Jurnal Riset Multidisiplin Edukasi
https://journal.hasbaedukasi.co.id/index.php/jurmie
Halaman : 389 - 410

IMPLEMENTASI ALGORITMA CONVOLUTIONAL NEURAL NETWORK (CNN) PADA EVALUASI STRATEGI PROMPT ENGINEERING TERHADAP AKURASI VISUAL QUESTION ANSWERING (VQA)

Muhammad Riyan[1], Khairuddin Nasution[2], Darjat Saripurna[3]
Universitas Islam Sumatera Utara, Medan, Indonesia [1,2,3]
Email: muhammadriyan031004@gmail.com[1], khairuddin_nst@uisu.ac.id[2], darjatsaripurna@gmail.com[3]

Informasi

Volume : 3
Nomor : 11
Bulan : November
Tahun : 2026
E-ISSN : 3062-9624

Abstract

Visual Question Answering (VQA) is the task of answering natural language questions about the content of an image. This study aims to identify the prompt engineering strategy with the highest VQA accuracy and to test whether the accuracy differs significantly among strategies. The model is a CNN–Transformer architecture with ResNet-50 as the visual encoder, BERT as the text encoder, and a cross-attention fusion layer. Five strategies were evaluated: zero-shot, few-shot, chain-of-thought, instruction-based, and role-prompting. The data consist of 8,000 questions from the VQA v2.0 validation set selected by stratified random sampling. Each strategy was trained with five random seeds and evaluated using VQA Accuracy and One-Way ANOVA. The results show that the instruction-based strategy achieved the highest mean accuracy of 37.54%, while the other strategies ranged from 36.63% to 37.17%. However, the differences were not statistically significant (p = 0.2897). The instruction-based strategy was then implemented in a web application prototype.

Keyword: *Visual Question Answering, Convolutional Neural Network, Transformer, Prompt Engineering, VQA Accuracy.*

Abstrak

Visual Question Answering (VQA) adalah tugas menjawab pertanyaan berbahasa alami tentang isi suatu citra. Penelitian ini bertujuan mengetahui strategi prompt engineering dengan akurasi VQA tertinggi serta menguji signifikansi perbedaan akurasi antarstrategi. Model yang digunakan adalah arsitektur CNN–Transformer dengan ResNet-50 sebagai visual encoder, BERT sebagai text encoder, dan lapisan fusi cross-attention. Lima strategi yang dievaluasi adalah zero-shot, few-shot, chain-of-thought, instruction-based, dan role-prompting. Data berupa 8.000 pertanyaan dari VQA v2.0 validation yang diambil secara stratified random sampling. Setiap strategi dilatih dengan lima random seed, lalu dievaluasi menggunakan VQA Accuracy dan One-Way ANOVA. Hasil menunjukkan strategi instruction-based memperoleh akurasi rata-rata tertinggi sebesar 37,54%, sedangkan strategi lainnya berkisar antara 36,63% dan 37,17%. Namun, perbedaan tersebut tidak signifikan secara statistik (p = 0,2897). Strategi instruction-based kemudian diterapkan pada purwarupa aplikasi web.

Kata Kunci : *Visual Question Answering, Convolutional Neural Network, Transformer, Prompt Engineering, VQA Accuracy.*

A. PENDAHULUAN

Visual Question Answering (VQA) merupakan salah satu tugas kecerdasan buatan yang menggabungkan computer vision dan natural language processing (NLP) untuk menghasilkan jawaban atas pertanyaan berbahasa alami mengenai suatu citra. Sistem VQA tidak hanya dituntut memahami informasi visual, tetapi juga menafsirkan pertanyaan dan menghubungkan keduanya untuk menghasilkan jawaban yang tepat. Penerapan VQA telah berkembang pada berbagai bidang, seperti pendidikan, sistem bantu bagi penyandang tunanetra, dan pencarian visual.

Relevansi VQA dalam konteks Indonesia telah ditunjukkan melalui pengembangan sistem VQA berbahasa Indonesia untuk mendukung pembelajaran visual anak taman kanak-kanak menggunakan model Bootstrapping Language-Image Pre-training (BLIP) yang dipadukan dengan model penerjemah No Language Left Behind (NLLB). Hasil penelitian menunjukkan bahwa tingkat ketepatan jawaban berbeda berdasarkan kategori pertanyaan, dengan akurasi mencapai 100% pada beberapa kategori, tetapi menurun hingga 87,5% pada kata keterangan dan 62,5% pada kata sifat. Hasil tersebut menunjukkan bahwa performa VQA dapat dipengaruhi oleh karakteristik pertanyaan, sehingga evaluasi akurasi perlu mempertimbangkan variasi kondisi yang digunakan (Hanifah et al., 2024).

Dalam pengembangan arsitektur VQA, kombinasi Convolutional Neural Network (CNN) dan Transformer menjadi salah satu pendekatan yang relevan untuk menggabungkan informasi visual dan tekstual. CNN mampu mengekstraksi fitur spasial dari citra, sedangkan Transformer dapat menangkap hubungan kontekstual melalui mekanisme attention. Penggunaan CNN sebagai encoder dan Transformer sebagai decoder pada tugas image captioning berbahasa Indonesia mampu menghasilkan keluaran teks yang relevan, dengan ResNet sebagai salah satu backbone yang memberikan hasil ekstraksi fitur yang baik (Mulyawan et al., 2023). Perbandingan EfficientNet-B0 dan ResNet-50 pada klasifikasi citra daun jagung juga menunjukkan akurasi masing-masing sebesar 94% dan 93%, sehingga keduanya dapat dipertimbangkan sebagai backbone CNN dalam pengolahan fitur visual (Pamungkas et al., 2023). Selain itu, perbandingan CNN konvensional, ResNet-50, dan Vision Transformer pada klasifikasi citra kanker payudara menunjukkan bahwa arsitektur berbasis residual dan mekanisme attention dapat memberikan performa yang lebih baik dibandingkan CNN konvensional (Grace & Gunawan, 2025).

Selain arsitektur, strategi penyusunan masukan atau prompt engineering juga dapat memengaruhi keluaran model. Prompt engineering merupakan proses merancang masukan

agar model menghasilkan keluaran yang lebih relevan dan sesuai konteks tanpa harus mengubah bobot model (Rizky, 2025).

Pengaruh strategi prompt terhadap performa model telah ditunjukkan pada beberapa penelitian. Pada tugas ekstraksi entitas, penggunaan one-shot meningkatkan F1-Score dari 0,5722 pada zero-shot menjadi 0,7368, sedangkan skenario 20-shot mencapai 0,8023 (Shafelbilyunazra & Prasetya, 2025). Pada sistem tanya jawab mengenai informasi obat dan vitamin, pendekatan few-shot menghasilkan nilai BERTScore dan ROUGE yang lebih tinggi dibandingkan zero-shot (Haromain et al., 2025). Sementara itu, strategi chain-of-thought (CoT) dilaporkan meningkatkan validitas jawaban dan skor BERTScore dibandingkan zero-shot pada sistem retrieval-augmented generation (RAG) berbasis dokumen PDF (Yahya et al., 2026). Namun, penelitian lain menunjukkan bahwa few-shot menghasilkan akurasi 0,945 dan mengungguli chain-of-thought pada tugas klasifikasi sentimen berbahasa Indonesia (Rahmawati et al., 2026).

