID, *input* DAE, *caption* hasil DAE, dan hasil akhir *retouch*. Setelah seluruh *caption* berhasil diproses dan *diretouch*, evaluasi dilakukan dengan membandingkan hasil tersebut terhadap *ground truth caption* yang sama untuk menguji model DAE *original* (menggunakan *test set* yang sama). Untuk keperluan ini, dilakukan tokenisasi terlebih dahulu menggunakan *tokenizer* dari pustaka pycocoevalcap, yang akan menyesuaikan struktur data untuk semua *caption* menjadi format standar COCO. Setelah itu, *caption* hasil *retouch* dan *ground truth* diselaraskan berdasarkan nama *file* gambar.

Hasil evaluasi pada Tabel 4.21 menunjukkan bahwa *retouch caption* berhasil meningkatkan skor BLEU dan METEOR dibanding hasil evaluasi model DAE *original* pada Tabel 4.17, misalnya dari BLEU-1 sebesar 0.4167 menjadi 0.4323 dan METEOR dari 0.0862 menjadi 0.1314. Hal ini menunjukkan bahwa penyempurnaan dengan GPT-4o-mini berhasil

memperbaiki tata bahasa dan pemilihan kata yang lebih sesuai dengan *ground truth*. Namun demikian, skor CIDEr justru sedikit menurun dari 0.0087 menjadi 0.0074. Penurunan ini dapat dijelaskan karena CIDEr sangat sensitif terhadap koefisien TF-IDF spesifik pada *dataset* referensi, dan *caption* hasil *retouch* yang terdengar lebih generik mungkin kurang sesuai dengan pola TF-IDF yang jarang. Walaupun demikian, secara keseluruhan *caption* yang dihasilkan oleh proses ini menjadi lebih mudah dibaca dan tetap informatif serta memiliki tata bahasa yang lebih baik dibandingkan hasil DAE *original*.

Tabel 4.21 Hasil Evaluasi *Caption Retouch*

Metrik	***Retouch***
BLEU-1	0.4323
BLEU-2	0.2004
BLEU-3	0.0816
BLEU-4	0.0308
ROUGE-L	0.2431
METEOR	0.1314
CIDEr	0.0074

4.2 Pembahasan

Skema pengujian dilakukan terhadap enam aspek estetika secara terpisah menggunakan arsitektur CNN-LSTM, dilanjutkan dengan penggabungan caption menggunakan model DAE (*Denoising Autoencoder*), dan diakhiri dengan pemolesan (*retouch*) melalui *Large Language Model* (LLM). Masing-masing tahap ini berkontribusi terhadap kualitas akhir *caption* yang dihasilkan.

Tabel 4.22 Perbandingan Skor Model CNN-LSTM (Persen)

Aspek	Metode	BLEU-1	BLEU-2	BLEU-3	BLEU-4	ROUGE-L	CIDEr	METEOR
General Impression	CNN-LSTM (ResNet101-2LSTM-Attn) (Zou dkk., 2020)	3.72	3.72	0.00	0.00	3.48	0.07	1.83
	CNN-LSTM (ResNet101+ GloVe+Adam) (Tugas Akhir)	**27.68**	**12.93**	**6.62**	**2.96**	**18.07**	**7.52**	**5.42**
Color & Light	CNN-LSTM (ResNet101-2LSTM-Attn) (Zou dkk., 2020)	1.17	1.17	0.00	0.00	1.63	0.01	1.18
	CNN-LSTM (ResNet101+ GloVe+Adam) (Tugas Akhir)	**17.93**	**9.57**	**5.21**	**2.68**	**17.25**	**6.13**	**6.39**
Compositi-on	CNN-LSTM (ResNet101-2LSTM-Attn) (Zou dkk., 2020)	3.44	3.44	0.00	0.00	3.63	0.03	1.23

Aspek	Metode	BLEU-1	BLEU-2	BLEU-3	BLEU-4	ROUGE-L	CIDEr	METEOR
	CNN-LSTM (ResNet101+ GloVe+Adam) (Tugas Akhir)	**19.93**	**9.75**	**4.96**	**2.36**	**18.15**	**6.28**	**6.55**
DoF & Focus	CNN-LSTM (ResNet101-2LSTM-Attn) (Zou dkk., 2020)	2.68	2.68	0.00	0.00	2.55	0.08	1.24
	CNN-LSTM (ResNet101+ GloVe+Adam) (Tugas Akhir)	**19.78**	**8.94**	**4.24**	**1.96**	**16.51**	**6.88**	**5.59**
Subject	CNN-LSTM (ResNet101+ GloVe+Adam) (Tugas Akhir)	**14.27**	**6.92**	**3.44**	**1.41**	**14.96**	**6.36**	**6.73**
Use of Camera	CNN-LSTM (ResNet101+ GloVe+Adam) (Tugas Akhir)	**14.57**	**5.70**	**2.13**	**0.36**	**13.26**	**6.63**	**6.88**

Hasil evaluasi model CNN-LSTM pada Tabel 4.22 menunjukkan variasi performa tergantung pada aspek estetika yang diuji. Untuk setiap aspek dibandingkan dengan model CNN-LSTM (Zou dkk., 2020) dengan *encoder* ResNet101 dan *decoder* 2LSTM-Attention, model CNN-LSTM dengan *encoder* ResNet-101 dan *decoder* LSTM 512-Dim (GloVe, Adam) pada Tugas Akhir ini mencatat skor BLEU yang lebih tinggi khususnya pada BLEU-1 dan BLEU-2. Sebagai contoh, pada aspek *General Impression* skor BLEU-1 yang diperoleh sebesar 27.68 dibandingkan dengan 3.72 dari baseline CNN-LSTM (Zou dkk., 2020). Hal serupa juga terlihat pada aspek *Color & Light*, *Composition*, *Use of Camera*, serta *DoF & Focus* dengan selisih skor BLEU-1 yang konsisten di atas baseline. Metrik lainnya seperti CIDEr dan ROUGE-L juga memperlihatkan perbedaan. CIDEr tertinggi dalam eksperimen ini mencapai 7.52 pada aspek *General Impression*, melampaui skor CIDEr dari baseline pada aspek yang sama sebesar 0.07. Nilai METEOR dan ROUGE-L dari model ini juga menunjukkan kenaikan dibandingkan baseline pada hampir semua aspek.

Perbedaan performa ini dapat dijelaskan oleh beberapa faktor dan sumber daya. Pertama, dalam Tugas Akhir ini, *vocabulary fine-tuning* yang digunakan dibatasi hanya pada domain *aesthetic captioning*, dan sebelumnya dilakukan *pretraining* dari *dataset* umum berupa MS COCO 2017 yang telah difilter pada *keyword* makanan. Sedangkan model CNN-LSTM yang diusulkan oleh (Zou dkk., 2020) tidak menggunakan pendekatan *transfer learning*, melainkan hanya dilakukan *training* menggunakan *dataset aesthetic captioning*. *Output caption* akhir dari model DAE menunjukkan struktur yang lebih koheren secara internal, meskipun masih ditemukan token seperti <unk> serta struktur kalimat yang belum sepenuhnya optimal. Untuk mengatasi hal ini, proses *retouch* menggunakan LLM telah diterapkan. Hasilnya adalah *caption* yang secara sintaksis lebih stabil dan dapat merepresentasikan keenam aspek estetika yang dimaksud secara eksplisit.

Tabel 4.23 menunjukkan perbandingan skor evaluasi antara model DAE sebelum dan sesudah dilakukannya proses *retouch* menggunakan model GPT-4o-mini. Secara umum, dapat diamati adanya peningkatan skor pada hampir seluruh metrik evaluasi, khususnya pada metrik BLEU dan METEOR. Skor BLEU-1 hingga BLEU-4 meningkat secara konsisten yang

menandakan bahwa hasil *retouch* cenderung memiliki kemiripan *n*-gram yang lebih tinggi dengan *caption* referensi. Peningkatan paling signifikan terjadi pada metrik METEOR, dari 0.0862 menjadi 0.1314, yang mencerminkan adanya perbaikan pada kesesuaian semantik, sinonim, dan struktur gramatikal dalam *caption* hasil *retouch*.

Tabel 4.23 Perbandingan Model DAE dan *Caption Retouch*

Metrik	**DAE *Original***	***Retouch***
BLEU-1	0.4167	**0.4323**
BLEU-2	0.1771	**0.2004**
BLEU-3	0.0699	**0.0816**
BLEU-4	0.0244	**0.0308**
ROUGE-L	0.2244	**0.2431**
METEOR	0.0862	**0.1314**
CIDEr	**0.0087**	0.0074

Selain itu, metrik ROUGE-L juga meningkat dari 0.2244 menjadi 0.2431, yang menunjukkan bahwa *retouch* mampu menghasilkan urutan kata yang lebih sejalan dengan struktur referensi. Hal ini sejalan dengan tujuan *retouch* yang menekankan pada perbaikan kefasihan dan kepaduan antar aspek *caption*. Terdapat sedikit penurunan pada skor CIDEr, dari 0.0087 menjadi 0.0074. Penurunan ini dapat dijelaskan oleh sifat CIDEr yang mengandalkan pembobotan TF-IDF terhadap *n*-gram spesifik dari *caption* referensi. Dalam kasus ini, hasil *retouch* yang lebih terfokus, ringkas, dan natural cenderung menghilangkan *n*-gram yang jarang, sehingga mengurangi kesamaan terhadap *caption* referensi secara statistik, meskipun kualitas semantiknya meningkat. Dengan demikian, meskipun peningkatan skor tidak drastis secara keseluruhan, hasil *retouch* menunjukkan perbaikan yang signifikan dalam aspek keterbacaan dan kesesuaian semantik, yang tidak selalu sepenuhnya tercermin dalam metrik berbasis *n*-gram seperti BLEU dan CIDEr.

