Studi Ablasi *Hyperparameter* pada Model *Image Captioning* dengan Integrasi Fitur Deteksi Objek

Moh. Novil Ma'arij[1], Yuni Yamasari[2]

[1,2] Program Studi S1 Teknik Informatika, Universitas Negeri Surabaya
[1]mohnovil.22134@mhs.unesa.ac.id
[2]yuniyamasari@unesa.ac.id

Abstrak*—*Image captioning* berbasis arsitektur *encoder-decoder* dengan mekanisme Bahdanau *Attention* umumnya hanya mengandalkan fitur global CNN, yang belum sepenuhnya mampu merepresentasikan detail objek, atribut, lokasi, maupun hubungan antar objek dalam citra. Berbagai penelitian mengatasi keterbatasan ini dengan menambahkan fitur objek dari model *object detector,* namun bukti empiris mengenai efektivitas perbandingan langsung antar jenis *object detector,* khususnya YOLOv8, pada tugas ini masih terbatas. Penelitian ini membandingkan kinerja tiga skenario integrasi fitur, yaitu *Baseline* (fitur global Xception), *One-Stage* (Xception + YOLOv8), dan *Two-Stage* (Xception + Faster R-CNN), menggunakan *dual projection feature fusion* dan *decoder* GRU dengan Bahdanau *Attention*. Eksperimen dilakukan pada dataset Flickr8k dengan 18 skenario hasil kombinasi tiga skenario model, tiga konfigurasi *hyperparameter*, dan dua ukuran *batch size*, dievaluasi menggunakan metrik BLEU, METEOR, ROUGE-L, CIDEr, dan SPICE. Hasil ANOVA dua arah menunjukkan bahwa konfigurasi *hyperparameter* merupakan faktor paling dominan terhadap variasi kinerja model (46,0% variansi CIDEr; 80,3% variansi SPICE), diikuti ukuran *batch* (24,3%; 6,4%), sedangkan pemilihan arsitektur *object detector* tidak berpengaruh signifikan secara statistik (2,3%; 4,1%; p>0,05). Skenario *Two-Stage* (Faster R-CNN) memperoleh rata-rata kinerja tertinggi, namun selisihnya relatif kecil terhadap skenario lain. Konfigurasi terbaik diperoleh pada kombinasi *batch size* 16 dan *label smoothing*, dengan CIDEr tertinggi 0,4285. Temuan ini mengindikasikan bahwa optimasi strategi pelatihan memberikan dampak lebih besar terhadap kualitas *caption* dibandingkan pemilihan jenis *object detector.

***Kata Kunci*— *image captioning*, Bahdanau *attention*, YOLOv8, Faster R-CNN, optimasi *hyperparameter*.**

I. PENDAHULUAN

Penggunaan media sosial dan berbagai *platform* digital menghasilkan volume gambar yang sangat besar, dengan estimasi sekitar 3000 TB gambar baru diunggah ke Facebook setiap hari [1]. Namun, banyak aset digital masih memiliki metadata gambar yang kosong, tidak lengkap, atau tidak akurat sehingga menghambat proses pengelolaan, pengindeksan, dan pencarian gambar [2], [3]. *Image captioning* menjadi salah satu solusi untuk menghasilkan deskripsi gambar secara otomatis yang dapat dimanfaatkan sebagai metadata pada proses *indexing* dan pencarian citra [4]. Pendekatan ini juga mampu mengurangi waktu dan biaya anotasi manual, terutama pada dataset berskala besar dan layanan digital[5], [6].

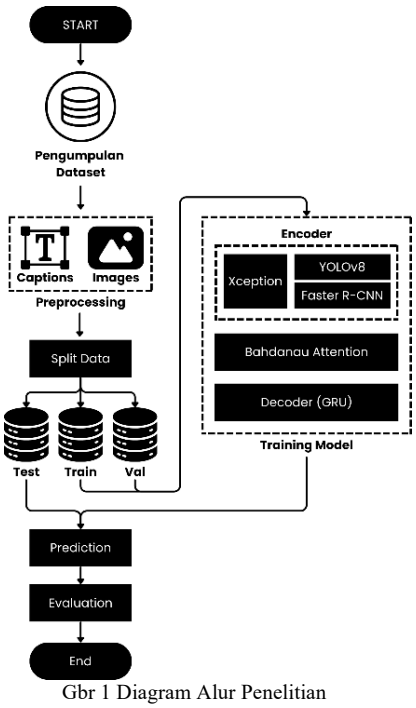
Arsitektur *encoder-decoder* merupakan pendekatan yang paling banyak digunakan dalam penelitian *image captioning*, di mana *encoder* mengekstraksi representasi visual dari gambar dan *decoder* menghasilkan *caption* secara bertahap, umumnya satu kata pada setiap *time step* [7]. Untuk meningkatkan kualitas *caption*, berbagai penelitian menambahkan fitur objek dari model *object detector* karena fitur global CNN belum sepenuhnya mampu merepresentasikan detail objek, atribut, lokasi, maupun hubungan antar objek. Meskipun demikian, fitur objek juga memiliki keterbatasan, seperti kurangnya konteks global dan ketergantungan pada proses pelatihan *object detector.* Oleh karena itu, penelitian terbaru cenderung menggabungkan fitur global dan fitur objek agar diperoleh representasi visual yang lebih komprehensif [8].

Penggabungan fitur *object detection* dengan fitur CNN telah menunjukkan potensi dalam meningkatkan kinerja *image captioning*. Al-Malla dkk. [9] mengintegrasikan fitur objek dari YOLOv4 dengan fitur global Xception dan melaporkan peningkatan skor CIDEr sebesar 15,04% dibandingkan penggunaan fitur global saja. Namun, bukti empiris mengenai efektivitas penggunaan *object detector* pada *image captioning* masih terbatas. Sebagian besar penelitian hanya mengevaluasi satu jenis *object detector* sehingga belum diketahui model yang paling efektif untuk menghasilkan *caption.* Selain itu, meskipun YOLOv8 telah menunjukkan peningkatan performa pada berbagai tugas *object detection*, pemanfaatannya sebagai ekstraktor fitur dalam *image captioning* masih relatif sedikit dieksplorasi [10], [11].

Berdasarkan kondisi tersebut, penelitian ini membandingkan secara langsung YOLOv8 dan Faster R-CNN sebagai ekstraktor fitur objek yang dikombinasikan dengan fitur global dari Xception pada model *image captioning* berbasis arsitektur *encoder-decoder* dengan mekanisme Bahdanau *Attention.* Penelitian ini tidak mengusulkan arsitektur baru, melainkan menganalisis pengaruh kedua *object detector* terhadap kualitas semantik *caption* yang dihasilkan serta mengevaluasi strategi optimasi pelatihan untuk meningkatkan kinerja model.