Hasil penelitian tersebut menunjukkan bahwa pengaruh strategi prompt engineering terhadap performa model tidak selalu memiliki pola yang sama. Perbedaan hasil dapat dipengaruhi oleh tugas, karakteristik data, dan jenis model yang digunakan. Sebagian besar penelitian tersebut juga berfokus pada model dengan modalitas teks, seperti klasifikasi sentimen, ekstraksi entitas, dan tanya jawab berbasis dokumen. Dengan demikian, masih diperlukan penelitian yang menguji strategi prompt pada tugas yang melibatkan kombinasi informasi visual dan tekstual.

Pada bidang VQA, penelitian sebelumnya telah menunjukkan penerapan VQA berbahasa Indonesia, tetapi menggunakan BLIP dan NLLB serta belum menjadikan strategi prompt engineering sebagai variabel yang diuji secara terkontrol (Hanifah et al., 2024). Di sisi lain, penggunaan CNN dan Transformer telah diterapkan pada tugas image captioning, tetapi belum membahas evaluasi strategi prompt pada tugas VQA (Mulyawan et al., 2023). Kondisi tersebut menunjukkan adanya peluang penelitian untuk menguji hubungan antara strategi prompt engineering dan akurasi pada arsitektur VQA berbasis CNN-Transformer.

Berdasarkan uraian latar belakang yang telah dijelaskan, rumusan masalah dalam penelitian ini berfokus pada strategi prompt engineering manakah di antara lima strategi yang diuji yang mampu menghasilkan akurasi Visual Question Answering (VQA) tertinggi, baik secara keseluruhan maupun berdasarkan masing-masing kategori pertanyaan, yaitu yes/no, number, dan other. Selain itu, penelitian ini juga bertujuan untuk mengetahui apakah terdapat

perbedaan yang signifikan secara statistik terhadap tingkat akurasi VQA yang dihasilkan oleh kelima strategi prompt engineering tersebut.

Agar penelitian ini lebih terarah dan dapat diselesaikan sesuai dengan waktu serta sumber daya yang tersedia, penelitian ini memiliki beberapa batasan, yaitu CNN visual encoder berupa ResNet-50 atau EfficientNet-B0 dan text encoder yang digunakan berada dalam kondisi pretrained dan dibekukan (frozen), sehingga tidak dilakukan pelatihan ulang secara penuh (full fine-tuning), melainkan hanya lapisan fusi Transformer dan Answer Head yang menjalani proses pelatihan. Data penelitian dibatasi pada subset VQA v2.0 validation set sebanyak kurang lebih 5.000 hingga 10.000 pasangan citra dan pertanyaan, bukan keseluruhan dataset VQA v2.0. Pendekatan penjawaban yang digunakan juga dibatasi pada metode closed-vocabulary classification, yaitu model memilih jawaban dari sekumpulan kelas jawaban yang telah ditentukan dan bukan menghasilkan jawaban secara generatif atau bebas. Selain itu, penelitian ini hanya berfokus pada perbandingan pengaruh lima strategi prompt engineering terhadap akurasi VQA, bukan pada eksplorasi maupun optimasi arsitektur model baru. Pertanyaan dan jawaban yang digunakan dalam penelitian ini menggunakan bahasa Inggris sesuai dengan bahasa asli dataset VQA v2.0, sehingga hasil penelitian tidak secara langsung menggambarkan performa model terhadap teks berbahasa Indonesia. Selanjutnya, seluruh proses pelatihan dan pengujian model dilaksanakan pada lingkungan komputasi awan, yaitu Google Colaboratory dan/atau Kaggle Notebook, dengan mempertimbangkan keterbatasan alokasi GPU berdasarkan kuota gratis yang tersedia pada masing-masing platform.

Berdasarkan rumusan masalah yang telah diuraikan, penelitian ini bertujuan untuk mengidentifikasi strategi prompt engineering di antara lima strategi yang diuji yang mampu menghasilkan akurasi Visual Question Answering (VQA) tertinggi, baik secara keseluruhan maupun pada masing-masing kategori pertanyaan, yaitu yes/no, number, dan other. Selain itu, penelitian ini juga bertujuan untuk menganalisis signifikansi statistik dari perbedaan tingkat akurasi VQA yang dihasilkan oleh kelima strategi prompt engineering tersebut.

B. METODE PENELITIAN

Penelitian ini menggunakan metode eksperimental untuk menganalisis pengaruh lima strategi prompt engineering, yaitu zero-shot, few-shot, chain-of-thought, instruction-based, dan role-prompting terhadap akurasi model Visual Question Answering (VQA) berbasis arsitektur CNN-Transformer. Tahap penelitian diawali dengan analisis permasalahan, kebutuhan sistem, karakteristik data, serta perangkat keras dan perangkat lunak yang

digunakan. Data penelitian bersumber dari VQA v2.0 Validation Dataset yang memiliki 214.354 pertanyaan dari 40.504 citra Microsoft COCO. Sampel penelitian diambil sebanyak 5.000–10.000 pasangan citra dan pertanyaan menggunakan teknik stratified random sampling berdasarkan tiga kategori jawaban, yaitu yes/no, number, dan other. Data tersebut kemudian dibagi menjadi data latih, validasi, dan uji dengan rasio 70:15:15. Perangkat yang digunakan meliputi Python, PyTorch, Hugging Face Transformers, Google Colaboratory, serta berbagai pustaka pendukung untuk pengolahan data, pelatihan, dan evaluasi model.

Tahapan penyelesaian penelitian meliputi pengumpulan dan praproses data, penyusunan lima templat prompt engineering, perancangan arsitektur CNN-Transformer, pelatihan model, pengujian akurasi, serta analisis statistik. Arsitektur model terdiri atas CNN sebagai visual encoder, text encoder untuk memproses pertanyaan, lapisan fusi cross-attention berbasis Transformer, dan answer head untuk menghasilkan prediksi jawaban menggunakan pendekatan closed-vocabulary classification. Pengujian dilakukan menggunakan data uji yang sama pada setiap strategi dengan pengulangan minimal tiga kali menggunakan random seed berbeda. Hasil akurasi dianalisis menggunakan uji normalitas Shapiro-Wilk, dilanjutkan One-Way ANOVA atau Kruskal-Wallis sesuai distribusi data, serta pengujian post-hoc apabila diperlukan. Strategi dengan akurasi terbaik selanjutnya diimplementasikan dalam purwarupa aplikasi berbasis web menggunakan FastAPI, HTML, CSS, JavaScript, dan SQLite. Pengujian aplikasi dilakukan melalui metode black-box testing untuk memastikan seluruh fungsi berjalan sesuai rancangan.

C. HASIL DAN PEMBAHASAN

Implementasi

Tabel 1 Lingkungan implementasi

Komponen	Google Colaboratory	Laptop
Prosesor / GPU	GPU NVIDIA Tesla T4	AMD Ryzen 7 7735HS 3,20 GHz, AMD Radeon Graphics (tanpa CUDA)
Memori	sesuai alokasi Colab	RAM 16 GB
Sistem operasi	Linux (Colab)	Windows 11 Home 64-bit
Penyimpanan	Google Drive (data, hasil, checkpoint)	SSD 477 GB
Peran	praproses, ekstraksi fitur, 25 pelatihan, evaluasi, uji statistik, menjalankan backend web	mengakses notebook, Drive, dan antarmuka web melalui peramban

Perangkat lunak yang digunakan adalah Python dengan pustaka PyTorch dan torchvision untuk CNN, Hugging Face Transformers untuk BERT, pandas, scikit-learn, SciPy, dan statsmodels untuk pengolahan data dan statistik, serta FastAPI untuk backend aplikasi web. Seluruh tahapan dijalankan melalui satu notebook Colab sehingga urutan eksperimen dapat diulang dengan konfigurasi dan random seed yang sama.