(Halaman ini sengaja dikosongkan)

BAB 5
KESIMPULAN DAN SARAN

5.1 Kesimpulan

Penelitian ini bertujuan untuk membangun sebuah sistem *Image Aesthetic Caption Generator* yang mampu menghasilkan deskripsi otomatis terhadap gambar makanan dengan mempertimbangkan enam aspek estetika visual komposisi (*composition*), pencahayaan dan warna (*color & light*), *depth of field* dan fokus, kesan umum (*general impression*), subjek utama (*subject*), serta penggunaan kamera (*use of camera*). Arsitektur sistem terdiri dari tiga tahap utama yaitu model CNN-LSTM untuk menghasilkan *caption* per aspek, model *Denoising Autoencoder* untuk menggabungkan *caption* menjadi satu narasi, dan proses *retouch* menggunakan *Large Language Model* berbasis instruksi untuk menyempurnakan kalimat akhir. Kesimpulan yang dapat ditarik dari penelitian ini adalah sebagai berikut:

1. Model CNN-LSTM berhasil digunakan untuk menghasilkan deskripsi estetika yang relevan dan cukup terperinci terhadap citra makanan berdasarkan enam aspek visual. Model ini menunjukkan peningkatan performa, khususnya pada aspek-aspek yang lebih terukur secara visual seperti *composition* dan *color & light*. Contohnya, pada aspek *composition*, model menunjukkan kenaikan skor BLEU-1 dari 3.44% menjadi 19.93%, serta peningkatan pada METEOR dari 1.23% menjadi 6.55%. Demikian pula, pada aspek *color & light*, skor BLEU-1 meningkat dari 1.17% menjadi 17.93%, dan METEOR dari 1.18% menjadi 6.39%. Model juga menunjukkan peningkatan yang berarti pada aspek lainnya, seperti *depth of field & focus*, di mana BLEU-1 naik dari 2.68% menjadi 19.78%, dan METEOR dari 1.24% menjadi 5.59%. Namun, tantangan masih ditemukan pada aspek yang lebih subjektif dan konseptual seperti *general impression* dan *use of camera*, meskipun tetap terdapat peningkatan skor. Misalnya, BLEU-1 untuk *general impression* naik dari 3.72% menjadi 27.68%, namun metrik METEOR hanya naik ke 5.42%, mengindikasikan bahwa model masih mengalami kesulitan dalam menangkap nuansa semantik dan estetika yang bersifat lebih abstrak.
2. Penambahan dua aspek baru yaitu *subject* dan *use of camera* ke dalam modul *Single-Aspect Captioning* (SAC) memberikan kontribusi penting dalam memperkaya cakupan deskripsi estetika citra makanan. Meskipun skor *caption* pada aspek *use of camera* masih tergolong rendah secara individual dengan BLEU-1 dan BLEU-4 hanya 14.57% dan 0.36%, dan skor evaluasi untuk aspek *subject* hanya 14.27% dan 1.41%. Skor tersebut tergolong rendah dibandingkan dengan keempat aspek estetika lainnya. Kehadiran dua aspek tambahan ini terbukti bermanfaat dalam membentuk narasi gabungan yang lebih utuh dan informatif. Hal ini dibuktikan melalui peningkatan skor evaluasi setelah proses *caption retouch* pada model DAE. Skor CIDEr sedikit menurun dari 0.0087 menjadi 0.0074 dan terjadi peningkatan konsisten pada metrik lain seperti BLEU-1 dari 0.4167 menjadi 0.4323, ROUGE-L dari 0.2244 menjadi 0.2431, dan METEOR (dari 0.0862 menjadi 0.1314). Peningkatan ini menunjukkan bahwa integrasi caption dari semua aspek, termasuk *subject* dan *use of camera* dalam narasi akhir memberikan nilai tambah semantik dan kelengkapan informasi yang lebih baik.
3. Model *Denoising Autoencoder* (DAE) efektif dalam menggabungkan enam *caption* aspek menjadi satu kalimat naratif dengan mempertahankan muatan semantik utama dari setiap aspek. Namun, hasil output DAE masih menunjukkan keterbatasan dari segi kelancaran bahasa dan koherensi sintaksis, yang terlihat dari skor evaluasi yang relatif rendah pada metrik seperti BLEU-4 (0.0244) dan METEOR (0.0862). Proses *caption*

retouch menggunakan *Large Language Model* (GPT-4o-mini) berhasil meningkatkan kualitas deskripsi akhir secara signifikan tanpa menambahkan informasi baru. Hal ini dibuktikan dengan peningkatan skor pada hampir seluruh metrik evaluasi, BLEU-1 naik dari 0.4167 menjadi 0.4323**,** ROUGE-L dari 0.2244 menjadi 0.2431, dan METEOR dari 0.0862 menjadi 0.1314. Meskipun skor CIDEr sedikit menurun dari 0.0087 menjadi 0.0074. Dengan demikian, pendekatan yang menggabungkan model DAE sebagai perangkai semantik dengan LLM sebagai penyempurna sintaksis terbukti efektif dalam menghasilkan *caption* estetika yang utuh, koheren, dan informatif.

5.2 Saran

Berdasarkan pelaksanaan dan evaluasi Tugas Akhir ini, terdapat beberapa saran yang dapat diberikan baik untuk pengembangan sistem lebih lanjut maupun untuk penelitian selanjutnya:

1. Perluasan dan perbaikan *dataset* anotasi aspek. *Dataset* yang digunakan saat ini memiliki keterbatasan dalam jumlah dan kualitas label untuk masing-masing aspek estetika. Untuk hasil yang lebih akurat, sebaiknya digunakan *dataset* yang lebih besar, bervariasi, dan memiliki anotasi yang lebih ketat dari sisi semantik dan estetika.
2. Eksplorasi representasi multimodal. Untuk meningkatkan kualitas caption yang bersifat abstrak seperti *general impression* dan *use of camera*, pendekatan multimodal yang menggabungkan visual features dengan representasi teks atau metadata dapat menjadi strategi yang menjanjikan.
3. Penggunaan *pretrained* model berbasis *vision-language* seperti CLIP atau BLIP dapat menjadi alternatif untuk menggantikan CNN-LSTM klasik yang terbatas dalam pemahaman semantik tingkat tinggi. Model-model ini dapat memahami hubungan antara gambar dan kalimat secara lebih mendalam karena dilatih pada korpus multimodal yang masif.
4. Integrasi proses *retouch* dengan *in-the-loop* learning. Saat ini, proses *retouch* dengan LLM dilakukan secara pasca-pelatihan. Di masa depan, sistem dapat dirancang agar *retouch feedback* dari LLM digunakan untuk melakukan *fine-tuning* ulang terhadap model DAE atau *decoder* LSTM agar model belajar langsung menghasilkan *caption* yang lebih alami.

DAFTAR PUSTAKA

Alzubaidi, L., Zhang, J., Humaidi, A. J., Al-Dujaili, A., Duan, Y., Al-Shamma, O., Santamaría, J., Fadhel, M. A., Al-Amidie, M., & Farhan, L. (2021). Review of deep learning: concepts, CNN architectures, challenges, applications, future directions. *Journal of Big Data*, *8*(1). https://doi.org/10.1186/s40537-021-00444-8

Banerjee, S., & Lavie, A. (t.t.). *METEOR: An Automatic Metric for MT Evaluation with Improved Correlation with Human Judgments*.

Bengio, Y., Yao, L., Alain, G., & Vincent, P. (2013). *Generalized Denoising Auto-Encoders as Generative Models*. http://arxiv.org/abs/1305.6663

Blei, D. M., Ng, A. Y., & Jordan, M. I. (2003). Latent Dirichlet Allocation. Dalam *Journal of Machine Learning Research* (Vol. 3).

Çetin, H. A., Doğan, E., & Tüzün, E. (2021). A review of code reviewer recommendation studies: Challenges and future directions. Dalam *Science of Computer Programming* (Vol. 208). Elsevier B.V. https://doi.org/10.1016/j.scico.2021.102652

Chang, K.-Y., Lu, K.-H., & Chen, C.-S. (t.t.). *Aesthetic Critiques Generation for Photos*.

Chen, P., Guo, Z., Haddow, B., & Heafield, K. (2024). *Iterative Translation Refinement with Large Language Models*.

Das, M., K., S., & Alphonse, P. J. A. (2023). *A Comparative Study on TF-IDF feature Weighting Method and its Analysis using Unstructured Dataset*.

