Image Captioning Menggunakan Metode Inception-V3 dan Transformer

Vidha Rossa Pratiwi, Jasman Pardede

Institut Teknologi Nasional (Itenas) Bandung
Email: vidharossa13@mhs.itenas.ac.id

Received DD MM YYYY **|** *Revised* DD MM YYYY **|** *Accepted* DD MM YYYY

ABSTRAK

Pada bidang Computer Vision terdapat masalah yang muncul, seperti objek yang dideteksi pada gambar tidak dapat memberikan pemahaman secara konteks. Dengan memanfaatkan object detection yang telah digunakan sebelumnya, hal tersebut dapat dimanfaatkan untuk menghasilkan satu atau beberapa kalimat yang mendeskripsikan konteks gambar. Hal ini disebut Image Captioning yang merupakan proses menghasilkan teks deskripsi yang diberikan pada suatu gambar. Untuk melakukan Image Captioning dibutuhkan dua ilmu yaitu Computer Vision untuk mengenali objek dan Natural Language Processing (NLP) untuk menghasilkan kalimat deskripsi. Metode yang digunakan pada penelitian ini yaitu Inception-V3 dan Transformer. Penelitian dilakukan menggunakan dataset Flickr8k yang memiliki 8000 gambar dan 40000 kalimat caption. Model dievaluasi dengan cara menghitung skor BLEU. Berdasarkan model tersebut, nilai rata-rata skor BLEU-1, BLEU-2, BLEU-3, dan BLEU-4 yang didapatkan adalah (0.306, 0.184, 0.123, 0.084).

***Kata kunci**: Inception-V3, Transformer, BLEU, Computer Vision, Natural Language Processing (NLP)*

ABSTRACT

Some problems arise as the field of computer vision advances, such as objects detected in images cannot provide contextual understanding. Utilizing the previously used object detection, it can generate one or more sentences describing the image context. It is known as image captioning, the process of producing descriptive text for an image based on what someone sees. Two fields are required for image captioning: computer vision to recognize objects and natural language processing (NLP) to produce descriptive sentences. The study's methods used were Inception-V3 and Transformer. The study was conducted using the Flickr8k dataset, which contains 8000 images and 40000 caption sentences. The model was evaluated by calculating the BLEU (Bilingual Evaluation Understudy) score. Based on the model, the obtained average scores of BLEU-1, BLEU-2, BLEU-3, and BLEU-4 using the model were (0.306, 0.184, 0.123, 0.084).

***Keywords**: Inception-V3, Transformer, BLEU, Computer Vision, Natural Language Processing (NLP)*

1. PENDAHULUAN

Computer vision adalah salah satu bidang teknologi dan ilmu pengetahuan yang mengalami kemajuan yang signifikan, seperti pada area *image processing* terdapat *image classification* dan *object detection*. Tetapi dengan seiringnya kemajuan pada *computer vision* terdapat juga beberapa masalah yang muncul, seperti gambar yang telah diklasifikasikan atau objek yang dideteksi pada gambar tersebut tidak dapat dipahami seperti apa konten dari gambar tersebut. Maka dari itu, dengan memanfaatkan *image classification* dan *object detection* yang telah digunakan sebelumnya dapat secara otomatis menghasilkan satu atau beberapa kalimat untuk mendeskripsikan atau memahami isi dari suatu gambar. Hal ini dapat disebut sebagai *image captioning* (**Liu et al., 2018**)