Hasil Praproses dan Pembagian Data

Tabel 2 Distribusi kategori jawaban pada populasi, sampel, dan setiap split

Himpunan data	Jumlah pertanyaan	Jumlah citra	Proporsi	yes/no (%)	number (%)	other (%)
Populasi bersih	214.354	40.504	–	37,57	13,13	49,30
Sampel	8.000	7.021	100%	37,58	13,12	49,30
Train	5.583	4.901	69,79%	37,15	13,18	49,67
Validation	1.183	1.049	14,79%	37,28	14,54	48,18
Test	1.234	1.071	15,42%	39,79	11,51	48,70

Tabel 2 menunjukkan tiga hal. Pertama, stratified sampling berhasil menjaga proporsi kategori sampel sama dengan populasi hingga dua angka desimal. Kedua, karena pembagian dilakukan per citra, rasio split bergeser tipis dari 70/15/15 menjadi 69,79/14,79/15,42 dan proporsi kategori per split sedikit berbeda dari populasi; test set memuat yes/no sedikit lebih banyak (39,79%) dan number sedikit lebih sedikit (11,51%). Ketiga, tidak ada satu pun citra yang muncul di dua split berbeda (irisan citra train–validation, train–test, dan validation–test seluruhnya 0), sehingga model tidak pernah melihat citra uji selama pelatihan.

Kosakata jawaban dibentuk dari 1.000 jawaban ternormalisasi yang paling sering muncul pada split train saja. Kosakata ini mencakup jawaban yang benar untuk 97,9% pertanyaan train dan 92,2% pertanyaan validation. Jika model selalu memilih jawaban terbaik yang ada di kosakata, VQA Accuracy tertinggi yang mungkin dicapai adalah 92,99% pada train, 85,00% pada validation, dan 83,78% pada test (yes/no 100%, number 89,08%, other 69,27%). Angka pada test set ini dihitung setelah seluruh eksperimen selesai dan hanya digunakan sebagai batas atas pembanding, tidak untuk memilih konfigurasi.

Tiga contoh few-shot dipilih secara acak dari split train, satu untuk setiap kategori, dan digunakan secara tetap pada seluruh pertanyaan dan seluruh seed: (1) *Does this sandwich look healthy?* → yes (yes/no). (2) *What is the catcher's number?* → 26 (number). (3) *What is painted on the plane?* → kangaroo (other)

Implementasi Arsitektur CNN–Transformer

Tabel 3 Konfigurasi model dan pelatihan

Komponen	Konfigurasi
Visual encoder	ResNet-50 pretrained ImageNet (IMAGENET1K_V2), dibekukan; keluaran 49 region × 2048 dimensi
Text encoder	bert-base-uncased, dibekukan; panjang token maksimum 128
Lapisan fusi	2 lapisan; d_model 512; 8 attention head; FFN 2048; dropout 0,1; Query = teks, Key/Value = citra
Pooling	token [CLS] setelah fusi
Answer Head	LayerNorm → Linear 512→512 → GELU → Dropout → Linear 512→1.000 → softmax
Fungsi loss	soft-target cross-entropy (target dari 10 jawaban anotator)
Optimizer	AdamW, learning rate 1×10^{-4}, weight decay 0,01, warmup 1 epoch lalu cosine decay
Batch size / epoch maksimum	64 / 30
Early stopping	patience 5 epoch, dipantau dari VQA Accuracy validation
Random seed pelatihan	42, 123, 2024, 7, 99 (seed data tetap 42)

Tabel 4 Jumlah parameter model

Komponen	Jumlah parameter	Status
ResNet-50 (visual encoder)	23.508.032	dibekukan
BERT-base-uncased (text encoder)	108.891.648	dibekukan
Lapisan fusi cross-attention	9.879.040	dilatih
Answer Head	776.680	dilatih
Total	**143.055.400**	10.655.720 dilatih (7,45%)

Pembekuan encoder diverifikasi dengan menghitung sidik jari (hash SHA-256) seluruh bobot ResNet-50 dan BERT sebelum dan sesudah setiap pelatihan. Pada ke-25 pelatihan, sidik jari tersebut tidak berubah, sehingga dapat dipastikan bahwa perbedaan hasil antarstrategi hanya berasal dari lapisan fusi dan Answer Head yang dilatih, bukan dari perubahan encoder.

Implementasi Lima Strategi Prompt Engineering

Tabel 5 Bentuk teks masukan setiap strategi prompt

Strategi	Teks awalan sebelum pertanyaan
Zero-shot	(tanpa awalan)
Few-shot	Here are examples of questions about images and their short answers. Question: Does this sandwich look healthy? Answer: yes. Question: What is the catcher's number? Answer: 26. Question: What is painted on the plane? Answer: kangaroo.
Chain-of-Thought	Let's think step by step. First, find the objects in the image that the question is about. Second, look at their color, number, position, or action. Third, use these observations to give the final short answer.

Strategi	Teks awalan sebelum pertanyaan
Instruction-based	Answer the following question about the image. Reply with a single word or a short phrase. Use yes or no for yes/no questions and digits for counting questions.
Role-prompting	You are a helpful visual assistant. You look carefully at the image and answer questions using only the information that is visible in the image.

Tabel 6 Panjang teks masukan setelah tokenisasi

Strategi	Rata-rata token	Minimum	Maksimum	Terpotong (> 128 token)
Zero-shot	9,43	5	24	0
Role-prompting	37,43	33	52	0
Instruction-based	43,43	39	58	0
Chain-of-Thought	58,43	54	73	0
Few-shot	61,43	57	76	0

Batas 128 token yang sama untuk kelima strategi cukup untuk seluruh masukan: prompt terpanjang (few-shot) hanya mencapai 76 token, sehingga tidak ada teks yang terpotong pada train, validation, maupun test. Dengan demikian, perbedaan hasil antarstrategi tidak disebabkan oleh hilangnya bagian pertanyaan.

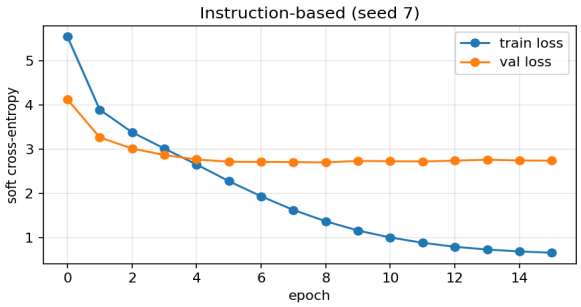
Karena model memakai text encoder BERT dan Answer Head klasifikasi, strategi prompt tidak membuat model menghasilkan teks atau langkah penalaran. Chain-of-Thought dalam penelitian ini dioperasionalkan sebagai templat instruksi penalaran bertahap yang ikut dienkode BERT, sehingga memengaruhi representasi pertanyaan melalui self-attention. Demikian pula, contoh few-shot hanya berupa teks pasangan pertanyaan–jawaban tanpa citra contoh.