II. METODOLOGI PENELITIAN

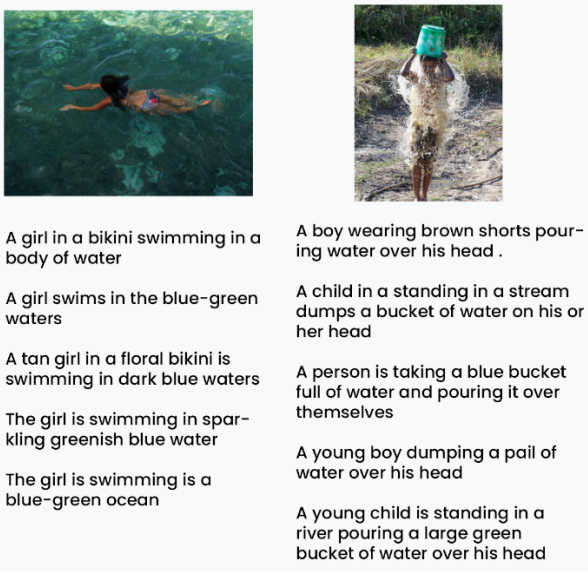
Tahapan penelitian yang dilakukan pada penelitian ini ditunjukkan pada Gbr 1.



Gbr 1 Diagram Alur Penelitian

A. *Pengumpulan Dataset*

Dataset yang digunakan pada penelitian ini adalah Flickr8k yang diperoleh dari repositori terbuka Kaggle. Dataset ini terdiri atas dua jenis data, yaitu *file* gambar berformat .jpg dan *file* teks berformat .txt. Secara keseluruhan, dataset ini memuat 8.091 gambar dan 40.455 *caption*, di mana setiap gambar memiliki lima *caption* yang mendeskripsikan isi gambar dari sudut pandang yang berbeda. Keberagaman susunan kalimat pada *caption* untuk gambar yang sama memberikan variasi deskripsi dengan makna yang tetap serupa, sehingga membantu model mempelajari hubungan antara informasi visual dan representasi bahasa secara lebih baik. Contoh gambar beserta lima *caption* yang menyertainya ditunjukkan pada Gbr 2.



Gbr 2 Sampel Gambar dan Caption Dataset

B. *Pra-Pemrosesan Data*

Tahap pra-pemrosesan data bertujuan untuk menyiapkan data agar sesuai dengan kebutuhan model *image captioning*. Proses ini terdiri atas dua jalur utama, yaitu pra-pemrosesan data citra dan pra-pemrosesan data teks.

1) Pra-pemrosesan Data Citra

Pada tahap ini, seluruh citra diubah ukurannya *(resizing)* menjadi 299 × 299 piksel agar sesuai dengan ukuran masukan model CNN Xception. Selanjutnya, nilai piksel dinormalisasi ke rentang -1 hingga 1 menggunakan fungsi *preprocessing* bawaan Xception untuk menjaga stabilitas proses pelatihan. Setelah itu, setiap citra diekstraksi menggunakan dua jenis model. Model Xception digunakan untuk menghasilkan fitur spasial *(spatial features),* sedangkan YOLOv8 atau Faster R-CNN digunakan untuk menghasilkan fitur objek yang meliputi informasi lokasi *bounding box,* skor kepercayaan *(confidence score)*, dan kelas objek yang terdeteksi. Hasil ekstraksi fitur dari masing-masing model kemudian disimpan dalam format NumPy (.npy) sehingga dapat digunakan secara langsung pada proses pelatihan tanpa perlu melakukan ekstraksi ulang.

2) Pra-pemrosesan Data Teks

Pada tahap awal, *file caption* dikonversi ke format JSON karena pustaka evaluasi pycocoevalcap dan pycocotools menggunakan format tersebut sebagai masukan dalam proses evaluasi. Selanjutnya, *caption* dibersihkan melalui proses *text cleaning* dengan menghapus tanda baca, angka, serta karakter yang tidak diperlukan. Setelah proses pembersihan selesai, setiap *caption* ditambahkan token khusus *<start>* pada awal kalimat dan *<end>* pada akhir kalimat untuk menandai batas awal dan akhir urutan kata.

Tahap berikutnya adalah membagi dataset menjadi data pelatihan, validasi, dan pengujian sesuai pembagian yang telah ditentukan. Setelah proses pembagian selesai, dibentuk kosakata *(vocabulary)* berdasarkan 3.000 kata yang paling sering muncul pada data pelatihan. Kata-kata yang tidak termasuk dalam kosakata tersebut dipetakan ke token *<unk> (unknown)*.

Sebelum proses *encoding*, seluruh *caption* diseragamkan panjangnya *(padding)* berdasarkan panjang *caption* terpanjang pada dataset. *Caption* yang memiliki jumlah kata lebih sedikit akan ditambahkan *token <pad>* hingga mencapai panjang yang sama. Terakhir, setiap kata pada *caption* dikonversi menjadi representasi numerik *(integer encoding)* berdasarkan indeks pada *vocabulary*, sehingga dapat digunakan sebagai masukan bagi model selama proses pelatihan.

C. *Splitting Dataset*

Pembagian dataset dilakukan berdasarkan gambar *(image-level splitting)* sehingga seluruh *caption* yang terkait dengan satu gambar selalu berada pada subset yang sama. Pendekatan ini bertujuan untuk mencegah terjadinya *data leakage*, yaitu

kondisi ketika gambar yang sama muncul pada data pelatihan dan data pengujian atau validasi melalui *caption* yang berbeda. Pada penelitian ini, pembagian dataset menggunakan metode Karpathy split, yang terdiri atas 6.091 gambar untuk data pelatihan *(training),* 1.000 gambar untuk data validasi *(validation),* dan 1.000 gambar untuk data pengujian *(testing).*

D. *Arsitektur Model*

Pendekatan yang digunakan pada penelitian ini adalah arsitektur *encoder-decoder* dengan mekanisme Bahdanau *Attention*. Arsitektur ini terdiri atas dua komponen utama, yaitu *encoder* yang bertugas mengekstraksi representasi visual dari citra dan *decoder* yang menghasilkan *caption* secara bertahap.

1) Encoder

Pada penelitian ini, fitur global citra diekstraksi menggunakan model Xception, sedangkan fitur objek diperoleh menggunakan YOLOv8 atau Faster R-CNN sesuai dengan skenario eksperimen yang digunakan. Xception menghasilkan fitur spasial berukuran 10 × 10 × 2048, yang kemudian diubah *(reshape)* menjadi representasi 100 × 2048. Sementara itu, YOLOv8 atau Faster R-CNN mendeteksi objek pada citra dan memilih maksimal 20 objek berdasarkan nilai *confidence score* tertinggi. Setiap objek direpresentasikan dalam bentuk [x, y, w, h, *confidence, class_id*]. Apabila jumlah objek yang terdeteksi kurang dari 20, maka dilakukan *zero padding* untuk menjaga konsistensi ukuran masukan model.

Selanjutnya, fitur dari Xception dan fitur objek diproyeksikan secara terpisah menggunakan *dual projection feature fusion.* Proses ini mengubah dimensi fitur Xception dari 100 × 2048 menjadi 100 × 256, sedangkan fitur objek diproyeksikan menjadi 20 × 256. Untuk mencegah slot hasil *zero padding* memperoleh perhatian selama proses *attention,* diterapkan *attention mask* yang memberikan nilai sangat rendah pada slot kosong sehingga tidak memengaruhi perhitungan bobot *attention.* Tahap terakhir pada *encoder* adalah menggabungkan *(concatenate)* kedua representasi fitur sehingga diperoleh representasi visual akhir berukuran 120 × 256 yang digunakan sebagai masukan bagi *decoder*.