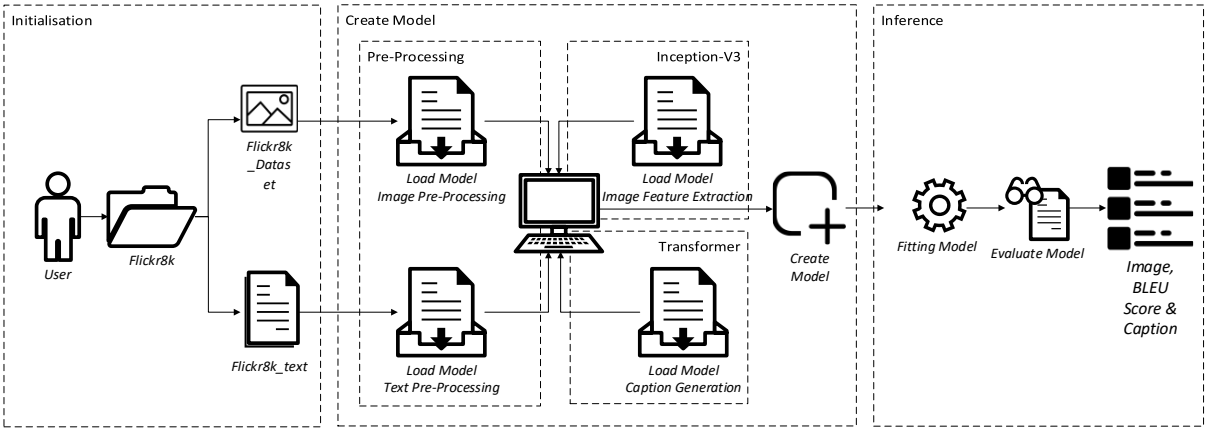
Image captioning adalah teks singkat atau deskripsi yang diberikan pada suatu citra gambar berdasarkan apa yang dilihat oleh seseorang. Tujuan dilakukannya *image captioning* yaitu untuk memberikan pemahaman terhadap informasi yang ingin disampaikan oleh gambar tersebut (**Nursikuwagus et al., 2020**). Terdapat banyak sekali pengaplikasian dari *image captioning* ini, seperti penggunan dalam asisten virtual, membantu tunanetra, interaksi manusia dan komputer, dan beberapa aplikasi *natural language processing* (NLP) lainnya (**Srinivasan et al., 2018**)

Image captioning dapat dilakukan dengan menggunakan gabungan dari dua metode yang berbeda, seperti dari *computer vision* dan *natural language process*sing (NLP) (**Vinyals et al., 2015**). *Computer vision* digunakan untuk mengenali objek pada gambar dengan menggunakan *Convolutional Neural Network* (CNN), dan NLP digunakan untuk menghasilkan deskripsi atau *caption* dari gambar tersebut berupa bahasa alami dengan menggunakan *Recurrent Neural Network* (RNN) (**Devlin et al., 2015**). Tetapi, terdapat permasalahan pada RNN yang diusulkan seperti terjadinya *vanishing gradient*, adanya *recurrent* yang dapat mencegah komputasi paralel, dan ketergantungan jarak jauh. Masalah-masalah tersebut dapat diatasi dengan menggunakan Transformer sebagai pengganti RNN (**Vaswani et al., 2017**) . Penelitian ini menggunakan teknologi dari *computer vision* yaitu metode Inception-V3 dan NLP yaitu metode Transformer. Untuk menguji seberapa baik hasil dari program yang telah dibuat, hasil dari *image captioning* akan dievaluasi dengan menggunakan BLEU (*Bilingual Evaluation Understudy*). BLEU adalah sebuah algoritma yang digunakan untuk mengevaluasi kualitas hasil dari mesin terjemahan (**Ardhi et al., 2018**) dan juga untuk model *image captioning* (**Al-faruq, 2021**). Nilai yang dihasilkan dari BLEU ini berada diantara 0 sampai 1, yang berarti semakin tinggi nilai BLEU, maka semakin baik model tersebut.

2. METODOLOGI PENELITIAN

2.1. Perancangan Umum

Perancangan sistem dibagi menjadi 6 bagian, yaitu *initialization, pre-processing, image feature extraction, caption generation, create model*, dan *inference*. Alur kerja sistem penelitian secara umum digambarkan menggunakan blok diagram yang ditunjukkan pada Gambar 1