Hasil Pelatihan Model

Tabel 7 Ringkasan pelatihan per strategi (rata-rata 5 seed)

Strategi	Waktu pelatihan (detik)	Epoch dijalankan	Epoch terbaik	Suhu kalibrasi T
Zero-shot	142,4	13,8	7,8	1,06
Few-shot	188,2	15,0	9,0	1,09
Chain-of-Thought	189,3	15,4	9,4	1,10
Instruction-based	199,8	17,2	11,2	1,16
Role-prompting	180,3	15,6	9,6	1,08

Early stopping menghentikan pelatihan setelah 11 hingga 24 epoch, jauh sebelum batas 30 epoch. Zero-shot paling cepat karena masukannya paling pendek (rata-rata 9 token), sedangkan strategi dengan awalan lebih panjang memerlukan waktu per epoch yang lebih besar.



Gambar 1 Kurva loss pelatihan dan validasi strategi Instruction-based (seed 7)

Gambar 1 memperlihatkan pola yang sama pada seluruh run. Training loss terus turun (pada contoh ini dari 5,54 menjadi 0,66), sedangkan validation loss berhenti turun di sekitar epoch ke-5 sampai ke-8 pada nilai sekitar 2,7. Rata-rata seluruh run, validation loss minimum adalah 2,67 pada epoch ke-7,8, sementara training loss akhir rata-rata 0,73. Selisih yang lebar ini menunjukkan model mulai menghafal data latih (overfitting) setelah beberapa epoch, yang wajar mengingat hanya 5.583 pertanyaan latih dipakai untuk 10,7 juta parameter yang dilatih. Early stopping dengan memantau VQA Accuracy validation memastikan checkpoint yang disimpan berasal dari epoch sebelum penurunan kemampuan generalisasi yang berarti.

Setelah pelatihan, skor softmax dikalibrasi dengan temperature scaling pada data validation. Nilai T yang diperoleh berkisar 0,91–1,29 (rata-rata 1,10), artinya keyakinan model sebelum kalibrasi sudah cukup dekat dengan akurasinya. Expected Calibration Error (ECE) pada validation turun dari rata-rata 0,085 menjadi 0,074 setelah kalibrasi, dan ECE pada test rata-rata 0,070. Skor terkalibrasi ini cukup informatif: dari 30.850 prediksi test seluruh run, prediksi dengan skor ≥ 50% memiliki VQA Accuracy rata-rata 61,6%, sedangkan prediksi dengan skor < 50% hanya 17,2%. Hasil ini menjadi dasar penggunaan skor keyakinan terkalibrasi pada aplikasi web.

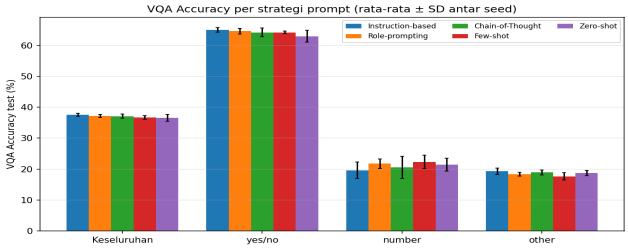
Hasil VQA Accuracy Lima Strategi Prompt

Tabel 8 VQA Accuracy test (%) setiap strategi, rata-rata ± simpangan baku dari 5 seed

Strategi	Keseluruhan	yes/no	number	other
Instruction-based	**37,54 ± 0,50**	**65,07 ± 0,75**	19,63 ± 2,67	**19,29 ± 1,07**
Role-prompting	37,17 ± 0,47	64,64 ± 0,90	21,76 ± 1,56	18,37 ± 0,62
Chain-of-Thought	37,13 ± 0,67	64,22 ± 1,43	20,52 ± 3,58	18,92 ± 0,79
Few-shot	36,73 ± 0,64	64,23 ± 0,42	**22,32 ± 2,16**	17,66 ± 1,15

Strategi	Keseluruhan	yes/no	number	other
Zero-shot	36,63 ± 1,11	62,99 ± 1,88	21,45 ± 2,08	18,68 ± 0,85

Keterangan: nilai tercetak tebal adalah rata-rata tertinggi pada kolom tersebut. Jumlah pertanyaan test: yes/no 491, number 142, other 601.



Gambar 2 VQA Accuracy per strategi prompt dan per kategori (rata-rata ± SD dari 5 seed)

Berdasarkan Tabel 8 dan Gambar 4.2, jawaban atas rumusan masalah pertama adalah sebagai berikut.

1. **Keseluruhan:** Instruction-based tertinggi (37,54%), diikuti Role-prompting (37,17%), Chain-of-Thought (37,13%), Few-shot (36,73%), dan Zero-shot (36,63%). Keempat strategi yang menambahkan awalan semuanya berada di atas zero-shot, tetapi rentangnya kurang dari 1 poin persen.
2. **yes/no:** Instruction-based tertinggi (65,07%) dan Zero-shot terendah (62,99%), dengan selisih 2,08 poin persen.
3. **number:** Few-shot tertinggi (22,32%), sedangkan Instruction-based justru terendah (19,63%). Kategori ini memiliki simpangan baku antar-seed terbesar (1,56–3,58), karena hanya terdiri atas 142 pertanyaan.
4. **other:** Instruction-based tertinggi (19,29%) dan Few-shot terendah (17,66%).

Perbedaan antarkategori jauh lebih besar daripada perbedaan antarstrategi. Semua strategi mencapai sekitar 63–65% pada yes/no, tetapi hanya sekitar 20% pada number dan 18–19% pada other. Sebagai pembanding, menjawab "yes" untuk semua pertanyaan yes/no sudah menghasilkan 63,91%, dan menjawab "no" menghasilkan 63,46%. Model tidak sekadar menebak satu jawaban (48,3% prediksi yes/no berupa "yes" dan 51,5% berupa "no"), tetapi akurasinya pada yes/no hampir sama dengan tebakan tetap tersebut: empat strategi hanya 0,3–1,2 poin persen di atasnya, dan Zero-shot (62,99%) bahkan sedikit di bawahnya. Pada kategori number, prediksi didominasi jawaban "2" (35,3%) dan "1" (29,0%), yang menunjukkan model banyak mengandalkan pola jawaban yang sering muncul. Seluruh hasil ini masih jauh di bawah batas atas kosakata pada test set (83,78%).

Meskipun rata-rata akurasinya hampir sama, kelima strategi tidak selalu memberikan

jawaban yang sama. Pada setiap seed, kelima strategi menghasilkan jawaban identik hanya pada rata-rata 32,5% pertanyaan test. Artinya, strategi prompt mengubah banyak prediksi individual, tetapi jumlah prediksi yang menjadi benar dan yang menjadi salah hampir saling meniadakan sehingga akurasi rata-rata tidak banyak berubah.

Hasil Analisis Statistik

Uji Normalitas Shapiro-Wilk

Tabel 9 Hasil uji Shapiro-Wilk (nilai p) per strategi dan per metrik

Strategi	Keseluruhan	yes/no	number	other
Zero-shot	0,2356	0,1478	0,2152	0,5327
Few-shot	0,8238	0,7678	0,3647	0,2646
Chain-of-Thought	0,4595	0,9227	0,6864	0,4917
Instruction-based	0,8392	0,9179	0,8540	0,9318
Role-prompting	0,9164	0,1892	0,5759	0,9309
Residual gabungan	0,4588	0,3666	0,8090	0,3057

Seluruh nilai p lebih besar dari 0,05, sehingga data dianggap berdistribusi normal dan pengujian dilanjutkan dengan One-Way ANOVA dan uji lanjut Tukey HSD sesuai rancangan pada Bab III. Uji Levene juga tidak menunjukkan perbedaan varians antarstrategi (p = 0,1265–0,7570). Perlu dicatat bahwa dengan hanya 5 pengamatan per strategi, daya uji Shapiro-Wilk rendah, sehingga hasil normalitas ini dibaca dengan hati-hati.