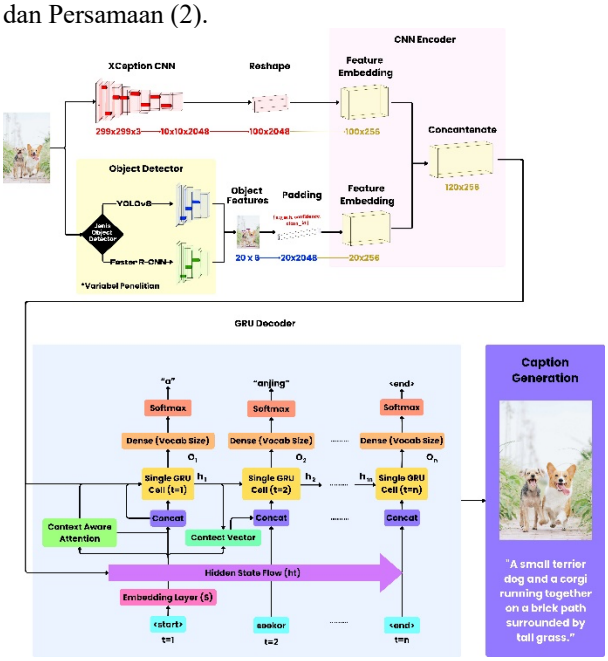
2) Decoder

Representasi visual berukuran 120 × 256 yang dihasilkan *encoder* menjadi masukan bagi *decoder* berbasis *Gated Recurrent Unit* (GRU) untuk menghasilkan *caption* kata demi kata. Pada setiap *time step*, mekanisme Bahdanau *Attention* menghitung tingkat kesesuaian antara *hidden state decoder* saat itu dengan seluruh 120 fitur visual, yang terdiri atas 100 fitur spasial dari Xception dan 20 fitur objek dari YOLOv8 atau Faster R-CNN. Hasil perhitungan tersebut menghasilkan *attention weight* berupa distribusi probabilitas yang menunjukkan tingkat relevansi masing-masing fitur terhadap kata yang sedang dihasilkan.

Setelah proses *concatenation,* seluruh 120 fitur diperlakukan sebagai satu himpunan fitur tanpa mekanisme pembeda secara eksplisit mengenai asal fitur tersebut. Dengan demikian, mekanisme *attention* tidak memilih salah satu sumber fitur secara eksklusif, melainkan mempelajari tingkat kepentingan setiap fitur secara independen melalui satu distribusi softmax. Pendekatan ini memungkinkan *context vector* yang terbentuk memanfaatkan kombinasi informasi dari fitur global maupun fitur objek sesuai dengan kebutuhan pada setiap *time step.*

Selanjutnya, *context vector* digabungkan dengan *embedding* kata sebelumnya sebagai masukan ke GRU untuk menghasilkan *hidden state* baru. *Hidden state* tersebut kemudian diproyeksikan melalui dua lapisan *fully connected* (fc1 dan fc2) untuk menghasilkan skor *(logit)* terhadap seluruh kosakata *(vocabulary)*. Kata dengan probabilitas tertinggi dipilih sebagai keluaran pada *time step* tersebut. Proses ini berlangsung secara berulang hingga model menghasilkan token <*end*> atau mencapai panjang maksimum *caption* yang telah ditentukan. Rangkaian kata yang dihasilkan dari seluruh *time step* tersebut merupakan *caption* akhir.

Diagram arsitektur keseluruhan model ditunjukkan pada Gbr 3, sedangkan formulasi matematis dari GRU dan Bahdanau *Attention* disajikan pada Persamaan (1) dan Persamaan (2).



Gbr 3 Diagram Keseluruhan Arsitektur Model

$$z_t = \sigma\,(W_z x_t + U_z h_{t-1} + b_z)$$
$$r_t = \sigma(W_r x_t + U_r h_{t-1} + b_r)$$
$$\tilde{h}_t = \tanh\,(W_h x_t + U_h\,(r_t \odot h_{t-1}) + b_h)$$
$$h_t = (1 - z_t) \odot h_{t-1} + z_t \odot \tilde{h}_t \qquad (1)$$

$$e_{ij} = score\left(h_{i-1}, a_j\right)$$
$$a_{ti} = \frac{\exp(e_{ti})}{\sum_k \exp\left(e_{tk}\right)}$$
$$= \sum_i a_{ti} v_i \quad (2)$$

E. *Skenario Eksperimen*

Penelitian ini menggunakan 18 skenario eksperimen yang diperoleh dari kombinasi tiga jenis skenario model, tiga konfigurasi *hyperparameter*, dan dua ukuran *batch size.* Seluruh eksperimen dilakukan menggunakan dataset Flickr8k.

Tabel I KONFIGURASI PELATIHAN

Variabel	Konfigurasi
Skenario Model	*Baseline, One-Stage* (YOLOv8), *Two-Stage* (Faster R-CNN)
Konfigurasi *Hyperparameter*	- Konfigurasi 1: *Dropout* = 0,3; *Teacher Forcing* (TF) *decay* = −10/*epoch*; minimum TF = 20% - Konfigurasi 2: *Dropout* = 0,4; TF *decay* = −5/*epoch*; minimum TF = 20% - Konfigurasi 3: *Dropout* = 0,4; TF *decay* = −5/*epoch;* minimum TF = 40%; *Label Smoothing* = 0,1
Batch Size	16 dan 32
Dataset	Flickr8k
3 Skenario × 3 Konfigurasi *Hyperparameter* × 2 Jenis *Batch Size* = 18 Skenario Eksperimen	

F. *Evaluasi Model*

Kinerja model *image captioning* pada penelitian ini dievaluasi menggunakan lima metrik, yaitu BLEU, METEOR, ROUGE-L, CIDEr, dan SPICE. Metrik BLEU-1 hingga BLEU-4 digunakan untuk mengukur tingkat kesesuaian n-gram antara *caption* yang dihasilkan model dengan *caption* referensi melalui perhitungan *clipped* n-gram *precision* yang dikombinasikan dengan *brevity penalty[12]*. Selanjutnya, METEOR digunakan untuk mengevaluasi kesesuaian *caption* berdasarkan *exact matching, stemming,* dan *synonym matching,* sehingga lebih mampu menangkap kesamaan semantik dibandingkan BLEU [13].