Gambar 1. Blok Diagram Penelitian

Berikut merupakan penjelasan dari blok diagram penelitian yang dilakukan:
Pada tahap *initialization user* mempersiapkan dataset *Flickr8k* yang terdiri dari dua jenis data, yaitu *Flicker8k_Dataset* yang berisi data gambar dan *Flicker8k_Text* yang berisi data *caption.* Setelah data berhasil didapatkan lalu data gambar ditampilkan beserta dengan 5 caption yang mendeskripsikan gambar. Kemudian pada bagian *pre-processing* dilakukan *text pre-processing* terhadap data *caption* yang meliputi *case folding*, *remove punctuation*, *remove number*, *remove single character*, *tokenizing*, dan *mark caption*. Lalu dilakukan *image pre-processing* pada data gambar yang meliputi mengubah ukuran gambar sesuai dengan *input* model Inception-v3. Selanjutnya membuat model menggunakan Inception-V3 untuk *image feature extraction* dan Transformer untuk *caption generation*. Setelah model berhasil dibuat, selanjutnya dilakukan *model fitting* dengan melakukan *epoch* sebanyak 100 kali untuk melatih model yang dibuat. Model yang sudah dilatih lalu dievaluasi menggunakan BLEU *scoring*. Hasil dari *caption generation* dan evaluasi ditampilkan berupa kalimat *caption* yang diprediksi dan skor BLEU bagi setiap gambar

2.2. Dataset

Dataset yang digunakan adalah Flick8k yang diambil dari situs kaggle.com. *Dataset* ini memiliki sebanyak 8091 gambar dan 40455 kalimat caption. Setiap gambar memiliki 5 kalimat caption yang mendeskripsikan gambar. Pada penelitian ini digunakan sebanyak 8000 data gambar dan 40000 kalimat referensi. Untuk membagi *data training* dan *data validation, dataset* dibagi dengan perbandingan 8:2, yaitu sebanyak 6400 gambar dengan 32000 kalimat *caption training* dan 1600 gambar dengan 8000 kalimat *caption validation*.

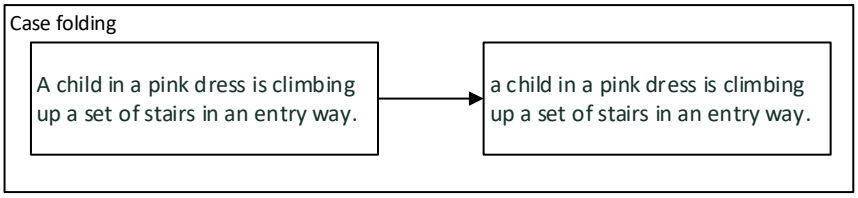
2.3. Pre-Processing

Image Pre-processing data merupakan standar operasi untuk melakukan pelatihan data pada neural network untuk menyeragamkan ukuran citra yang dilakukan terhadap data gambar, dengan melakukan *resize* atau memperkecil ukuran gambar menjadi ukuran tertentu (x*y) dengan arah horizontal (x) dan vertikal (y) (**Tammina, 2019**). *Pre-processing* yang dilakukan pada gambar terdiri dari 2 proses yaitu mengubah ukuran gambar menjadi 299 x 299 x 3 dan menormalisasi gambar agar nilai piksel *input* memiliki nilai diantara -1 hingga 1. Hal tersebut dilakukan agar data *input* sesuai dengan ketentuan *input* dari Inception-V3.

Text pre-processing dilakukan untuk mereduksi beberapa bentuk kata menjadi satu bentuk yang dapat diprediksi dan dianalisis untuk tugas tertentu (**Kadhim, 2018**). *Pre-processing* yang dilakukan pada *caption* terdiri dari 6 proses yaitu:

a. Case Folding

Case Folding adalah proses mengubah semua huruf kapital (*upper case*) menjadi huruf kecil (*lower case*).