Uji One-Way ANOVA

Tabel 10 Hasil One-Way ANOVA

Metrik	F	p	η^2	Levene p	Keputusan
Keseluruhan	1,340	0,2897	0,211	0,5068	tidak signifikan
yes/no	2,100	0,1187	0,296	0,1265	tidak signifikan
number	0,904	0,4801	0,153	0,6141	tidak signifikan
other	2,252	0,0996	0,311	0,7570	tidak signifikan

Seluruh hipotesis nol, yaitu bahwa rata-rata VQA Accuracy kelima strategi sama, tidak dapat ditolak. Uji Tukey HSD juga tidak menemukan satu pun dari 10 pasangan strategi yang berbeda signifikan pada keempat metrik. Pasangan dengan selisih terbesar pada akurasi keseluruhan adalah Instruction-based dibanding Zero-shot: selisih 0,92 poin persen, interval kepercayaan 95% [−0,43; 2,27], p adj = 0,2883. Pada yes/no, pasangan yang sama memiliki selisih 2,07 poin persen dengan p adj = 0,0821, mendekati tetapi tidak mencapai batas signifikansi.

Nilai η^2 antara 0,15 dan 0,31 menunjukkan bahwa secara deskriptif strategi prompt

menjelaskan sebagian variasi akurasi antar-run. Namun, dengan hanya 5 run per strategi, perbedaan sebesar ini belum cukup kuat untuk dibedakan dari variasi akibat random seed.

Analisis Tambahan

ANOVA satu arah mengasumsikan kelompok saling independen, padahal kelima strategi diuji dengan test set dan seed yang sama. Oleh karena itu, dilakukan analisis tambahan yang memperhitungkan sifat berpasangan tersebut. Analisis ini melengkapi, bukan menggantikan, analisis utama.

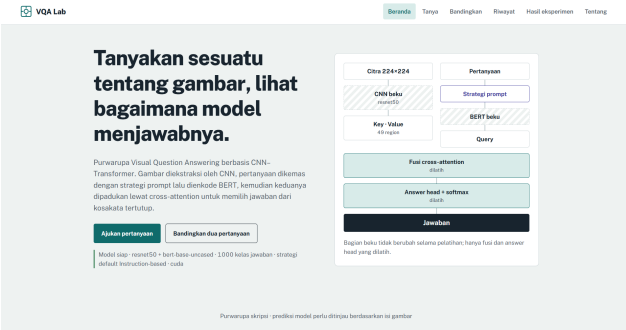
1. **Seed sebagai blok:** uji Friedman (p = 0,2752–0,4510) dan ANOVA pengukuran berulang (p = 0,0993–0,3992) pada keempat metrik sama-sama tidak signifikan.
2. **Tingkat pertanyaan:** skor VQA setiap pertanyaan test dirata-ratakan antar seed, lalu diuji dengan Friedman dan Wilcoxon signed-rank berpasangan dengan koreksi Bonferroni. Uji Friedman tidak signifikan pada keempat metrik (p = 0,2533–0,3820). Dari 40 perbandingan berpasangan, hanya satu yang signifikan setelah koreksi, yaitu Instruction-based lebih tinggi 1,63 poin persen daripada Few-shot pada kategori other (Wilcoxon p = 0,0043, p Bonferroni = 0,0434, IK 95% bootstrap [0,47; 2,78]). Karena uji omnibusnya tidak signifikan, temuan tunggal ini diperlakukan sebagai indikasi, bukan bukti perbedaan.

Berdasarkan analisis utama dan analisis tambahan, jawaban atas rumusan masalah kedua adalah bahwa perbedaan VQA Accuracy antara zero-shot, few-shot, chain-of-thought, instruction-based, dan role-prompting tidak signifikan secara statistik pada α = 0,05, baik untuk akurasi keseluruhan maupun untuk kategori yes/no, number, dan other.

Implementasi dan Pengujian Aplikasi Web

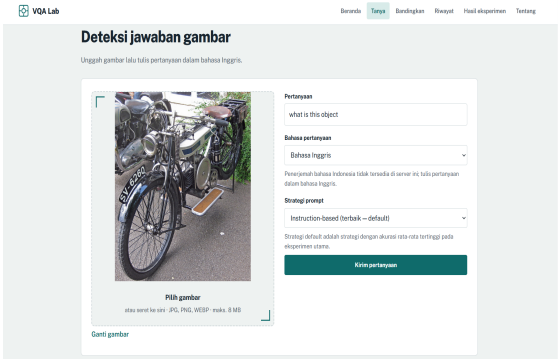
Aplikasi web purwarupa berhasil dijalankan dengan model hasil eksperimen. Backend FastAPI memuat checkpoint kelima strategi satu kali saat dijalankan; setiap strategi diwakili oleh run dengan VQA Accuracy validation tertinggi di antara kelima seed-nya, dan strategi default adalah Instruction-based (seed 99, akurasi test 38,31%) karena memiliki rata-rata akurasi keseluruhan tertinggi. Pada pengujian, backend dijalankan di Google Colab dengan GPU (perangkat cuda), dan antarmuka dibuka melalui peramban.

Tampilan Aplikasi

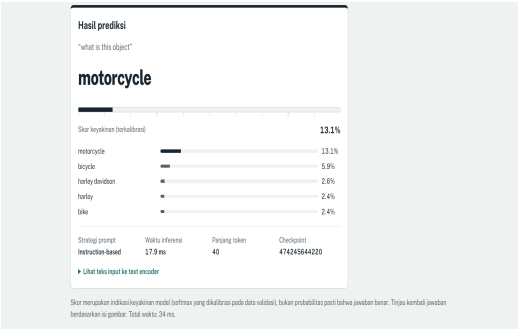


Gambar 3 Halaman Beranda

Halaman Beranda (Gambar 3) menampilkan alur model dan status model yang dimuat: ResNet-50 dan BERT-base-uncased, 1.000 kelas jawaban, serta strategi default Instruction-based.