Ding, S., Qu, S., Xi, Y., & Wan, S. (2020). Stimulus-driven and concept-driven analysis for image caption generation. *Neurocomputing*, *398*, 520–530. https://doi.org/10.1016/j.neucom.2019.04.095

Dou, H., Tian, X., Lyu, X., Zhu, J., Li, J., & Guo, L. (2025). *Speech Translation Refinement using Large Language Models*.

Ghosal, K., Rana, A., & Smolic, A. (2019). Aesthetic image captioning from weakly-labelled photographs. *Proceedings - 2019 International Conference on Computer Vision Workshop, ICCVW 2019*, 4550–4560. https://doi.org/10.1109/ICCVW.2019.00556

Grün, F., Rupprecht, C., Navab, N., & Tombari, F. (2016). *A Taxonomy and Library for Visualizing Learned Features in Convolutional Neural Networks*.

Hosna, A., Merry, E., Gyalmo, J., Alom, Z., Aung, Z., & Azim, M. A. (2022). Transfer learning: a friendly introduction. *Journal of Big Data*, *9*(1). https://doi.org/10.1186/s40537-022-00652-w

Jalilifard, A., Caridá, V. F., Mansano, A. F., Cristo, R. S., & da Fonseca, F. P. C. (2021). *Semantic Sensitive TF-IDF to Determine Word Relevance in Documents*. https://doi.org/10.1007/978-981-33-6977-1

Jin, X., Lv, J., Zhou, X., Xiao, C., Li, X., & Zhao, S. (2022). Aesthetic image captioning on the FAE-Captions dataset. *Computers and Electrical Engineering*, *101*. https://doi.org/10.1016/j.compeleceng.2022.107866

Jin, X., Zou, D., Wu, L., Zhao, G., Li, X., Zhang, X., Zhou, B., Ge, S., & Zhou, X. (2019). Aesthetic attributes assessment of images. *MM 2019 - Proceedings of the 27th ACM International Conference on Multimedia*, 311–319. https://doi.org/10.1145/3343031.3350970

Kang, Q., Chen, E. J., Li, Z. C., Luo, H. Bin, & Liu, Y. (2023). Attention-based LSTM predictive model for the attitude and position of shield machine in tunneling. *Underground Space (China)*, *13*, 335–350. https://doi.org/10.1016/j.undsp.2023.05.006

Karpathy, A., & Fei-Fei, L. (2015). *Deep Visual-Semantic Alignments for Generating Image Descriptions*.

Kasneci, E., Sessler, K., Küchemann, S., Bannert, M., Dementieva, D., Fischer, F., Gasser, U., Groh, G., Günnemann, S., Hüllermeier, E., Krusche, S., Kutyniok, G., Michaeli, T., Nerdel, C., Pfeffer, J., Poquet, O., Sailer, M., Schmidt, A., Seidel, T., … Kasneci, G. (2023). ChatGPT for good? On opportunities and challenges of large language models for education. Dalam *Learning and Individual Differences* (Vol. 103). Elsevier Ltd. https://doi.org/10.1016/j.lindif.2023.102274

Ke, J., Ye, K., Yu, J., Wu, Y., Milanfar, P., & Yang, F. (2023). VILA: Learning Image Aesthetics from User Comments with Vision-Language Pretraining. *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, *2023-June*, 10041–10051. https://doi.org/10.1109/CVPR52729.2023.00968

Ko, J., Baek, J., & Hwang, S. J. (2025). *Real-time Verification and Refinement of Language Model Text Generation*.

Lin, C.-Y. (t.t.). *ROUGE: A Package for Automatic Evaluation of Summaries*.

Lin, T.-Y., Maire, M., Belongie, S., Bourdev, L., Girshick, R., Hays, J., Perona, P., Ramanan, D., Zitnick, C. L., & Dollár, P. (2014). *Microsoft COCO: Common Objects in Context*.

Manojkumar, V. K., Mathi, S., & Gao, X. Z. (2022). An Experimental Investigation on Unsupervised Text Summarization for Customer Reviews. *Procedia Computer Science*, *218*, 1692–1701. https://doi.org/10.1016/j.procs.2023.01.147

Mikolov, T., Chen, K., Corrado, G., & Dean, J. (2013). *Efficient Estimation of Word Representations in Vector Space*.

Naveed, H., Khan, A. U., Qiu, S., Saqib, M., Anwar, S., Usman, M., Akhtar, N., Barnes, N., & Mian, A. (2023). *A Comprehensive Overview of Large Language Models*. http://arxiv.org/abs/2307.06435

Oliva, M., Banik, S., Josifovski, J., & Knoll, A. (2022). *Graph Neural Networks for Relational Inductive Bias in Vision-based Deep Reinforcement Learning of Robot Control*. http://arxiv.org/abs/2203.05985

Panigrahi, U., Sahoo, P. K., Panda, M. K., & Panda, G. (2024). A ResNet-101 deep learning framework induced transfer learning strategy for moving object detection. *Image and Vision Computing*, *146*. https://doi.org/10.1016/j.imavis.2024.105021

Papineni, K., Roukos, S., Ward, T., & Zhu, W.-J. (t.t.). *BLEU: a Method for Automatic Evaluation of Machine Translation*.

Pennington, J., Socher, R., & Manning, C. (2014). Glove: Global Vectors for Word Representation. *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 1532–1543. https://doi.org/10.3115/v1/D14-1162

Sasibhooshan, R., Kumaraswamy, S., & Sasidharan, S. (2023). Image caption generation using Visual Attention Prediction and Contextual Spatial Relation Extraction. *Journal of Big Data*, *10*(1). https://doi.org/10.1186/s40537-023-00693-9

Spence, C., Motoki, K., & Petit, O. (2022). Factors influencing the visual deliciousness / eye-appeal of food. Dalam *Food Quality and Preference* (Vol. 102). Elsevier Ltd. https://doi.org/10.1016/j.foodqual.2022.104672

Staudemeyer, R. C., & Morris, E. R. (2019). *Understanding LSTM -- a tutorial into Long Short-Term Memory Recurrent Neural Networks*. http://arxiv.org/abs/1909.09586

Sutskever, I., Vinyals, O., & Le, Q. V. (2014). *Sequence to Sequence Learning with Neural Networks*.

Vedantam, R., Zitnick, C. L., & Parikh, D. (2014). *CIDEr: Consensus-based Image Description Evaluation*. http://arxiv.org/abs/1411.5726

Venkatesh, Hegde, S. U., S, Z. A., & Y, N. (2021). Hybrid CNN-LSTM Model with GloVe Word Vector for Sentiment Analysis on Football Specific Tweets. *2021 International Conference on Advances in Electrical, Computing, Communication and Sustainable Technologies (ICAECT)*, 1–8. https://doi.org/10.1109/ICAECT49130.2021.9392516

Wiseman, S., & Rush, A. M. (2016). *Sequence-to-Sequence Learning as Beam-Search Optimization*.

Xu, G., Wang, X., Wu, X., Leng, X., & Xu, Y. (2024). *Development of Skip Connection in Deep Neural Networks for Computer Vision and Medical Image Analysis: A Survey*.

Yeo, Y. Y., See, J., Wong, L. K., & Goh, H. N. (2021). GENERATING AESTHETIC BASED CRITIQUE FOR PHOTOGRAPHS. *Proceedings - International Conference on Image Processing, ICIP*, *2021-September*, 2523–2527. https://doi.org/10.1109/ICIP42928.2021.9506385

Zou, X., Lin, C., Zhang, Y., & Zhao, Q. (2020). To be an Artist: Automatic Generation on Food Image Aesthetic Captioning. *Proceedings - International Conference on Tools with Artificial Intelligence, ICTAI*, *2020-November*, 779–786. https://doi.org/10.1109/ICTAI50040.2020.00124

(Halaman ini sengaja dikosongkan)

LAMPIRAN

Lampiran 1. Hasil Inferensi Pada *Subset Test*

677727.jpg	Color & Light	*very nice use of color*
	Composition	*very creative and composition is great shot*
	DOF and Focus	*i like the colors and the colors are good focus and the lighting is a little too much*
	General Impression	*great idea and the lighting is a bit too much*
	Subject	*great shot im not sure i like the subject and the subject is great*
	Use of Camera	*abstract having something that you like the <unk> in the challenge of the <unk>*
	DAE Original	*i just love the crops that you do like <unk> as they have been a great shot*
	DAE Retouch	*I just love the crops that you do, as they have been a great shot with nice use of color, very creative composition, and the lighting is a bit too much, though the colors are good focus.*

221317.jpg	Color & Light	*i like the idea but i like the lighting is a bit harsh and the background is a bit too dark*
	Composition	*i like the background is the background is the background is the background is the background*
	DOF and Focus	*its <unk> image is a little too much of the image and the image is very nice image*
	General Impression	*i like the idea but i think this would have been better if you did a great shot*
	Subject	*very nice nice shot*
	Use of Camera	*the is something like the the challenge i think to the challenge of the challenge of the <unk> and the the and the challenge of the i think to the it is*
	DAE Original	*i love the <unk> of the <unk> of the <unk> cool <unk>*
	DAE Retouch	*I love the idea of the very nice shot of the colorful sprinkles on the donut, but the lighting is a bit harsh and the background is a bit too dark, while the image composition is nice but could be improved for focus.*