Selain itu, ROUGE-L digunakan untuk mengukur kesamaan urutan kata berdasarkan *Longest Common Subsequence* (LCS) antara *caption* hasil prediksi dan *caption* referensi[14]. Sementara itu, CIDEr mengevaluasi kesesuaian *caption* menggunakan pembobotan TF-IDF pada n-gram yang dibandingkan melalui *cosine similarity,* sehingga memberikan bobot lebih besar pada kata atau frasa yang lebih informatif dan jarang muncul[15]. Terakhir, SPICE digunakan untuk mengevaluasi kualitas semantik *caption* dengan membandingkan representasi *scene graph* yang mencakup objek, atribut, dan relasi antar objek pada *caption* hasil prediksi terhadap *caption* referensi[16]. Formulasi matematis dari masing-masing metrik evaluasi disajikan pada Persamaan (3) hingga Persamaan (7).

$$BLUE = BP.\exp(\sum_{n=1}^{N} w_n \log p_n) \quad (3)$$

$$F_{mean} = \frac{P\,.\,R}{\alpha\,.\,P + (1-\alpha)\,.\,R}\ ,$$
$$Pen = \gamma\,.frag^{\beta}\ ,$$
$$METEOR = (1 - Pen)\,.F_{mean} \quad (4)$$

$$R_{LCS} = \frac{LCS\,(X,Y)}{m}$$
$$P_{LCS} = \frac{LCS(X,Y)}{n}$$
$$F_{LCS} = \frac{(1 + \beta^2)\,R_{LCS}P_{LCS}}{R_{LCS} + \beta^2\,P_{LCS}} \quad (5)$$

$$CIDEr_n(c_i, S_i)$$
$$= \frac{1}{m}\sum_{j=1}^{m} \frac{g^n\,(c_i)\cdot g^n(s_{ij})}{\|g^n\,(c_i)\|\,\|g^n(s_{ij})\|} \quad (6)$$

$$P(c,S) = \frac{\left|T\left(G(c)\right) \otimes T\left(G(S)\right)\right|}{\left|T\left(G(c)\right)\right|}$$
$$R(c,S) = \frac{\left|T\left(G(c)\right) \otimes T\left(G(S)\right)\right|}{\left|T\left(G(S)\right)\right|}$$
$$SPICE(c,S) =$$
$$F_1(c,S) = \frac{2\,.P(c,S)\,.R(c,S)}{P(c,S) + R(c,S)} \quad (7)$$

III. HASIL DAN PEMBAHASAN

Bagian ini menyajikan hasil evaluasi seluruh eksperimen yang dilakukan untuk membandingkan kinerja tiga skenario arsitektur *image captioning*, yaitu *Baseline, One-Stage* (YOLOv8), dan *Two-Stage* (Faster R-CNN). Setiap skenario diuji menggunakan tiga konfigurasi *hyperparameter* dan dua ukuran *batch size*, sehingga diperoleh total 18 skenario eksperimen. Evaluasi dilakukan menggunakan metrik BLEU-1, BLEU-2, BLEU-3, BLEU-4, METEOR, ROUGE-L, CIDEr, dan SPICE. Hasil lengkap seluruh eksperimen disajikan pada Tabel II sebagai dasar analisis terhadap pengaruh penggunaan *object detector* serta konfigurasi pelatihan terhadap kualitas *caption* yang dihasilkan.

Tabel II HASIL EVALUASI MODEL BERDASARKAN METRIK BLEU-4, METEOR, DAN ROUGE-L

Sk	**Conf**	***Batch***	**B4**	**M**	**R**
SC1	C1	16	0.1689	0.1596	0.4302
SC1	C1	32	0.165	0.1597	0.4259
SC1	C2	16	0.1523	0.1588	0.4347
SC1	C2	32	0.1576	0.1538	0.4327
SC1	C3	16	0.1868	0.1618	0.4281
SC1	C3	32	0.1832	0.1591	0.4286
SC2	C1	16	0.1695	0.1614	0.434
SC2	C1	32	0.1687	0.1607	0.4306

SC2	C2	16	0.1472	0.1543	0.4259
SC2	C2	32	0.1616	0.1528	0.4332
SC2	C3	16	0.1837	0.1614	0.4285
SC2	C3	32	0.1731	0.1547	0.4215
SC3	C1	16	0.1716	0.1652	0.4367
SC3	C1	32	0.1692	0.1656	0.4328
SC3	C2	16	0.1535	0.1564	0.4329
SC3	C2	32	0.1544	0.155	0.4343
SC3	C3	16	0.1937	0.1734	0.4396
SC3	C3	32	0.17	0.16	0.4209

Tabel III HASIL EVALUASI MODEL BERDASARKAN METRIK CIDER DAN SPICE

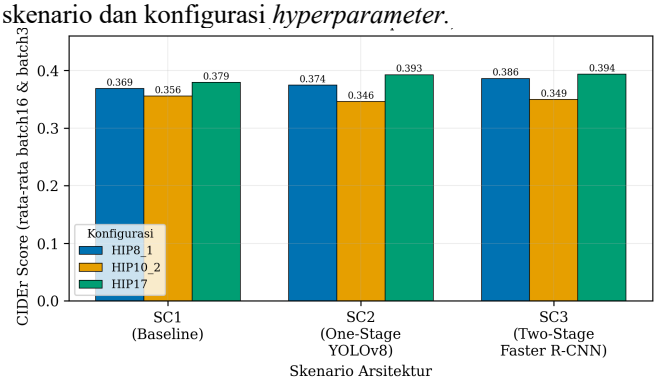
Sk	Conf	*Batch*	C	S
SC1	C1	16	0.3774	0.1151
SC1	C1	32	0.3597	0.1117
SC1	C2	16	0.3652	0.1074
SC1	C2	32	0.346	0.1048
SC1	C3	16	0.3883	0.1193
SC1	C3	32	0.3703	0.1145
SC2	C1	16	0.381	0.1173
SC2	C1	32	0.3678	0.1168
SC2	C2	16	0.3447	0.1042
SC2	C2	32	0.3474	0.1035
SC2	C3	16	0.4285	0.1214
SC2	C3	32	0.3567	0.1151
SC3	C1	16	0.3936	0.1186
SC3	C1	32	0.3784	0.1187
SC3	C2	16	0.3509	0.1065
SC3	C2	32	0.348	0.1051
SC3	C3	16	0.42	0.1268
SC3	C3	32	0.3673	0.1162

Keterangan:
- Sk = Skenario
- SC1 = *Baseline*
- SC2 = *One-Stage* (YOLOv8)
- SC3 = *Two-Stage* (Faster R-CNN)
- Conf = Konfigurasi *hyperparameter*
- B4 = BLEU-4
- M = METEOR
- R = ROUGE-L
- C = CIDEr
- S = SPICE

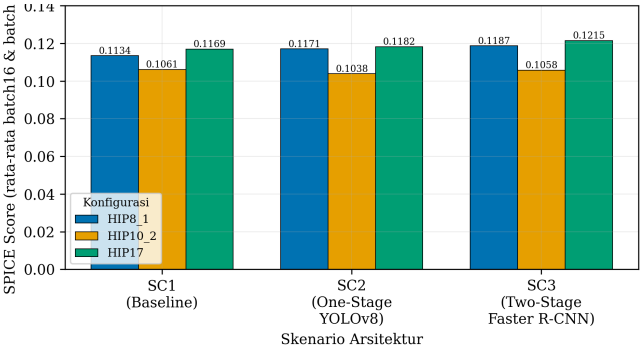
Berdasarkan Tabel II dan Tabel III, terlihat bahwa kinerja model dipengaruhi oleh kombinasi skenario arsitektur, konfigurasi *hyperparameter*, dan *batch size*. Secara umum, Konfigurasi 3 dengan *batch size* 16 menghasilkan performa terbaik pada ketiga skenario. Pada metrik BLEU-4, METEOR, dan ROUGE-L, skenario *Two-Stage* (Faster R-CNN) mencapai nilai tertinggi masing-masing sebesar 0,1937, 0,1734, dan 0,4396. Sementara itu, pada metrik CIDEr dan SPICE, skenario *One-Stage* (YOLOv8) memperoleh nilai CIDEr tertinggi sebesar 0,4285, sedangkan nilai SPICE tertinggi diperoleh skenario *Two-Stage* (Faster R-CNN) sebesar 0,1268. Temuan ini menunjukkan bahwa penggunaan fitur objek memberikan pengaruh yang berbeda pada setiap metrik evaluasi sehingga diperlukan analisis lebih lanjut untuk menjelaskan faktor-faktor yang menyebabkan perbedaan tersebut.