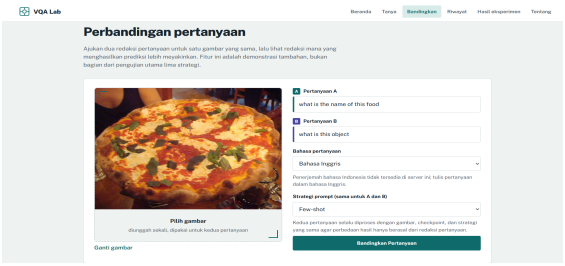


Gambar 4 Halaman Tanya

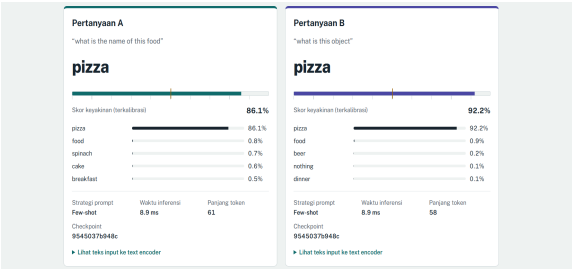


Gambar 5 Hasil prediksi pertanyaan tunggal

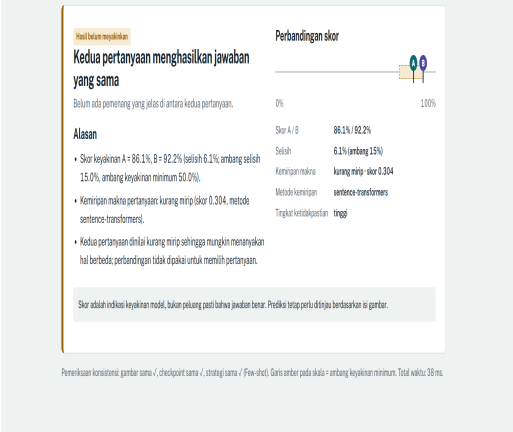
Gambar 5 menunjukkan hasil untuk pertanyaan *what is this object* pada foto sepeda motor. Model menjawab "motorcycle" dengan skor keyakinan terkalibrasi 13,1%, diikuti kandidat "bicycle" (5,9%), "harley davidson" (2,6%), "harley" (2,4%), dan "bike" (2,4%), dengan waktu inferensi model 17,9 ms. Jawabannya tepat, tetapi skornya rendah karena keyakinan model tersebar pada beberapa jawaban yang berkaitan.



Gambar 6 Halaman Perbandingan Pertanyaan

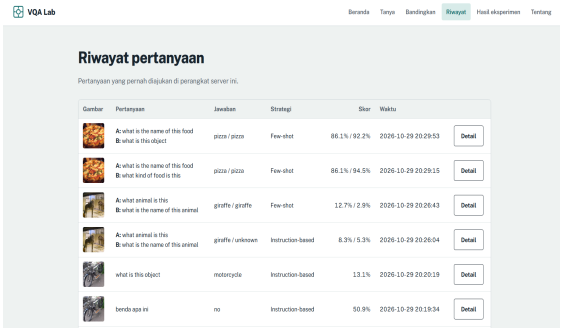


Gambar 7 Kartu hasil Pertanyaan A dan B

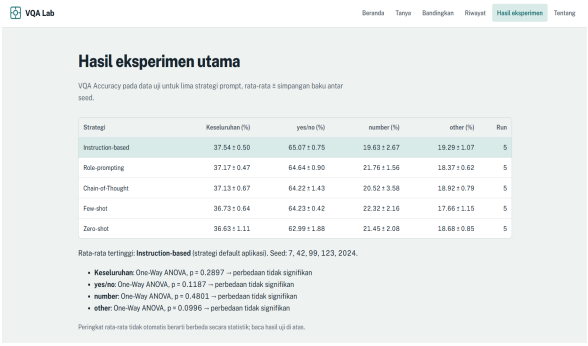


Gambar 8 Ringkasan perbandingan dua pertanyaan

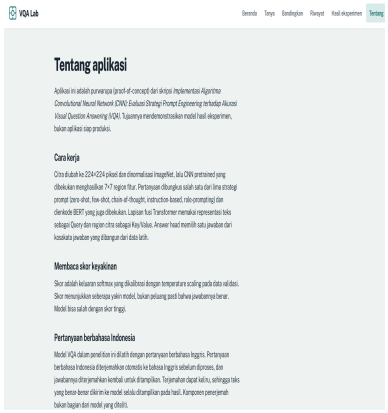
Gambar 6 sampai 8 menunjukkan fitur tambahan Perbandingan Pertanyaan pada foto pizza dengan strategi Few-shot. Pertanyaan A (*what is the name of this food*) dan B (*what is this object*) sama-sama dijawab "pizza", dengan skor 86,1% dan 92,2%. Pemeriksa kemiripan makna menilai kedua pertanyaan kurang mirip (skor 0,304), sehingga sistem tidak memilih pertanyaan pemenang dan menampilkan status "Hasil belum meyakinkan" dengan tingkat ketidakpastian tinggi. Pemeriksaan konsistensi menunjukkan kedua pertanyaan diproses dengan gambar, checkpoint, dan strategi yang sama. Fitur ini adalah demonstrasi tambahan dan tidak digunakan untuk menjawab rumusan masalah.



Gambar 9 Halaman Riwayat



Gambar 10 Halaman Hasil Eksperimen



Gambar 11 Halaman Tentang

Halaman Riwayat (Gambar 9) menyimpan pertanyaan beserta gambar mini, jawaban, strategi, skor, dan waktu. Halaman Hasil Eksperimen (Gambar 10) menampilkan angka yang sama dengan Tabel 8 dan ringkasan uji ANOVA pada Tabel 10. Halaman Tentang (Gambar 11) menjelaskan cara kerja model, arti skor keyakinan, dan batasan fitur.

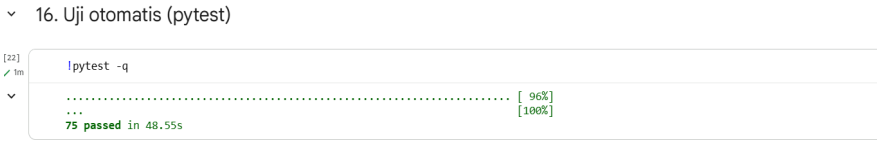
Pengujian Black-Box

Tabel 11 Hasil pengujian black-box aplikasi web

No	Skenario	Langkah pengujian	Hasil yang diharapkan	Hasil aktual	Status
1	Membuka aplikasi	membuka alamat aplikasi	Beranda tampil dengan status model siap	Beranda tampil dengan status "Model siap" dan strategi default Instruction-based (Gambar 4.3)	Berhasil
2	Mengunggah gambar	memilih berkas JPG pada halaman Tanya	pratinjau gambar tampil	pratinjau foto sepeda motor tampil (Gambar 4.4)	Berhasil
3	Mengirim pertanyaan bahasa Inggris	mengisi *what is this object* lalu menekan Kirim pertanyaan	jawaban, skor, kandidat teratas, dan strategi tampil	jawaban "motorcycle" 13,1%, lima kandidat, strategi Instruction-based, 17,9 ms (Gambar 4.5)	Berhasil
4	Memilih strategi lain	memilih Few-shot lalu mengirim	strategi pada hasil sesuai pilihan	kartu hasil mencantumkan strategi Few-shot (Gambar 4.7)	Berhasil
5	Membandingkan dua pertanyaan	mengunggah satu gambar, mengisi Pertanyaan A dan B, lalu menekan Bandingkan Pertanyaan	dua kartu hasil, ringkasan, dan pemeriksaan konsistensi tampil	dua jawaban "pizza" (86,1% dan 92,2%), status "Hasil belum meyakinkan", konsistensi gambar/checkpoint /strategi terpenuhi (Gambar 4.7–4.8)	Berhasil
6	Menolak rekomendasi bila makna berbeda	membandingkan dua pertanyaan dengan kemiripan rendah	tidak ada pertanyaan yang direkomendasikan	kemiripan 0,304 (kurang mirip), tidak ada pemenang (Gambar 4.8)	Berhasil
7	Melihat riwayat	membuka halaman Riwayat	daftar pertanyaan sebelumnya tampil	daftar dengan gambar mini, jawaban, strategi, skor, dan waktu tampil (Gambar 4.9)	Berhasil

No	Skenario	Langkah pengujian	Hasil yang diharapkan	Hasil aktual	Status
8	Melihat hasil eksperimen	membuka halaman Hasil eksperimen	tabel lima strategi sesuai hasil eksperimen	tabel sama dengan Tabel 4.8 (Gambar 4.10)	Berhasil

Pengujian Otomatis



Gambar 12 Hasil pengujian otomatis pytest di Google Colab

Selain pengujian black-box, kode diuji dengan 75 pengujian otomatis (pytest) yang memeriksa validitas dataset, bentuk tensor, pembekuan encoder, satu langkah pelatihan, penyimpanan dan pemuatan checkpoint, perhitungan VQA Accuracy dengan kasus yang diketahui hasilnya, logika rekomendasi perbandingan pertanyaan, serta respons REST API terhadap masukan valid dan tidak valid. Seluruh 75 pengujian lulus dalam 48,55 detik (Gambar 12).