645515.jpg	Color & Light	*i like the contrast and the lighting is a bit harsh*
	Composition	*a little more of the background is a little distracting*
	DOF and Focus	*i like the lighting and the colors are good focus is a bit distracting but the image is a little too much of the image is a bit distracting*
	General Impression	*i really like this photo the lighting is a bit too much the lighting and the <unk> in the glass and the background is a bit distracting but the lighting is the*
	Subject	*i like the <unk> of the bottle is a little more of the bottle and the bottle is a little too much of the subject*
	Use of Camera	*i like the the of the challenge of the <unk> and the <unk> and the <unk> of the <unk>*
	DAE Original	*i just love the crops that you do like <unk> as they have been a <unk> <unk>*
	DAE Retouch	*I really love the crops you do, as they highlight the beautiful lighting and contrast, but the background is a bit distracting, and while I enjoy the subject of the bottle, the focus and colors could be improved.*

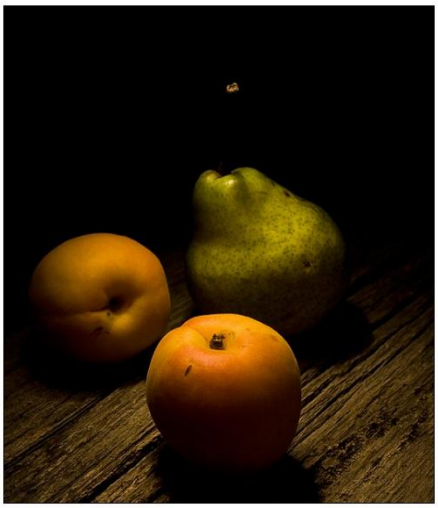
287943.jpg	Color & Light	*i like the lighting and the composition is a bit harsh and the background is a bit harsh*
	Composition	*the background is a little distracting*
	DOF and Focus	*i love the <unk> of the <unk> is a little too much of the image is too much of the image is too much of the image is a little too much*
	General Impression	*nice idea but the lighting is a bit of the <unk>*
	Subject	*i like the colors and the subject is a little more of the glass*
	Use of Camera	*i like the the <unk> and the <unk> and the <unk>*
	DAE Original	*<unk> i love the <unk> of the <unk> <unk> done*
	DAE Retouch	*I love the idea of the ice cream, but the lighting is a bit harsh, and while I like the colors and the glass subject, the composition is somewhat distracting with a background that draws attention away, and the focus feels a little too prominent in this image.*

466510.jpg	Color & Light	*i like the lighting and the lighting is a bit harsh and the lighting is a bit harsh*
	Composition	*beautiful composition and a good composition and the challenge*
	DOF and Focus	*this is a little dark and the background is a little dark*
	General Impression	*great shot i think the idea is the <unk> of the <unk>*
	Subject	*great detail and the lighting is a little more of the subject*
	Use of Camera	*this is something like the the challenge and the <unk> and the <unk>*
	DAE Original	*i like the idea and the lighting looks just a little too bright the <unk> is slightly offputting*
	DAE Retouch	*I like the idea of this great shot with great detail, but the lighting looks just a little too harsh and bright, making the composition and background appear a bit dark and slightly offputting.*

894191.jpg	Color & Light	*very nice shot good lighting and composition*
	Composition	*i like the composition and the composition is good*
	DOF and Focus	*i like the <unk> <unk> <unk> <unk> <unk> <unk> <unk> <unk> <unk> <unk> <unk> <unk> <unk>*
	General Impression	*looks like the lighting is a bit too much to the photo*
	Subject	*i like the <unk> the <unk> is a little too much to be a little more of the glass*
	Use of Camera	*i like the the challenge of the challenge of the challenge and something is something to do it to it it*
	DAE Original	*i love the <unk> of the <unk> of the <unk> of the <unk> cool*
	DAE Retouch	*I love the composition of the photo, though the lighting seems a bit strong, and the use of the camera captures the nice colors of the cupcake and cup, making it a cool shot with good depth of field.*

724149.jpg	Color & Light	*great use of dof*
	Composition	*nice composition and composition is a little distracting*
	DOF and Focus	*i like the idea of the image and the composition is very good*
	General Impression	*i like the idea and the <unk> is <unk> <unk> <unk> <unk> <unk> <unk> <unk> <unk> <unk> <unk> <unk> <unk> <unk>*
	Subject	*im not sure about the <unk> of the <unk>*
	Use of Camera	*this is is something like the the challenge of the challenge of the <unk> of the <unk>*
	DAE Original	*i love the idea and the lighting looks just a little too bright the <unk> is slightly to me*
	DAE Retouch	*I love the idea and while the lighting looks just a little too bright, the subject is interesting, although I'm not sure about the use of camera, and the composition is nice but slightly distracting, with great use of depth of field.*

440322.jpg	Color & Light	*i like the idea of the background and the background is a bit harsh*
	Composition	*nice concept of the background is a little distracting*
	DOF and Focus	*very nice setup and the image is very nice*
	General Impression	*i like the <unk> of the <unk>*
	Subject	*i like the colors and the lighting is a little too much to be a bit more of the subject*
	Use of Camera	*i like the the challenge of the challenge of the <unk> and the <unk> and the a <unk> and the a <unk> and something is something to be a bit of the*
	DAE Original	*i like the idea of casting the glow onto the <unk> wish i could see more of them and the environment*
	DAE Retouch	*I like the idea of casting the glow onto the coffee beans, as the colors and lighting are a bit too much; I wish I could see more of them and the environment, but the setup is nice, though the background is a little harsh and distracting.*

528536.jpg	Color & Light	*very nice shot but the lighting is a bit harsh*
	Composition	*i like the composition and the colors are a little distracting*
	DOF and Focus	*i love the <unk> and <unk> i think this image is a little more light source*
	General Impression	*i like the idea but the lighting is a bit too dark*
	Subject	*nice detail and the subject is a little more of the subject*
	Use of Camera	*this is something something something to the the challenge of the challenge*
	DAE Original	*i love the idea and the lighting looks just a little too close to the right <unk>*
	DAE Retouch	*I love the idea, but the lighting is a bit too dark and harsh, while the composition is nice with some distracting colors; the detail in the subject is good, although the depth of field could be improved to enhance the overall shot.*

805901.jpg	Color & Light	*wow is a bit too dark for my taste*
	Composition	*i like the <unk> of the <unk> and the background is a little distracting*
	DOF and Focus	*great <unk> and <unk>*
	General Impression	*this is a great idea i like the <unk> of the <unk>*
	Subject	*im not sure a <unk> <unk> <unk>*
	Use of Camera	*very you this the the of of of of the and the and the and the challenge of the of of the and the challenge of the challenge of the of of*
	DAE Original	*i love the idea and the lighting looks just a little too close to the right <unk>*
	DAE Retouch	*I love the idea and the lighting is a bit too dark for my taste, while the composition of the subject with the background is a little distracting, but the depth of field and focus are great.*