A. *Perbandingan Kinerja Antar-Skenario Arsitektur*

Gbr. 4 dan Gbr. 5 menyajikan skor CIDEr dan SPICE rata-rata (agregat batch16 dan batch32) untuk tiap kombinasi skenario dan konfigurasi *hyperparameter*.



Gbr 4 Perbandingan CIDEr Antar Skenario dan Konfigurasi *Hyperparameter*



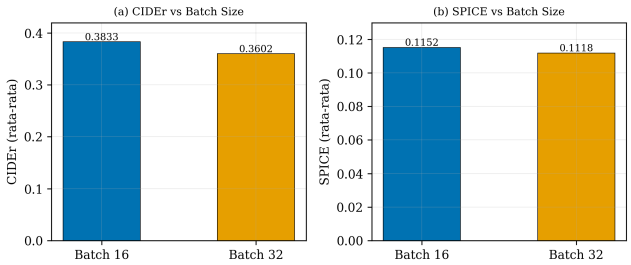
Gbr 5 Perbandingan SPICE Antar Skenario dan Konfigurasi *Hyperparameter*

Tabel IV RATA-RATA KINERJA PER SKENARIO

Skenario	CIDEr (mean±std)	SPICE (mean±std)
SC1 (Baseline)	0.3678 ± 0.0146	0.1121 ± 0.0053
SC2 (YOLOv8)	0.3710 ± 0.0312	0.1131 ± 0.0074
SC3 (Faster R-CNN)	0.3764 ± 0.0274	0.1153 ± 0.0082

Berdasarkan Gbr 4, Gbr 5, dan Tabel IV, SC3 (*Two-Stage* Faster R-CNN) memperoleh rata-rata kinerja tertinggi dengan nilai CIDEr 0,3764 ± 0,0274 dan SPICE 0,1153 ± 0,0082, diikuti oleh SC2 (*One-Stage* YOLOv8) dan SC1 *(Baseline)*. Namun, selisih rata-rata antar-skenario relatif kecil, yaitu sekitar 2,3% pada metrik CIDEr antara SC3 dan SC1. Selain itu, seluruh skenario menunjukkan pola yang serupa, yaitu Konfigurasi 3 (HIP17) secara konsisten menghasilkan skor tertinggi, sedangkan Konfigurasi 2 (HIP10) menghasilkan skor terendah. Temuan ini mengindikasikan bahwa selain jenis *object detector*, konfigurasi pelatihan juga berkontribusi terhadap kinerja model dan akan dibahas lebih lanjut pada bagian berikutnya.

B. *Pengaruh Ukuran Batch*



Gbr 6 . Pengaruh ukuran batch terhadap CIDEr dan SPICE

Gbr 6 menunjukkan bahwa penggunaan *batch size* 16 secara konsisten menghasilkan kinerja yang lebih baik dibandingkan *batch size* 32. Secara rata-rata, *batch size* 16 memperoleh skor CIDEr sebesar 0,3833 dan SPICE sebesar 0,1152, lebih tinggi dibandingkan *batch size* 32 yang masing-masing mencapai 0,3602 dan 0,1118. Selain itu, proses pelatihan dengan *batch size* 16 juga menghasilkan *training loss* yang lebih rendah (2,476 dibandingkan 2,703). Hasil uji statistik menunjukkan bahwa perbedaan tersebut memiliki ukuran efek yang besar (Cohen's d = 1,068; p = 0,038), yang mengindikasikan bahwa penggunaan *batch size* yang lebih kecil memberikan peningkatan kinerja yang signifikan pada dataset Flickr8k.

C. *Pengaruh Konfigurasi Hyperparameter*

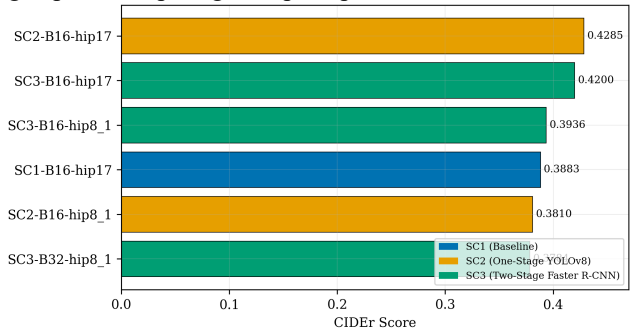
Tabel V KINERJA RATA-RATA PER KONFIGURASI HYPERPARAMETER

Konfigurasi	CIDEr	SPICE	Loss
HIP8_1 (C1)	0.3763	0.1164	2.457
HIP10_2 (C2)	0.3504 ↓	0.1053 ↓	2.493
HIP17 (C3)	0.3885 ↑	0.1189 ↑	2.818 ↑

Tabel V menunjukkan bahwa konfigurasi 3 (HIP17) menghasilkan kinerja terbaik dengan skor CIDEr sebesar 0,3885 dan SPICE sebesar 0,1189, sedangkan konfigurasi 2 (HIP10_2) memberikan performa terendah pada kedua metrik tersebut. Menariknya, konfigurasi 3 juga menghasilkan nilai *training loss* tertinggi (2,818). Kondisi ini disebabkan oleh penerapan *label smoothing* (ε = 0,1), yang mengubah distribusi target dari *one-hot* menjadi distribusi yang lebih halus sehingga nilai *cross-entropy loss* cenderung lebih tinggi dibandingkan tanpa *label smoothing*. Oleh karena itu, nilai *loss* yang lebih besar pada konfigurasi 3 tidak menunjukkan penurunan kualitas model, melainkan merupakan konsekuensi dari fungsi objektif yang digunakan. Hal ini tercermin dari peningkatan skor CIDEr dan SPICE, yang menunjukkan bahwa model menghasilkan *caption* dengan kualitas semantik yang lebih baik.

D. *Konfigurasi Terbaik*

Tabel VI menunjukkan bahwa konfigurasi dengan skor CIDEr tertinggi diperoleh oleh SC2 (YOLOv8) – Batch16 – HIP17 dengan nilai 0,4285, sedangkan skor SPICE tertinggi diperoleh oleh SC3 (Faster R-CNN) – Batch16 – HIP17 dengan nilai 0,1268. Meskipun skenario arsitektur yang digunakan berbeda, kedua konfigurasi terbaik memiliki kombinasi *hyperparameter* yang sama, yaitu Batch16 dan HIP17. Temuan ini mengindikasikan bahwa kombinasi *hyperparameter* tersebut memberikan kontribusi yang konsisten terhadap peningkatan kualitas *caption* pada metrik CIDEr maupun SPICE. Selain itu, Gbr 7 menunjukkan bahwa lima dari enam konfigurasi terbaik berasal dari konfigurasi 3 (HIP17) dan konfigurasi 1 (HIP8_1), sedangkan konfigurasi 2 (HIP10_2) tidak termasuk dalam enam konfigurasi dengan skor CIDEr tertinggi. Hasil ini memperkuat temuan pada Tabel V bahwa Konfigurasi 3 (HIP17) merupakan konfigurasi pelatihan dengan performa paling baik pada penelitian ini.