Pembahasan

Pengaruh Strategi Prompt

Hasil ini berbeda dengan penelitian terdahulu yang menggunakan large language model (LLM). Shafelbilyunazra dan Prasetya (2025) melaporkan kenaikan F1-Score dari 0,5722 pada zero-shot menjadi 0,8023 pada 20-shot, Haromain et al. (2025) memperoleh skor BERTScore dan ROUGE yang lebih tinggi dengan few-shot, dan Yahya et al. (2026) melaporkan peningkatan validitas jawaban sebesar 37,14% dengan chain-of-thought. Pada LLM, prompt mengubah proses pembangkitan teks: model dapat meniru pola contoh dan menuliskan langkah penalaran sebelum menjawab.

Pada penelitian ini, kondisinya berbeda dalam tiga hal. Pertama, BERT hanya mengenkode teks dan tidak menghasilkan jawaban maupun langkah penalaran, sehingga chain-of-thought hanya berfungsi sebagai konteks tambahan, bukan penalaran yang benar-benar dijalankan. Kedua, awalan setiap strategi bersifat tetap dan sama untuk seluruh pertanyaan, sehingga tidak membawa informasi baru tentang gambar maupun pertanyaan tertentu. Lapisan fusi yang dilatih terpisah untuk setiap strategi dapat menyesuaikan diri terhadap awalan yang selalu sama tersebut. Ketiga, contoh few-shot hanya berupa teks tanpa

citra, sehingga tidak memberi contoh hubungan antara isi gambar dan jawaban. Ketiga hal ini menjelaskan mengapa kelima strategi menghasilkan akurasi rata-rata yang hampir sama, walaupun pada tingkat pertanyaan kelima strategi hanya memberikan jawaban identik pada 32,5% pertanyaan test.

Temuan ini sejalan dengan sintesis penelitian terdahulu pada Bab II bahwa pengaruh strategi prompt tidak konsisten antartugas dan antarmodel. Rahmawati et al. (2026) menemukan bahwa chain-of-thought tidak selalu unggul pada model berukuran kecil; pada penelitian ini pun chain-of-thought tidak lebih baik daripada instruksi biasa. Secara deskriptif, Instruction-based unggul pada akurasi keseluruhan, yes/no, dan other, yang mungkin terkait dengan isi instruksinya yang menyebut format jawaban singkat serta jawaban yes/no. Namun, Instruction-based justru terendah pada kategori number walaupun instruksinya menyebut penggunaan angka, dan seluruh keunggulannya tidak signifikan. Oleh karena itu, urutan peringkat strategi tidak dapat dijadikan dasar untuk menyatakan bahwa satu strategi lebih baik.

Perbedaan Antarkategori Pertanyaan

Kategori pertanyaan jauh lebih menentukan akurasi daripada strategi prompt, sejalan dengan anjuran Hanifah et al. (2024) untuk mengevaluasi VQA per kategori. Akurasi yes/no sekitar 63–65% setara dengan tebakan tetap "yes" (63,91%), sedangkan number sekitar 20% dan other sekitar 18–19%. Pada number, model cenderung memilih jawaban yang paling sering muncul ("2" dan "1"). Pola ini menunjukkan bahwa model lebih banyak mengandalkan pola bahasa dan frekuensi jawaban dibandingkan pemahaman isi gambar.

Ada tiga penyebab yang paling mungkin. Pertama, data latih hanya 5.583 pertanyaan untuk 1.000 kelas jawaban, sehingga sebagian besar kelas hanya memiliki sedikit contoh. Kedua, kurva loss (Gambar 4.1) menunjukkan overfitting sejak sekitar epoch ke-5 sampai ke-8, yaitu training loss terus turun sementara validation loss mendatar di sekitar 2,7. Ketiga, ResNet-50 dan BERT dibekukan sesuai batasan penelitian, sehingga fitur visual berasal dari pelatihan klasifikasi ImageNet dan tidak disesuaikan untuk menghitung objek atau mengenali atribut seperti warna dan jenis benda. Hanifah et al. (2024) mencapai akurasi tinggi dengan BLIP, yaitu model yang telah dilatih pada pasangan citra–teks berskala besar, sehingga perbedaan tingkat akurasi dengan penelitian ini terutama mencerminkan perbedaan skala data dan pra-pelatihan, bukan strategi prompt.

Variasi Antar-Seed dan Kekuatan Uji

Variasi akibat random seed sebanding dengan perbedaan antarstrategi. Misalnya, run

Zero-shot dengan seed 2024 mencapai 38,10%, lebih tinggi daripada rata-rata Instruction-based (37,54%), padahal rata-rata Zero-shot adalah yang terendah. Jika perbandingan hanya dilakukan dengan satu run per strategi, kesimpulan tentang strategi terbaik dapat berubah hanya karena pemilihan seed. Hal ini menegaskan pentingnya pengulangan dengan beberapa seed seperti yang dirancang pada Bab III.

Nilai η^2 sebesar 0,15–0,31 menunjukkan bahwa perbedaan antarstrategi secara deskriptif tidak nol, tetapi 5 pengamatan per strategi belum cukup untuk membedakannya dari variasi seed. Analisis tambahan yang memperhitungkan sifat berpasangan data (seed sebagai blok dan perbandingan per pertanyaan) memberikan kesimpulan yang sama. Dengan demikian, hasil tidak signifikan ini lebih tepat dibaca sebagai "tidak ditemukan perbedaan yang berarti" daripada "terbukti sama persis".

D. KESIMPULAN

Berdasarkan hasil penelitian dan pembahasan dapat disimpulkan bahwa arsitektur CNN–Transformer berhasil diimplementasikan untuk tugas Visual Question Answering (VQA) dengan menggunakan ResNet-50 beku sebagai visual encoder, bert-base-uncased beku sebagai text encoder, dua lapisan fusi cross-attention Transformer, serta answer head dengan klasifikasi 1.000 jawaban. Model memiliki 143.055.400 parameter, dengan 10.655.720 parameter (7,45%) yang dilatih melalui 25 kali eksperimen menggunakan lima strategi prompt engineering dan lima variasi random seed pada GPU NVIDIA Tesla T4 dengan total waktu pelatihan 75 menit. Hasil pengujian menunjukkan bahwa strategi instruction-based memperoleh rata-rata VQA Accuracy keseluruhan tertinggi sebesar 37,54% ± 0,50, diikuti role-prompting sebesar 37,17% ± 0,47, chain-of-thought sebesar 37,13% ± 0,67, few-shot sebesar 36,73% ± 0,64, dan zero-shot sebesar 36,63% ± 1,11. Strategi instruction-based juga menghasilkan akurasi tertinggi pada kategori yes/no sebesar 65,07% dan other sebesar 19,29%, sedangkan kategori number memperoleh akurasi tertinggi melalui strategi few-shot sebesar 22,32%. Meskipun demikian, hasil uji One-Way ANOVA menunjukkan bahwa perbedaan akurasi antarstrategi tidak signifikan secara statistik, dengan nilai p masing-masing sebesar 0,2897 untuk keseluruhan, 0,1187 untuk yes/no, 0,4801 untuk number, dan 0,0996 untuk other, yang seluruhnya lebih besar dari taraf signifikansi 0,05. Dengan demikian, H0 diterima dan uji lanjut Tukey HSD juga tidak menemukan perbedaan signifikan antar pasangan strategi, sehingga dapat disimpulkan bahwa instruction-based memiliki performa rata-rata

terbaik, tetapi keunggulannya belum terbukti signifikan secara statistik dibandingkan strategi lainnya.