Lampiran 2. Hasil Pemetaan Topik ke Aspek

ID Topik	Isi Topik	Aspek
0	detail (0.303) + place (0.084) + beer (0.071) + really nice (0.057) + product (0.055) + backdrop (0.039) + shot nice (0.034) + nice detail (0.027) + cookie (0.027) + brown (0.026) + nice shot (0.026) + great detail (0.025) + board (0.018) + product shot (0.018) + summer (0.012) + add (0.011) + fridge (0.011) + close (0.011) + direction (0.010) + great setup (0.010)	subject
1	youve (0.082) + drink (0.081) + reason (0.080) + stock (0.077) + bit harsh (0.077) + photography (0.056) + topic (0.051) + lighting bit (0.050) + framing (0.043) + honey (0.039) + variety (0.028) + stock photo (0.024) + tree (0.024) + youd (0.022) + treatment (0.021) + simple effective (0.020) + pencil (0.017) + entry challenge (0.016) + cap (0.014) + meaning (0.014)	color_light
2	lot (0.285) + entry (0.099) + interest (0.088) + element (0.074) + effort (0.059) + thought (0.050) + point (0.040) + others (0.037) + story (0.033) + chopstick (0.020) + extra point (0.020) + mug (0.019) + nice contrast (0.018) + etc (0.015) + darkness (0.013) + container (0.013) + bet (0.013) + date (0.012) + idea shot (0.011) + color detail (0.011)	composition
3	clarity (0.151) + piece (0.120) + processing (0.090) + placement (0.081) + flower (0.071) + post (0.067) + paper (0.060) + window (0.042) + post processing (0.031) + softness (0.027) + ingredient (0.023) + hole (0.020) + positioning (0.016) + nice clarity (0.014) + trading (0.013) + trading post (0.013) + beach (0.013) + editing (0.012) + game (0.012) + cloud (0.012)	dof_and_focus
4	challenge (0.555) + take (0.060) + quality (0.058) + take challenge (0.031) + great image (0.030) + color great (0.027) + cheese (0.024) + approach (0.020) + screen (0.014) + image challenge (0.012) + nice take (0.011) + fish (0.011) + cheer (0.010) + beautiful shot (0.010) + challenge good (0.009) + challenge nice (0.009) + challenge great (0.009) + well challenge (0.008) + spice (0.007) + creative take (0.007)	general_impression
5	crop (0.285) + macro (0.093) + black background (0.064) + tad (0.060) + feeling (0.057) + abstract (0.057) + wish (0.040) + tight crop (0.038) + interpretation (0.032) + cropping (0.024) + rock (0.022) + hehe (0.016) + perhaps little (0.015) + crisp clean (0.014) + macro shot (0.013) + foot (0.013) + lettuce (0.012) + reduction (0.011) + tc (0.010) + good macro (0.010)	dof_and_focus
6	composition (0.542) + nice composition (0.075) + good composition (0.051) + color composition (0.024) + appeal (0.023) + composition good (0.023) + mark (0.021) + composition color (0.021) + composition nice (0.016) + composition lighting (0.016) + composition great (0.014) + shade (0.013) + imagination (0.009) + color lighting (0.008) + excellent composition (0.008) + average (0.008) + exposure (0.007) + woman (0.007) + simple composition (0.006) + high mark (0.006)	composition
7	job (0.402) + good job (0.178) + great job (0.149) + sense (0.079) + humor (0.044) + mine (0.019) + excellent job (0.016) + wow great (0.014) + picture good (0.014) + laugh (0.011) + way (0.011) + sense humor (0.010) + picture great (0.010) + job lighting (0.008) + shot good (0.006) + well good (0.006) + thing (0.004) + shot great (0.003) + eye (0.003) + expression (0.003)	general_impression
8	texture (0.246) + bit (0.125) + spot (0.102) + rule (0.045) + cake (0.034) + bit much (0.031) + third (0.029) + perfect (0.028) + color texture (0.026) + sure (0.025) + think (0.020) + hot spot (0.019) + rule third (0.019) + difference (0.017) + stick (0.017) + color bit (0.016) + bit light (0.016) + great texture (0.014) + tiny bit (0.013) + bit tight (0.012)	color_light
9	life (0.157) + still life (0.114) + flash (0.106) + nice (0.084) + object (0.077) + crisp (0.061) + leaf (0.054) + world (0.035) + room (0.033) + yes (0.031) + pan (0.019) + house (0.018) + metal (0.017) + sheet (0.016) + sandwich (0.015) + value (0.014) + clean crisp (0.013) + low key (0.012) + see (0.010) + clue (0.009)	subject
10	something (0.373) + camera (0.125) + banana (0.063) + setting (0.032) + glad (0.030) + content (0.028) + hi (0.027) + form (0.024) + tripod (0.024) + blue background (0.018) + aperture (0.017) + thing (0.017) + im glad (0.017) + iso	use_of_camera

	(0.017) + bother (0.016) + representation (0.015) + way (0.011) + something similar (0.010) + something different (0.010) + future challenge (0.009)	
11	kind (0.168) + doesnt (0.153) + wall (0.052) + fact (0.047) + dish (0.040) + help (0.039) + middle (0.036) + beautiful (0.029) + doesnt work (0.026) + studio (0.026) + horizon (0.023) + carrot (0.023) + home (0.022) + study (0.021) + none (0.021) + pity (0.018) + line (0.016) + reminds (0.016) + decision (0.014) + thumbnail (0.013)	general_impression
12	title (0.300) + shape (0.103) + sort (0.062) + saturation (0.060) + type (0.058) + red (0.033) + mind (0.033) + wouldve (0.029) + seed (0.026) + style (0.024) + still good (0.018) + beauty (0.017) + thinking (0.016) + great title (0.016) + type shot (0.015) + punch (0.015) + cool color (0.013) + cover (0.013) + bland (0.012) + excellent color (0.011)	color_light
13	photo (0.627) + nice photo (0.037) + creativity (0.035) + bubble (0.035) + great photo (0.033) + good photo (0.028) + photo good (0.015) + darker (0.014) + yeah (0.014) + photo great (0.012) + back (0.011) + photo nice (0.010) + fond (0.008) + advertising (0.008) + photo little (0.007) + way (0.007) + great pic (0.007) + cool photo (0.006) + tint (0.006) + excellent photo (0.006)	general_impression
14	cool (0.099) + try (0.097) + pattern (0.088) + pop (0.068) + magazine (0.047) + cool shot (0.046) + guy (0.046) + layout (0.032) + pie (0.031) + collection (0.030) + night (0.029) + food (0.029) + friend (0.025) + high contrast (0.022) + distracts (0.022) + chicken (0.020) + bird (0.019) + nice try (0.017) + close crop (0.017) + archive collection (0.016)	composition
15	nothing (0.148) + bw (0.125) + fit (0.048) + alot (0.047) + pear (0.047) + art (0.047) + heart (0.045) + little good (0.037) + interesting (0.037) + improvement (0.032) + complementary color (0.023) + well color (0.022) + wider (0.022) + red green (0.020) + mix (0.020) + photo challenge (0.020) + family (0.019) + garbage (0.017) + criterion (0.017) + fabric (0.016)	color_light
16	glass (0.393) + bottle (0.200) + wine (0.068) + reflection (0.027) + cooky (0.025) + wine glass (0.024) + glow (0.016) + milk (0.016) + reflection glass (0.015) + hmm (0.014) + dark (0.014) + irene (0.013) + dark background (0.013) + eye (0.011) + drip (0.011) + color little (0.009) + shirt (0.009) + good pic (0.008) + id (0.008) + awesome shot (0.008)	subject
17	reflection (0.228) + time (0.228) + photograph (0.083) + dark (0.063) + surface (0.055) + next time (0.039) + little dark (0.035) + circle (0.028) + combination (0.023) + version (0.019) + backround (0.012) + round (0.012) + many time (0.011) + context (0.011) + compostion (0.010) + first time (0.010) + second (0.009) + nice reflection (0.009) + low angle (0.009) + thing (0.009)	dof_and_focus
18	opinion (0.134) + front (0.130) + perspective (0.102) + impact (0.088) + page (0.052) + person (0.045) + description (0.033) + advertisement (0.030) + sharper (0.028) + front page (0.026) + material (0.025) + criticism (0.022) + stop (0.022) + variation (0.019) + interesting shot (0.019) + little sharper (0.017) + challenge description (0.016) + otherwise great (0.012) + centre (0.012) + well opinion (0.012)	composition
19	critique (0.080) + club (0.057) + challenge (0.055) + critique club (0.054) + didnt (0.039) + voter (0.035) + greeting (0.029) + greeting critique (0.027) + issue (0.026) + score (0.020) + comment (0.020) + impression (0.020) + thing (0.020) + composition (0.018) + question (0.018) + well shot (0.016) + technique (0.016) + food shot (0.013) + first impression (0.013) + way (0.012)	general_impression
20	image (0.185) + nice image (0.113) + rest (0.096) + imho (0.058) + result (0.052) + year (0.041) + example (0.040) + technical (0.036) + one (0.033) + funny (0.030) + closeup (0.029) + image nice (0.026) + dinner (0.024) + good one (0.023) + grey (0.023) + phone (0.016) + emphasis (0.016) + fav (0.014) + relevance (0.013) + wonderful shot (0.013)	general_impression
21	fruit (0.294) + composition (0.140) + great composition (0.096) + veggie (0.050) + sign (0.045) + mouth (0.045) + smoke (0.043) + idea composition (0.034) + vegetable (0.032) + star (0.031) + image great (0.031) + fruit nice (0.016) + check (0.013) + vote (0.013) + nice fruit (0.012) + light composition (0.012) + way (0.012) + thing (0.012) + comment (0.010) + number (0.010)	composition
22	one (0.151) + setup (0.137) + effect (0.128) + job (0.116) + nice job (0.115) + simplicity (0.059) + everyone (0.027) + well nice (0.023) + good dof (0.023)	dof_and_focus