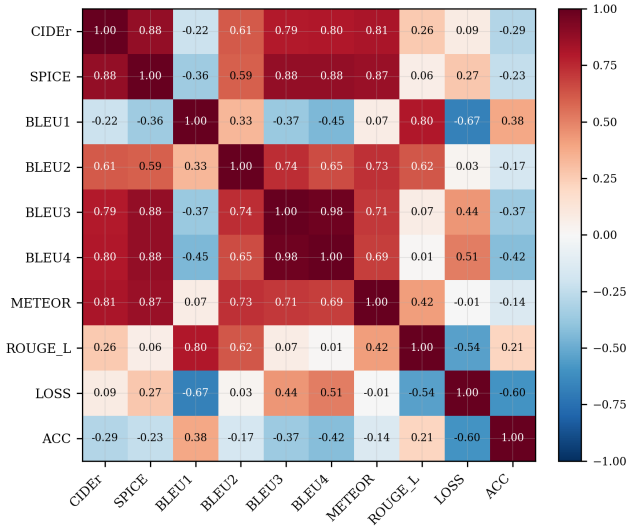


Gbr 7 Enam konfigurasi terbaik berdasarkan CIDEr

Tabel VI KONFIGURASI TERBAIK BERDASARKAN CIDER DAN SPICE

Peringkat	Konfigurasi	CIDEr	SPICE
1 (CIDEr)	SC2 (YOLOv8) – Batch16 – hip17	0.4285	0.1214
1 (SPICE)	SC3 (Faster R-CNN) – Batch16 – hip17	0.4200	0.1268

E. *Korelasi Antar-Metrik Evaluasi*

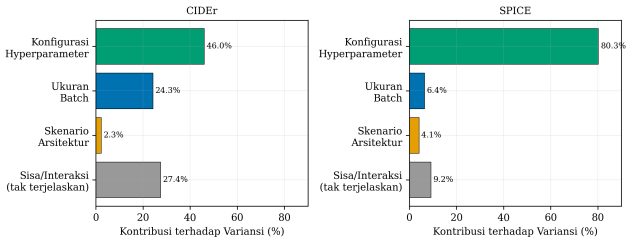


Gbr 8 Matriks korelasi Pearson antar-metrik evaluasi

Analisis korelasi pada Gbr 8 menunjukkan bahwa CIDEr memiliki korelasi yang sangat lemah dengan *training loss* (r = 0,085) serta berkorelasi negatif dengan BLEU-1 (r = −0,218) dan akurasi token (ACC, r = −0,292). Sebaliknya, CIDEr berkorelasi kuat dengan SPICE (r = 0,878), METEOR (r = 0,815), dan BLEU-4 (r = 0,800). Hasil ini menunjukkan bahwa

metrik yang mengevaluasi kesesuaian semantik dan struktur kalimat memiliki hubungan yang lebih erat dengan kualitas *caption* dibandingkan metrik berbasis akurasi token atau *unigram matching.* Temuan tersebut mendukung penggunaan CIDEr dan SPICE sebagai metrik utama dalam mengevaluasi model *image captioning*, karena keduanya lebih mampu merepresentasikan kualitas deskripsi yang dihasilkan.

F. *Dekomposisi Variansi (ANOVA Dua Arah)*



Gbr 9 Dekomposisi variansi CIDEr dan SPICE berdasarkan ANOVA dua arah

Tabel VII KONTRIBUSI VARIANSI ANTARFAKTOR (R^2 CIDER=0.726; R^2 SPICE=0.908)

Faktor	K.CIDEr	K.SPICE	S.(p)
Konfigurasi Hyperparameter	46,0%	80,3%	CIDEr p=0.0027 SPICE p<0.0001
Ukuran Batch	24,3%	6,4%	CIDEr p=0.0069 SPICE p=0.0134
Skenario Arsitektur	2,3%	4,1%	tidak signifikan (p=0.62; p=0.11)
Sisa/interaksi tak terjelaskan	27,4%	9,2%	—

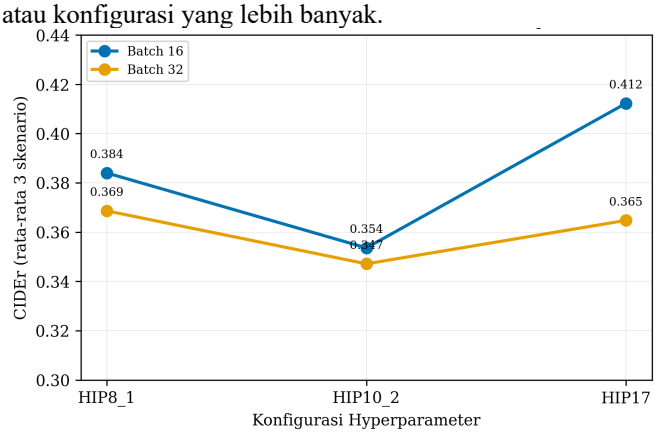
Keterangan:
- K.CIDEr = Kontribusi CIDEr
- K.SPICE = Kontribusi SPICE
- S(p) = Signifikansi (p)

Gbr 9 dan Tabel VII menunjukkan bahwa konfigurasi *hyperparameter* merupakan faktor yang memberikan kontribusi terbesar terhadap variasi kinerja model, dengan menjelaskan 46,0% variansi CIDEr dan 80,3% variansi SPICE ($p = 0{,}0027$ dan $p < 0{,}0001$). Sebaliknya, ukuran *batch* memberikan kontribusi sedang, yaitu 24,3% terhadap CIDEr dan 6,4% terhadap SPICE, serta berpengaruh signifikan pada kedua metrik ($p < 0{,}05$). Adapun skenario arsitektur hanya berkontribusi 2,3% variansi CIDEr dan 4,1% variansi SPICE, dengan pengaruh yang tidak signifikan secara statistik ($p > 0{,}05$).

Hasil tersebut menunjukkan bahwa strategi pelatihan memiliki pengaruh yang lebih besar terhadap performa *image captioning* dibandingkan pemilihan *object detector* yang digunakan. Dengan kata lain, optimasi *hyperparameter* memberikan dampak yang lebih besar terhadap peningkatan kualitas *caption* dibandingkan perbedaan antar skenario *Baseline*, YOLOv8, dan Faster R-CNN pada dataset Flickr8k. Temuan ini juga sejalan dengan hasil eksperimen ablasi sebelumnya yang menunjukkan bahwa konfigurasi pelatihan merupakan faktor dominan dalam menentukan performa model, meskipun proporsi kontribusinya berbeda akibat perbedaan variabel dan rancangan eksperimen yang dianalisis.