E. DAFTAR PUSTAKA

Amelia, W. (2023). Implementasi Jaringan Syaraf Tiruan Sebagai Sistem Diagnosis Stunting Balita Menggunakan Metode Backpropagation Di Puskesmas Brebes. Jurnal Minfo Polgan, 12(2), 1874–1883. https://doi.org/10.33395/jmp.v12i2.13044

Arwansyah, A., Sy, H., & Susanto, C. (2026). PROSIDING SEMINAR ILMIAH SISTEM INFORMASI DAN TEKNOLOGI INFORMASI Pusat Penelitian dan Pengabdian pada Masyarakat (P3M) Universitas Dipa Makassar Implementasi Mixture of Experts (MoE) Berbasis CNN dan Vision Transformer untuk Klasifikasi Citra. 15(2), 95–104. https://doi.org/10.36774/sisiti.v15i2.2524

Cibro, E., & Yahfizham. (2024). Mengidentifikasi Dasar Algoritma Pemrograman. Jurnal Ilmiah Sistem Informasi Dan Ilmu Komputer, 4(1), 194–206. https://doi.org/10.55606/juisik.v4i1.763

Goyal, Y., Khot, T., Summers-Stay, D., Batra, D., & Parikh, D. (2017). Making the V in VQA Matter: Elevating the Role of Image Understanding in Visual Question Answering. (CVPR), 6904–6913. http://arxiv.org/abs/1612.00837

Grace, S. J., & Gunawan, D. (2025). PERBANDINGAN CNN, RESNET50, DAN VISION TRANSFORMER UNTUK KLASIFIKASI KANKER PAYUDARA BERBASIS WEB. Rabit : Jurnal Teknologi Dan Sistem Informasi Univrab, 10(2), 945–956. https://doi.org/10.36341/rabit.v10i2.6420

Hanifah, A., Wakhidatus Sholikah, R., & Hari Ginardi, R. (2024). Visual Question Answering Bahasa Indonesia Berbasis Deep Learning untuk Pembelajaran Visual Anak TK Indonesian Visual Question Answering Based on Deep Learning for Kindergarten's Visual Learning. In Februari (Vol. 23, Number 1).

Haromain, I., Munir, S., & Rahmah, A. (2025). Analisa Prompt Engineering pada Large Language Model dengan Retrieval-Augmented Generation untuk Informasi Obat dan Vitamin. Indonesian Journal Computer Science, 4(2), 144–153. https://jurnal.bsi.ac.id/index.php/ijcs/article/view/10005

itsmariodias. (2023). VQA Validation Dataset [Data set]. Kaggle. https://www.kaggle.com/datasets/itsmariodias/vqa-validation-dataset

Kurniawati, L., Priyanto, D., Ningsih, N. S., Syahrir, M., & Rismayati, R. (2025). Perbandingan

Metode Berbasis Decision Tree dalam Deteksi Penyakit Paru-Paru. Jurnal Bumigora Information Technology (BITe), 7(1), 51–62. https://doi.org/10.30812/bite.v7i1.4909

Mulyawan, R., Sunyoto, A., & Muhammad, A. H. (2023). INTERNATIONAL JOURNAL ON INFORMATICS VISUALIZATION journal homepage : www.joiv.org/index.php/joiv INTERNATIONAL JOURNAL ON INFORMATICS VISUALIZATION Pre-Trained CNN Architecture Analysis for Transformer-Based Indonesian Image Caption Generation Model. www.joiv.org/index.php/joiv

Nalendra Zaradiva, M., & Dewi, I. A. (2026). Perbandingan Kinerja Model ResNet-50 dan MobileNetV2 untuk Klasifikasi Citra Menggunakan Metode CNN. DNS-DIGITAL NEXUS SYSTEMATIC JOURNAL, 02(2), 13–22. https://www.kaggle.com/datasets/aryashah2k/breast-

Pamungkas, W. G., Wardhana, M. I. P., Sari, Z., & Azhar, Y. (2023). Leaf Image Identification: CNN with EfficientNet-B0 and ResNet-50 Used to Classified Corn Disease. Jurnal RESTI, 7(2), 326–333. https://doi.org/10.29207/resti.v7i2.4736

Rahmawati, R., Teknik, E., & Salsabila, A. S. (2026). Evaluasi Teknik Prompt Engineering pada LLM Berukuran Kecil untuk Klasifikasi Sentimen Bahasa Indonesia. DNS-DIGITAL NEXUS SYSTEMATIC JOURNAL, 02(2), 1–7.

Rizky, M. A. (2025). Analisis Efektivitas Dua Jenis Gaya Prompt dalam Model LLM Berbasis RAG. Jurnal Komtika (Komputasi Dan Informatika), 9(1), 76–86. https://doi.org/10.31603/komtika.v9i1.13488

Setyadi, R. A., Rahman, S., Manurung, D., Hasanah, M., & Indrawati, A. (2024). IMPLEMENTASI TRANSFER LEARNING UNTUK KLASIFIKASI PENYAKIT PADA DAUN CABAI MENGGUNAKAN CNN. Jurnal Teknologi Informasi, 5(2), 304–315. https://doi.org/10.46576/djtechno

Shafelbilyunazra, A., & Prasetya, D. D. (2025). Pengaruh Few-shot Learning pada Kinerja LLM untuk Ekstraksi Entitas Iklan Lowongan Kerja. Jurnal Pustaka AI (Pusat Akses Kajian Teknologi Artificial Intelligence), 5(2), 418–427. https://doi.org/10.55382/jurnalpustakaai.v5i2.1069

Wijaya, I. N. S. W., Seputra, K. A., & Dewi, N. P. N. P. (2025). FINE TUNNING MODEL INDOBERT UNTUK ANALISIS SENTIMEN BERITA PARIWISATA INDONESIA. Jurnal Pendidikan Teknologi Dan Kejuruan, 22(2), 195–204. https://www.detik.com/search/searchall?query=wisata&siteid=3&sortby=time&fromdatex=01/01/2022&

Yahya, K., Moeis, D., Thamrin, M., Yunarti, S., Felicia Watratan, A., & Mallu, S. (2026). Analisis Perbandingan Efektivitas Zero-Shot vs Chain-of-Thought Prompting dalam Meningkatkan Presisi Informasi pada LLM Berbasis Dokumen PDF. RIGGS: Journal of Artificial Intelligence and Digital Business, 5(1), 2394–2402. https://doi.org/10.31004/riggs.v5i1.6524