	+ nice setup (0.022) + chip (0.018) + nice clear (0.016) + comment everyone (0.014) + desaturation (0.013) + comment (0.013) + warmth (0.012) + everyone one (0.012) + dimension (0.011) + overall nice (0.009) + setup shot (0.009)	
23	center (0.106) + view (0.085) + model (0.062) + yum (0.056) + haha (0.048) + base (0.044) + control (0.033) + great dof (0.032) + point (0.031) + high key (0.028) + key (0.028) + wouldnt (0.027) + order (0.027) + snapshot (0.024) + eye (0.023) + toast (0.022) + mushroom (0.022) + point view (0.020) + ive (0.019) + today (0.017)	dof_and_focus
24	use (0.317) + tomato (0.074) + good use (0.062) + nice use (0.058) + use color (0.042) + bokeh (0.040) + great use (0.037) + number (0.027) + dof (0.025) + use dof (0.024) + billboard (0.019) + use light (0.018) + closer (0.018) + space (0.017) + card (0.016) + use negative (0.016) + negative space (0.014) + letter (0.013) + nut (0.013) + pas (0.011)	color_light
25	focus (0.644) + good focus (0.043) + sharp focus (0.034) + little soft (0.021) + soft focus (0.019) + focus good (0.018) + great focus (0.015) + nice focus (0.014) + little focus (0.013) + focus nice (0.010) + focus little (0.010) + color focus (0.010) + focus color (0.009) + eye (0.009) + focus bit (0.008) + slightly focus (0.008) + focus composition (0.008) + focus great (0.008) + part (0.008) + crisp focus (0.006)	dof_and_focus
26	food (0.265) + work (0.239) + plate (0.178) + great work (0.035) + butter (0.029) + lit (0.027) + amount (0.025) + well lit (0.023) + peanut (0.019) + excellent work (0.012) + nicely lit (0.012) + chair (0.012) + photography (0.011) + menu (0.011) + presentation (0.011) + photoshop (0.010) + wonderful color (0.010) + food photography (0.009) + peanut butter (0.009) + way (0.008)	subject
27	tone (0.181) + balance (0.105) + cute (0.063) + motion (0.058) + white balance (0.052) + cast (0.050) + blur (0.040) + yummy (0.037) + cute idea (0.037) + wasnt (0.031) + onion (0.028) + design (0.021) + color cast (0.020) + wait (0.019) + color tone (0.019) + show (0.018) + competition (0.016) + color balance (0.012) + bug (0.012) + nature (0.011)	color_light
28	sharpness (0.145) + splash (0.111) + arrangement (0.108) + ten (0.043) + tighter (0.041) + sky (0.036) + lighting nice (0.033) + still nice (0.030) + juice (0.028) + maybe bit (0.027) + top ten (0.026) + note (0.025) + stand (0.025) + nice arrangement (0.023) + cause (0.020) + wonder (0.020) + tighter crop (0.019) + dust (0.018) + vivid (0.016) + brilliant (0.015)	composition
29	background (0.450) + cup (0.070) + milk (0.063) + coffee (0.056) + work (0.054) + nice work (0.040) + white background (0.028) + bean (0.026) + well (0.016) + background color (0.015) + color background (0.012) + corn (0.011) + shop (0.009) + plain (0.008) + red background (0.008) + background nice (0.007) + way (0.007) + favourite (0.006) + different background (0.006) + background good (0.006)	composition
30	strawberry (0.151) + monitor (0.103) + scene (0.099) + glare (0.088) + brighter (0.068) + kitchen (0.064) + label (0.055) + range (0.050) + print (0.037) + restaurant (0.024) + tool (0.023) + tablecloth (0.022) + melon (0.019) + dessert (0.019) + money (0.018) + tonal range (0.017) + preference (0.017) + bit brighter (0.015) + composition dof (0.014) + dof shallow (0.014)	color_light
31	comment (0.166) + people (0.129) + thanks (0.113) + good color (0.067) + author (0.057) + lack (0.047) + message (0.036) + thank (0.035) + dpc (0.033) + popcorn (0.023) + vote (0.023) + photographer (0.023) + thanks comment (0.018) + cut (0.018) + dnmc (0.018) + straw (0.017) + addition (0.016) + many people (0.014) + voter (0.013) + get (0.012)	general_impression
32	angle (0.202) + space (0.117) + spoon (0.066) + fork (0.058) + drop (0.051) + curve (0.040) + negative space (0.040) + mood (0.028) + different angle (0.025) + kinda (0.023) + handle (0.023) + coloring (0.022) + potato (0.022) + steam (0.022) + cat (0.020) + poster (0.019) + movie (0.018) + feel (0.017) + interesting idea (0.016) + gold (0.013)	composition
33	sorry (0.148) + foreground (0.100) + youre (0.086) + lemon (0.075) + thing (0.043) + voting (0.040) + berry (0.040) + blurry (0.039) + movement (0.035) + really great (0.034) + position (0.030) + really cool (0.029) + lip (0.027) +	dof_and_focus

	submission (0.021) + lovely color (0.021) + score (0.021) + figure (0.021) + im sorry (0.020) + oh (0.019) + oof (0.018)	
34	image (0.640) + noise (0.030) + eye (0.026) + viewer (0.026) + size (0.025) + item (0.024) + bg (0.024) + good image (0.019) + part (0.016) + image good (0.015) + portion (0.010) + feel (0.009) + way (0.008) + attention (0.006) + grainy (0.006) + sharp image (0.006) + whole (0.006) + line (0.006) + image little (0.006) + neat image (0.006)	dof_and_focus
35	depth (0.166) + hand (0.148) + field (0.132) + depth field (0.115) + anything (0.107) + grape (0.049) + ill (0.036) + background little (0.023) + emotion (0.020) + right hand (0.018) + shallow depth (0.017) + wrinkle (0.016) + suit (0.014) + heck (0.013) + lovely shot (0.011) + white white (0.010) + icecream (0.010) + intention (0.010) + link (0.010) + ew (0.009)	dof_and_focus
36	choice (0.115) + bowl (0.092) + table (0.082) + comp (0.058) + word (0.039) + grain (0.038) + half (0.030) + cloth (0.027) + play (0.026) + good choice (0.021) + thing (0.021) + meal (0.020) + wood (0.019) + definition (0.018) + rim (0.017) + jar (0.017) + change (0.017) + text (0.016) + tho (0.016) + bag (0.015)	composition
37	everything (0.179) + simple (0.133) + skin (0.062) + nice simple (0.045) + hair (0.045) + maybe little (0.040) + nice clean (0.037) + purple (0.037) + clean (0.035) + little less (0.028) + brightness (0.025) + clean shot (0.022) + dot (0.019) + edit (0.019) + color clarity (0.018) + shez (0.018) + great light (0.016) + simple clean (0.016) + cool image (0.016) + joke (0.015)	color_light
38	apple (0.276) + shadow (0.264) + stem (0.049) + way (0.034) + excellent (0.025) + set (0.024) + harsh (0.024) + candle (0.023) + shot challenge (0.022) + breakfast (0.018) + little harsh (0.017) + good detail (0.017) + girl (0.016) + stretch (0.013) + garlic (0.012) + body (0.011) + harsh shadow (0.011) + brain (0.011) + swirl (0.010) + gotta (0.010)	color_light
39	idea (0.559) + great idea (0.091) + good idea (0.084) + nice idea (0.074) + execution (0.019) + idea good (0.019) + cool idea (0.017) + idea nice (0.014) + creative idea (0.014) + idea challenge (0.011) + idea great (0.011) + neat idea (0.011) + idea lighting (0.009) + idea color (0.008) + idea execution (0.007) + original idea (0.007) + mouse (0.005) + funny idea (0.005) + idea photo (0.004) + wife (0.004)	general_impression
40	side (0.224) + nice color (0.130) + blue (0.115) + flame (0.048) + color good (0.047) + cant (0.039) + whats (0.035) + right side (0.034) + tilt (0.032) + left side (0.030) + tea (0.030) + pot (0.025) + meet (0.024) + interesting composition (0.020) + meet challenge (0.013) + theyre (0.013) + square crop (0.013) + hand side (0.012) + dull (0.010) + branch (0.010)	color_light
41	look (0.352) + point (0.143) + score (0.101) + area (0.068) + isnt (0.066) + black white (0.053) + focal point (0.034) + high score (0.022) + look good (0.016) + eye (0.016) + doesnt look (0.016) + contest (0.015) + baby (0.014) + tray (0.011) + degree (0.009) + way (0.009) + master (0.008) + thing (0.007) + nice execution (0.007) + creative (0.006)	general_impression
42	luck (0.263) + good luck (0.236) + theme (0.091) + challenge (0.075) + course (0.048) + shot good (0.042) + luck challenge (0.042) + well good (0.033) + nice macro (0.016) + best luck (0.014) + challenge theme (0.012) + cliche (0.012) + long time (0.012) + jmo (0.010) + luck next (0.009) + nice good (0.009) + good good (0.008) + much glare (0.007) + jmo course (0.007) + course good (0.007)	general_impression
43	thats (0.255) + lol (0.166) + candy (0.085) + hope (0.076) + work (0.067) + good work (0.058) + great concept (0.046) + site (0.026) + store (0.018) + contrasting (0.017) + youll (0.017) + arent (0.016) + group (0.015) + lamp (0.015) + thing (0.014) + way (0.013) + street (0.012) + tad dark (0.011) + background great (0.011) + concept great (0.010)	composition
44	highlight (0.160) + really good (0.093) + end (0.076) + pick (0.070) + slice (0.070) + texture (0.061) + week (0.040) + please (0.039) + top pick (0.037) + rice (0.032) + nice texture (0.030) + th (0.030) + mess (0.029) + cork (0.024) + texture color (0.024) + good nice (0.021) + blown highlight (0.020) + best challenge (0.020) + otherwise (0.019) + good (0.017)	color_light