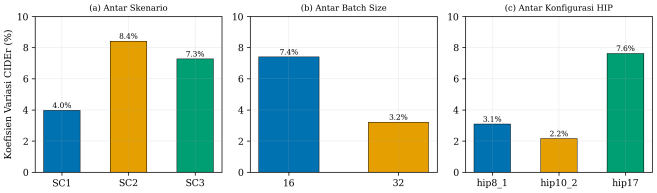
G. *Interaksi Hyperparameter × Ukuran Batch*

Gbr 10 menunjukkan adanya pola interaksi antara konfigurasi *hyperparameter* dan ukuran *batch* terhadap skor CIDEr. Perbedaan kinerja antara batch16 dan batch32 relatif kecil pada konfigurasi 1 (0,384 vs 0,369) dan konfigurasi 2 (0,354 vs 0,347), namun meningkat pada konfigurasi 3 (0,412 vs 0,365). Hasil ini mengindikasikan bahwa peningkatan performa yang diperoleh melalui konfigurasi 3 lebih optimal ketika menggunakan batch16. Meskipun demikian, hasil ANOVA menunjukkan bahwa efek interaksi belum signifikan secara statistik ($p = 0{,}136$), meskipun berkontribusi sebesar 14,1% terhadap variansi CIDEr. Oleh karena itu, pola interaksi ini perlu diinterpretasikan secara hati-hati dan memerlukan verifikasi lebih lanjut pada eksperimen dengan jumlah sampel atau konfigurasi yang lebih banyak.



Gbr 10 Plot interaksi konfigurasi hyperparameter dan ukuran batch terhadap CIDEr.

H. *Stabilitas Kinerja Antarlevel Faktor*



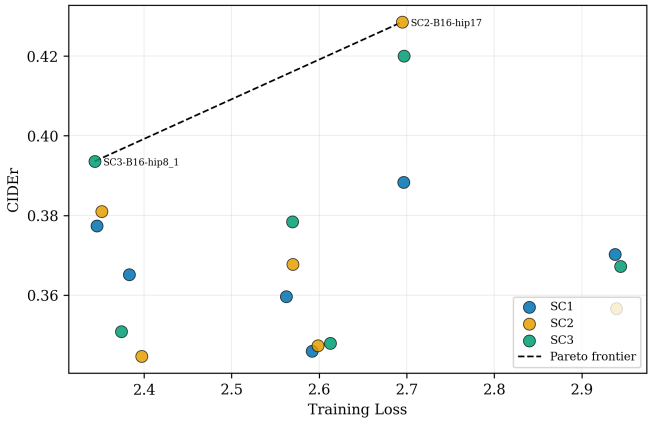
Gbr 11 Koefisien variasi (CV) CIDEr antarlevel skenario, batch, dan konfigurasi hyperparameter.

Gbr 11 menunjukkan koefisien variasi (*Coefficient of Variation/CV)* skor CIDEr pada setiap faktor eksperimen. Pada level skenario, SC1 (*Baseline*) memiliki variasi paling rendah (CV = 3,97%), sehingga menunjukkan kinerja yang paling konsisten di berbagai konfigurasi. Pada level *batch size*, *batch* 32 memiliki variasi yang lebih kecil (CV = 3,21%)

dibandingkan batch16 (CV = 7,40%), meskipun batch 16 menghasilkan rata-rata skor CIDEr yang lebih tinggi. Sementara itu, pada level konfigurasi *hyperparameter*, Konfigurasi 2 merupakan konfigurasi yang paling stabil, tetapi menghasilkan performa terendah, sedangkan konfigurasi 3 mencapai performa terbaik dengan variasi terbesar (CV = 7,62%). Temuan ini menunjukkan adanya *trade-off* antara performa dan konsistensi, di mana konfigurasi dengan kinerja tertinggi cenderung memiliki variasi hasil yang lebih besar dibandingkan konfigurasi yang lebih stabil.

I. *Analisis Pareto Frontier*

Analisis Pareto pada Gbr 12 menunjukkan bahwa dari 18 konfigurasi yang dievaluasi, hanya dua konfigurasi yang berada pada Pareto frontier, yaitu SC3–Batch16–HIP8_1 dan SC2–Batch16–HIP17. Konfigurasi pertama menawarkan *training loss* yang lebih rendah dengan skor CIDEr yang tetap kompetitif, sedangkan konfigurasi kedua menghasilkan skor CIDEr tertinggi dengan konsekuensi nilai *training loss* yang lebih besar. Sebaliknya, 16 konfigurasi lainnya didominasi *(dominated)*, karena terdapat konfigurasi lain yang memberikan kombinasi *loss* dan CIDEr yang lebih baik secara bersamaan. Temuan ini menunjukkan bahwa kedua konfigurasi tersebut merupakan pilihan yang paling efisien apabila mempertimbangkan kinerja model dan biaya optimasi secara simultan.



Gbr 12 *Trade-off training loss* vs CIDEr dan Pareto frontier dari 18 konfigurasi.

J. *Pembahasan Umum*

Secara keseluruhan, hasil penelitian menunjukkan bahwa SC3 (Faster R-CNN) memperoleh kinerja rata-rata tertinggi pada metrik CIDEr dan SPICE, sedangkan SC2 (YOLOv8) menghasilkan skor CIDEr tertinggi pada satu konfigurasi eksperimen. Meskipun demikian, perbedaan kinerja antar-skenario relatif kecil dan tidak signifikan secara statistik. Oleh karena itu, Faster R-CNN lebih tepat dipandang sebagai skenario yang memberikan performa lebih konsisten, bukan sebagai arsitektur yang unggul secara mutlak dibandingkan YOLOv8 maupun *baseline.*

Pada penerapanya, hasil ini menunjukkan bahwa pada dataset Flickr8k, pemilihan YOLOv8 atau Faster R-CNN dapat mempertimbangkan aspek efisiensi komputasi selain kualitas *caption* yang dihasilkan. Dengan perbedaan performa yang relatif kecil, peningkatan kualitas *caption* pada penelitian ini lebih banyak dipengaruhi oleh konfigurasi *hyperparameter* dan ukuran *batch* dibandingkan oleh pemilihan jenis *object detector.*

IV. KESIMPULAN

Penelitian ini membandingkan kinerja model *image captioning* berbasis Bahdanau *attention* dengan tiga skenario integrasi fitur objek (*baseline*, YOLOv8, Faster R-CNN) pada 18 eksperimen dataset Flickr8k. Hasil ANOVA dua arah menunjukkan bahwa konfigurasi *hyperparameter* merupakan faktor paling dominan (46,0% variansi CIDEr; 80,3% variansi SPICE), diikuti ukuran *batch* (24,3%; 6,4%), sedangkan pemilihan arsitektur detektor objek tidak signifikan secara statistik (2,3%; 4,1%; p>0,05).

Konfigurasi terbaik diperoleh pada kombinasi *batch* 16 dan hip17 *(label smoothing)*, dengan CIDEr tertinggi 0,4285 pada skenario SC2 (YOLOv8), sedangkan SC3-Batch16-hip8_1 menawarkan *trade-off loss*-CIDEr paling efisien pada analisis *Pareto frontier*. *Batch* 16 secara konsisten mengungguli *batch* 32 pada seluruh 18 titik data dengan *effect size* besar (Cohen's d=1,068).