45	imo (0.219) + clever (0.188) + stuff (0.157) + clever idea (0.087) + ha (0.071) + idea (0.065) + background bit (0.047) + anyone (0.042) + logo (0.028) + gradient (0.026) + well composed (0.019) + color white (0.018) + eye (0.004) + bit busy (0.004) + id (0.003) + way (0.003) + neat (0.002) + display (0.001) + part (0.001) + attention (0.001)	general_impression
46	dof (0.349) + guess (0.070) + shallow dof (0.051) + man (0.049) + love (0.045) + head (0.041) + p (0.041) + distraction (0.035) + hey (0.028) + shame (0.027) + burger (0.022) + pov (0.021) + nice dof (0.020) + portrait (0.020) + bunch (0.017) + mm (0.017) + id (0.015) + ah (0.013) + deep dof (0.011) + duck (0.010)	dof_and_focus
47	fun (0.192) + need (0.114) + day (0.112) + pretty good (0.072) + hue (0.059) + attempt (0.051) + basket (0.034) + effect (0.031) + twist (0.026) + nice effect (0.022) + red apple (0.020) + speck (0.020) + tack (0.019) + fun idea (0.017) + kernel (0.017) + tack sharp (0.017) + also good (0.016) + spirit (0.016) + light bit (0.015) + green apple (0.015)	color_light
48	border (0.150) + wow (0.139) + good shot (0.103) + distracting (0.055) + fan (0.047) + little distracting (0.041) + droplet (0.036) + factor (0.033) + much good (0.031) + green (0.028) + cherry (0.027) + wow factor (0.021) + water (0.021) + exposure (0.020) + symmetry (0.019) + water droplet (0.015) + havent (0.014) + still great (0.014) + shoot (0.014) + creative shot (0.013)	composition
49	great shot (0.157) + capture (0.141) + touch (0.085) + favorite (0.070) + nice touch (0.038) + nice capture (0.037) + hm (0.033) + expression (0.032) + favorite challenge (0.031) + pun (0.031) + book (0.028) + moment (0.028) + recipe (0.024) + great capture (0.024) + bar (0.019) + sun (0.018) + nice crisp (0.016) + square (0.015) + crisp (0.014) + superb (0.012)	general_impression
50	great color (0.110) + pepper (0.107) + vote (0.057) + presentation (0.055) + sugar (0.055) + box (0.050) + color nice (0.040) + nice sharp (0.028) + salt (0.028) + ground (0.027) + pasta (0.027) + bright color (0.023) + ok (0.022) + dog (0.022) + originality (0.019) + morning (0.019) + connection (0.015) + memory (0.014) + oil (0.014) + bright (0.013)	color_light
51	concept (0.359) + ad (0.092) + execution (0.065) + winner (0.049) + good concept (0.043) + nice concept (0.037) + lime (0.036) + beautiful color (0.031) + editing (0.030) + prop (0.025) + salad (0.022) + scheme (0.019) + appearance (0.017) + color scheme (0.016) + concept good (0.015) + halo (0.015) + wonderful (0.014) + humour (0.013) + country (0.013) + magazine (0.012)	composition
52	lighting (0.556) + good lighting (0.053) + nice lighting (0.035) + great lighting (0.026) + lighting good (0.023) + lighting little (0.019) + lighting color (0.017) + lighting composition (0.017) + kid (0.017) + desat (0.016) + little flat (0.016) + finger (0.015) + coke (0.013) + lighting harsh (0.013) + lighting great (0.013) + timing (0.012) + harsh (0.012) + little harsh (0.010) + otherwise nice (0.010) + well great (0.009)	color_light
53	right (0.114) + water (0.105) + frame (0.103) + bottom (0.096) + left (0.096) + corner (0.064) + face (0.052) + pea (0.037) + couple (0.027) + eye (0.019) + thing (0.017) + drop (0.016) + right corner (0.015) + blueberry (0.013) + beverage (0.013) + top right (0.013) + low right (0.012) + smile (0.011) + bottom right (0.011) + upper right (0.011)	composition
54	light (0.380) + ice (0.070) + cream (0.050) + action (0.037) + matter (0.037) + source (0.036) + light source (0.029) + ice cream (0.029) + liquid (0.029) + subject matter (0.023) + bump (0.020) + little light (0.016) + light color (0.016) + good light (0.013) + painting (0.013) + color light (0.012) + ball (0.012) + vibrant color (0.012) + nice light (0.012) + air (0.011)	dof_and_focus
55	contrast (0.383) + chocolate (0.097) + level (0.071) + color contrast (0.038) + speed (0.034) + shutter (0.032) + white (0.031) + shutter speed (0.025) + tip (0.024) + adjustment (0.024) + bite (0.017) + contrast color (0.017) + good contrast (0.016) + great contrast (0.015) + little contrast (0.014) + choice subject (0.013) + boost (0.011) + bumping (0.010) + display (0.010) + brightness (0.010)	color_light
56	top (0.177) + egg (0.153) + orange (0.147) + edge (0.118) + bread (0.062) + someone (0.059) + yellow (0.050) + yolk (0.022) + shell (0.020) + meat (0.017) + term (0.016) + backlighting (0.016) + section (0.015) + peel (0.015)	subject

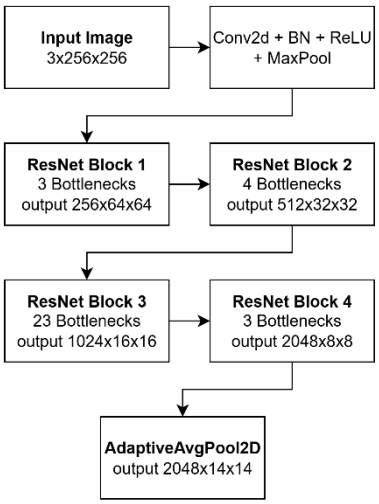
	+ somehow (0.012) + line (0.010) + ham (0.010) + part (0.009) + top glass (0.008) + composition bit (0.008)	
57	bit (0.607) + taste (0.054) + little bit (0.043) + knife (0.030) + bit dark (0.028) + fire (0.019) + bit focus (0.017) + bit soft (0.017) + name (0.017) + bit flat (0.014) + bit sharp (0.013) + neat (0.013) + finish (0.011) + pixel (0.010) + bit bright (0.010) + bit contrast (0.010) + trick (0.009) + compliment (0.007) + image bit (0.007) + perhaps bit (0.007)	color_light
58	picture (0.491) + dont (0.153) + pic (0.112) + nice picture (0.031) + care (0.026) + great picture (0.022) + good picture (0.022) + way (0.019) + pp (0.013) + nice pic (0.012) + rate (0.010) + dont care (0.009) + picture nice (0.009) + step (0.008) + complimentary color (0.007) + picture little (0.006) + justice (0.006) + part (0.006) + good way (0.006) + bottom glass (0.006)	general_impression
59	subject (0.304) + im (0.287) + im sure (0.084) + main subject (0.032) + pink (0.023) + little much (0.020) + aspect (0.018) + trouble (0.014) + sauce (0.013) + way (0.012) + pumpkin (0.012) + sweet (0.011) + warm (0.011) + shot im (0.011) + eye (0.011) + good score (0.010) + boring (0.009) + good subject (0.009) + subject nice (0.009) + crowd (0.008)	subject