Temuan ini memberi implikasi praktis bahwa upaya peningkatan kualitas *caption* untuk kebutuhan pengelolaan metadata gambar otomatis pada perusahaan sebaiknya diprioritaskan pada konfigurasi *hyperparameter* dan ukuran *batch*, sementara pemilihan jenis detektor objek dapat didasarkan pada pertimbangan efisiensi komputasi tanpa mengorbankan kualitas *caption* secara signifikan.

V. SARAN

Beberapa saran untuk penelitian selanjutnya adalah sebagai berikut.

- Memvalidasi kembali hierarki pengaruh faktor pelatihan (konfigurasi *hyperparameter*, ukuran *batch*, dan skenario arsitektur) pada dataset yang lebih besar, seperti Flickr30k dan MS-COCO, sehingga konsistensi temuan dapat dievaluasi pada berbagai skala dan karakteristik dataset.
- Melakukan pengujian asumsi ANOVA, seperti normalitas residual dan homogenitas varians, terutama pada penelitian dengan ukuran sampel yang relatif kecil, agar hasil analisis statistik, khususnya pada efek interaksi antar faktor, memiliki tingkat keandalan yang lebih tinggi.
- Mengeksplorasi mekanisme penggabungan fitur selain *concatenation*, misalnya *cross-attention* yang memproses fitur CNN dan fitur objek secara terpisah sebelum integrasi, untuk mengevaluasi apakah pendekatan tersebut mampu meningkatkan pemanfaatan informasi visual dan memperjelas kontribusi masing-masing sumber fitur terhadap kinerja model *image captioning.*
- Mengevaluasi model menggunakan data gambar perusahaan yang lebih beragam, baik dari sisi kualitas citra, sudut pengambilan, maupun kompleksitas latar belakang, guna menilai kemampuan generalisasi model pada kondisi yang lebih mendekati aplikasi di dunia nyata.

UCAPAN TERIMA KASIH

Penulis mengucapkan puji syukur kepada Tuhan Yang Maha Esa atas segala rahmat dan karunia-Nya sehingga penelitian ini dapat diselesaikan dengan baik. Penulis juga menyampaikan terima kasih kepada dosen pembimbing atas bimbingan, arahan, dan masukan yang diberikan selama proses penelitian, kepada pihak universitas atas dukungan akademik selama penyusunan penelitian, kepada kedua orang tua atas doa, dukungan, dan motivasi yang tiada henti, serta kepada teman-teman terdekat yang senantiasa memberikan semangat, bantuan, dan dukungan selama penyusunan penelitian ini.

REFERENSI

[1] J. M. Tomczak, "Deep Generative Modeling for Neural Compression," in *Deep Generative Modeling*, J. M. Tomczak, Ed., Cham: Springer International Publishing, 2022, pp. 173–188. doi: 10.1007/978-3-030-93158-2_8.

[2] E. Late, H. Ruotsalainen, and S. Kumpulainen, "Image searching in an open photograph archive: search tactics and faced barriers in historical research," *International Journal on Digital Libraries*, vol. 25, no. 4, pp. 715–728, 2024, doi: 10.1007/s00799-023-00390-1.

[3] Y. Hu, Z. Gui, J. Wang, and M. Li, "Enriching the metadata of map images: a deep learning approach with GIS-based data augmentation," *International Journal of Geographical Information Science*, vol. 36, no. 4, pp. 799–821, Apr. 2022, doi: 10.1080/13658816.2021.1968407.

[4] A. Krishnan, S. Rajesh, and S. S. S, "Text-based Image Retrieval Using Captioning," in *2021 Fourth International Conference on Electrical, Computer and Communication Technologies (ICECCT)*, 2021, pp. 1–5. doi: 10.1109/ICECCT52121.2021.9616897.

[5] A. Verma, A. K. Yadav, M. Kumar, and D. Yadav, "Automatic image caption generation using deep learning," *Multimed. Tools Appl.*, vol. 83, no. 2, pp. 5309–5325, 2024, doi: 10.1007/s11042-023-15555-y.

[6] A. Hegde, Y. Kuppala, and M. Ma, "FashionLIP: Automated Fashion Image Captioning Using Cascaded Contrastive Learning," in *2025 IEEE 8th International Conference on Multimedia Information Processing and Retrieval (MIPR)*, 2025, pp. 97–103. doi: 10.1109/MIPR67560.2025.00024.

[7] D. Abdal Hafeth and S. Kollias, "Insights into Object Semantics: Leveraging Transformer Networks for Advanced Image Captioning," *Sensors*, vol. 24, no. 6, Mar. 2024, doi: 10.3390/s24061796.

[8] L. Li, Y. Wei, and P. Ren, "Underwater Image Captioning Based on Feature Fusion," in *Proceedings of the 2024 7th International Conference on Image and Graphics Processing*, in ICIGP '24. New York, NY, USA: Association for Computing Machinery, 2024, pp. 322–326. doi: 10.1145/3647649.3647700.

[9] M. A. Al-Malla, A. Jafar, and N. Ghneim, "Image captioning model using attention and object features to mimic human image understanding," *J. Big Data*, vol. 9, no. 1, Dec. 2022, doi: 10.1186/s40537-022-00571-w.

[10] D. Das, M. R. Bayesh, M. A. Islam, and M. A. Rahman, "A Framework for Accurate and Efficient Image Captioning by Fusing Fine-Tuned YOLOv8 and LLMs," in *2025 International Conference on Quantum Photonics, Artificial Intelligence, and Networking (QPAIN)*, 2025, pp. 1–6. doi: 10.1109/QPAIN66474.2025.11171749.

[11] P. Goswami, L. Aggarwal, A. Kumar, R. Kanwar, and U. Vasisht, "Real-time evaluation of object detection models across open world scenarios," *Appl. Soft Comput.*, vol. 163, p. 111921, 2024, doi: https://doi.org/10.1016/j.asoc.2024.111921.

[12] I. Albadarneh, B. Hammo, and O. Al-Kadi, "Attention-based transformer models for image captioning across languages: An in-depth survey and evaluation," *ArXiv*, vol. abs/2506.05399, p., 2025, doi: 10.1016/j.cosrev.2025.100766.

[13] G. Luo, L. Cheng, C. Jing, C. Zhao, and G.-Z. Song, "A thorough review of models, evaluation metrics, and datasets on image captioning," *IET Image Process.*, vol. 16, pp. 311–332, 2021, doi: 10.1049/ipr2.12367.

[14] O. González-Chávez, G. Ruiz, D. Moctezuma, and T. A. Ramirez-delReal, "Are metrics measuring what they should? An evaluation of image captioning task metrics," Feb. 2023, [Online]. Available: http://arxiv.org/abs/2207.01733

[15] S. Lee *et al.*, "A Survey on Evaluation Metrics for Machine Translation," *Mathematics*, vol. 11, no. 4, Feb. 2023, doi: 10.3390/math11041006.

[16] T. Ghandi, H. Pourreza, and H. Mahyar, "Deep Learning Approaches on Image Captioning: A Review," *ACM Comput. Surv.*, vol. 56, pp. 1–39, 2022, doi: 10.1145/3617592.