Lampiran 3. Arsitektur Hybrid Model CNN-LSTM yang Digunakan

CNN Encoder:

```
EncoderCNN                                [1, 196, 2048]            --
├─Sequential: 1-1                         [1, 2048, 8, 8]           --
│    └─Conv2d: 2-1                        [1, 64, 128, 128]         (9,408)
│    └─BatchNorm2d: 2-2                   [1, 64, 128, 128]         (128)
│    └─ReLU: 2-3                          [1, 64, 128, 128]         --
│    └─MaxPool2d: 2-4                     [1, 64, 64, 64]           --
│    └─Sequential: 2-5                    [1, 256, 64, 64]          --
│    │    └─Bottleneck: 3-1               [1, 256, 64, 64]          (75,008)
│    │    └─Bottleneck: 3-2               [1, 256, 64, 64]          (70,400)
│    │    └─Bottleneck: 3-3               [1, 256, 64, 64]          (70,400)
│    └─Sequential: 2-6                    [1, 512, 32, 32]          --
│    │    └─Bottleneck: 3-4               [1, 512, 32, 32]          379,392
│    │    └─Bottleneck: 3-5               [1, 512, 32, 32]          280,064
│    │    └─Bottleneck: 3-6               [1, 512, 32, 32]          280,064
│    │    └─Bottleneck: 3-7               [1, 512, 32, 32]          280,064
│    └─Sequential: 2-7                    [1, 1024, 16, 16]         --
│    │    └─Bottleneck: 3-8               [1, 1024, 16, 16]         1,512,448
│    │    └─Bottleneck: 3-9               [1, 1024, 16, 16]         1,117,184
│    │    └─Bottleneck: 3-10              [1, 1024, 16, 16]         1,117,184
│    │    └─Bottleneck: 3-11              [1, 1024, 16, 16]         1,117,184
│    │    └─Bottleneck: 3-12              [1, 1024, 16, 16]         1,117,184
│    │    └─Bottleneck: 3-13              [1, 1024, 16, 16]         1,117,184
│    │    └─Bottleneck: 3-14              [1, 1024, 16, 16]         1,117,184
│    │    └─Bottleneck: 3-15              [1, 1024, 16, 16]         1,117,184
│    │    └─Bottleneck: 3-16              [1, 1024, 16, 16]         1,117,184
│    │    └─Bottleneck: 3-17              [1, 1024, 16, 16]         1,117,184
│    │    └─Bottleneck: 3-18              [1, 1024, 16, 16]         1,117,184
│    │    └─Bottleneck: 3-19              [1, 1024, 16, 16]         1,117,184
│    │    └─Bottleneck: 3-20              [1, 1024, 16, 16]         1,117,184
│    │    └─Bottleneck: 3-21              [1, 1024, 16, 16]         1,117,184
│    │    └─Bottleneck: 3-22              [1, 1024, 16, 16]         1,117,184
│    │    └─Bottleneck: 3-23              [1, 1024, 16, 16]         1,117,184
│    │    └─Bottleneck: 3-24              [1, 1024, 16, 16]         1,117,184
│    │    └─Bottleneck: 3-25              [1, 1024, 16, 16]         1,117,184
│    │    └─Bottleneck: 3-26              [1, 1024, 16, 16]         1,117,184
│    │    └─Bottleneck: 3-27              [1, 1024, 16, 16]         1,117,184
│    │    └─Bottleneck: 3-28              [1, 1024, 16, 16]         1,117,184
│    │    └─Bottleneck: 3-29              [1, 1024, 16, 16]         1,117,184
│    │    └─Bottleneck: 3-30              [1, 1024, 16, 16]         1,117,184
│    └─Sequential: 2-8                    [1, 2048, 8, 8]           --
│    │    └─Bottleneck: 3-31              [1, 2048, 8, 8]           6,039,552
│    │    └─Bottleneck: 3-32              [1, 2048, 8, 8]           4,462,592
│    │    └─Bottleneck: 3-33              [1, 2048, 8, 8]           4,462,592
├─AdaptiveAvgPool2d: 1-2                  [1, 2048, 14, 14]         --
==========================================================================================
Total params: 42,500,160
Trainable params: 42,274,816
Non-trainable params: 225,344
Total mult-adds (Units.GIGABYTES): 10.19
==========================================================================================
```

EncoderCNN yang digunakan adalah ResNet-style CNN yang telah dimodifikasi untuk menghasilkan fitur spasial dengan dimensi *output* `[2048, 14, 14]`, cocok untuk *image captioning*. Total parameter mencapai 42.500.160, dengan output yang mempertahankan konteks spasial yang kaya melalui *bottleneck blocks*. Berikut struktur utama arsitektur dalam bentuk blok.



- **ResNet Block:** Tiap blok terdiri dari bottleneck modules yang mengikuti struktur `1x1 → 3x3 → 1x1` convolution.
- **Adapted Output**: Alih-alih di-flatten untuk klasifikasi, output `2048x14x14` disimpan untuk **spatial attention** di decoder.

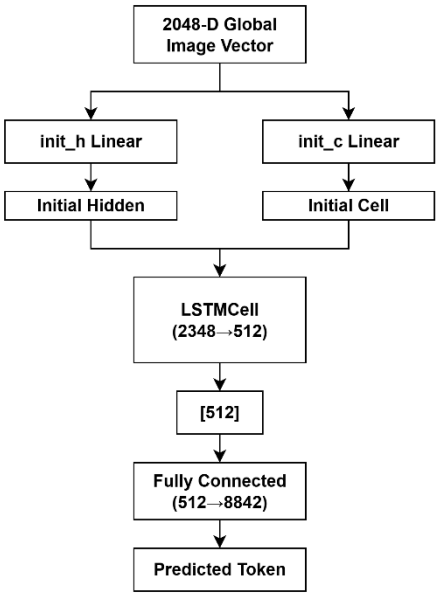
Block	Layer Type	Output Size	Param
Initial	Conv+BN+ReLU+Pool	64x64x64	9,408
Sequential(1)	3x Bottleneck	256x64x64	~225K
Sequential(2)	4x Bottleneck	512x32x32	~1.1M
Sequential(3)	23x Bottleneck	1024x16x16	~25M
Sequential(4)	3x Bottleneck	2048x8x8	~17M
AdaptiveAvgPool2d	Global Pooling	2048x14x14	0

Output `[batch, 2048, 14, 14]` dari CNN ini digunakan sebagai *spatial image features*. Selain itu, *global average pooled vector* `[2048]` digunakan untuk menginisialisasi h0 dan c0 pada LSTM *decoder*.

RNN Decoder:

```
Decoder forward pass successful. Output shape: torch.Size([2, 17, 8842])

DecoderRNN architecture:
DecoderRNN(
  (embedding): Embedding(8842, 300)
  (dropout): Dropout(p=0.5, inplace=False)
  (init_h): Linear(in_features=2048, out_features=512, bias=True)
  (init_c): Linear(in_features=2048, out_features=512, bias=True)
  (lstm): LSTMCell(2348, 512)
  (fc): Linear(in_features=512, out_features=8842, bias=True)
)
```



Layer	Dimensi Masukan	Dimensi Keluaran	Keterangan
Embedding	`[T, 8842]`	`[T, 300]`	*Input: caption tokens*
init_h/c	`[2048]`	`[512]`	Init *hidden* & *cell* from CNN features
LSTMCell	`[2348]`	`[512]`	2048 (*img*) + 300 (*text embedding*)
FC	`[512]`	`[8842]`	Output vocab logits

Decoder yang digunakan dalam penelitian ini merupakan model *Recurrent Neural Network* bertipe LSTM yang dilengkapi dengan *embedding layer*, *transformasi hidden state* awal, serta layer klasifikasi. *Decoder* ini menerima input berupa fitur visual dari *encoder* (berdimensi `[2048, 14, 14]`), dan menghasilkan urutan kata (*caption*) berdimensi `[T, vocab_size]`, dengan T adalah panjang *caption* (maksimum 17 token pada *forward pass* yang ditunjukkan). Secara umum, arsitektur ini mengikuti pendekatan *encoder-decoder sequence generation* yang umum digunakan dalam *image captioning*.

Langkah pertama dalam proses *decoding* adalah transformasi fitur global dari gambar menjadi vektor inisialisasi *hidden state* (h0) dan *cell state* (c0) untuk LSTM. Fitur global diperoleh melalui proses *average pooling* dari fitur spasial (dengan dimensi awal `[2048, 14,`

`14]`), sehingga menghasilkan vektor `[2048]`. Vektor ini kemudian dimasukkan ke dalam dua *layer* linear, yaitu `init_h` dan `init_c`, yang mengubah dimensi dari 2048 ke 512 sebagai ukuran *hidden state* LSTM. Transformasi ini penting agar LSTM dapat mulai dari *state* yang merepresentasikan konteks visual gambar secara global.

Setelah inisialisasi, *decoder* menerima *input* berupa token-token teks dalam bentuk indeks kata, yang kemudian dipetakan menjadi vektor *embedding* berukuran 300 melalui *layer embedding*. Ukuran *embedding* ini sesuai dengan GloVe 300D, meskipun pada implementasi ini model tidak menggunakan *pretrained embedding* (pretrained_embeddings=None) dan membangun *embedding* dari awal (`freeze_embeddings=False`). Vektor *embedding* dari token saat ini kemudian dikonkatenasi dengan vektor fitur visual spasial yang telah diringkas (dalam eksperimen ini digunakan fitur global `[2048]`), sehingga membentuk vektor input LSTM dengan ukuran 2348 (hasil dari 2048 + 300). LSTM kemudian memproses vektor ini dengan ukuran *hidden state* 512 pada setiap *time step*.

Arsitektur LSTM digunakan dalam bentuk LSTMCell, bukan LSTM standar, yang memungkinkan kontrol manual pada proses per-token dan integrasi teknik seperti *attention* pada versi lanjutannya. Output dari setiap langkah LSTM kemudian diberikan ke *layer* fc (*fully connected*) berukuran 512 → 8842, untuk menghasilkan distribusi probabilitas atas seluruh kosakata (*vocab size* 8842). Output inilah yang digunakan untuk memilih kata berikutnya dalam proses *decoding caption*, baik melalui *argmax*, *sampling*, atau *beam search*.

(Halaman ini sengaja dikosongkan)

BIODATA PENULIS

Penulis dilahirkan di Kediri, 20 Juni 2003, merupakan anak pertama dari 3 bersaudara. Penulis telah menempuh pendidikan formal yaitu di TK Stella Maris BSD, SDK Stella Maris BSD dan SDK Petra Kota Kediri, SMPN 8 Kota Kediri dan SMKN 1 Kota Kediri. Setelah lulus dari SMKN pada tahun 2021, Penulis mengikuti SNMPTN dan diterima di Departemen Teknik Informatika FTEIC - ITS pada tahun 2021 dan terdaftar dengan NRP 5025211048.

Di Departemen Teknik Informatika Penulis sempat aktif di beberapa kegiatan Seminar yang diselenggarakan oleh Departemen, dan aktif sebagai Asisten Dosen maupun Praktikum.