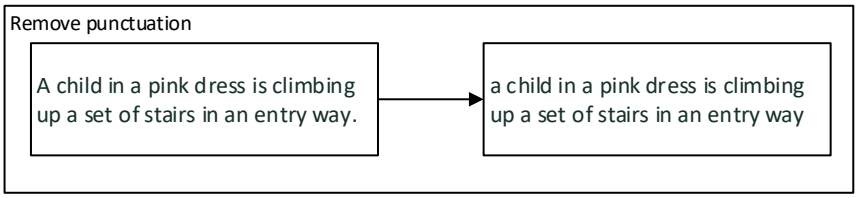


Gambar 2. Tahap Case Folding

Pada Gambar 2 teks input "*A child in a pink dress is climbing up a set of staird in an entry way.*" dimana kata "A" berubah menajdi "a".

b. Remove Punctuation

Remove punctuation adalah proses menghapus tanda baca seperti '!"#$%&()*+.,-/:;=?@[\]^_`{|}~'. Tanda baca atau *special character* tidak menambah nilai pada pemahaman teks dan dapat menyebabkan *noise* atau gangguan ke dalam algoritma (Yadav, 2020).

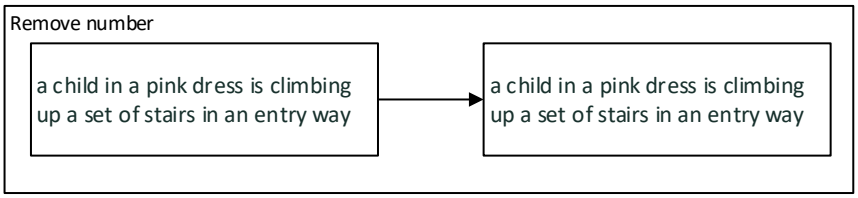


Gambar 3. Tahap Case Folding

Pada Gambar 3 terdapat tanda baca titik "." yang di akhir kalimat. Titik tersebut dihapus.

c. Remove Numeric

Remove Numeric adalah proses menghapus tulisan angka. Sama seperti tanda baca, angka yang terdapat pada teks juga tidak menambahkan banyak informasi pada *processing*. Jadi, angka bisa dihapus dari kalimat caption (Yadav, 2020).

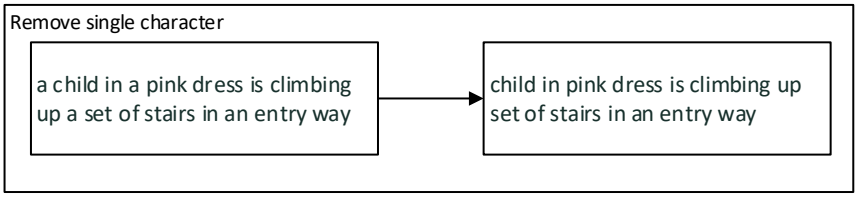


Gambar 4. Tahap Case Folding

Pada Gambar 4 tidak terdapat angka pada kalimat *caption*, jadi tidak ada perubahan yang dialami kalimat.

d. Remove Single Character

Remove Single Character adalah proses menghapus karakter tunggal contohnya seperti huruf "a" dan "s". Karakter tunggal ini dianggap tidak memiliki arti sehingga bisa dihapus (Malik, 2022).

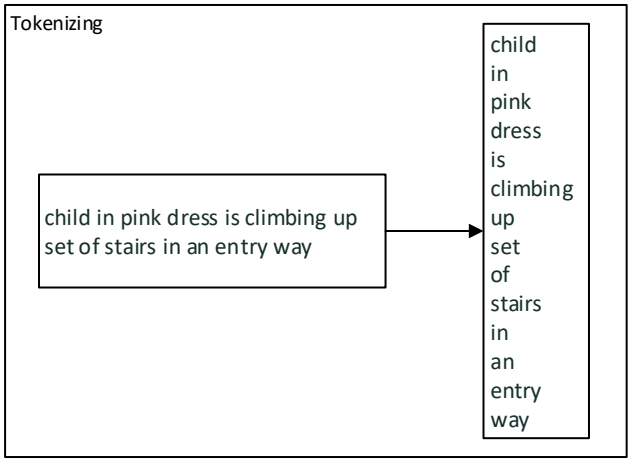


Gambar 5. Tahap Case Folding

Pada Gambar 5 terdapat karakter tunggal "a", sehingga output dari proses ini adalah "*child in pink dress is climbing up set of stairs in an entry way*".

e. Tokenizing

Tokenizing adalah proses memisahkan setiap kata dari kalimat.

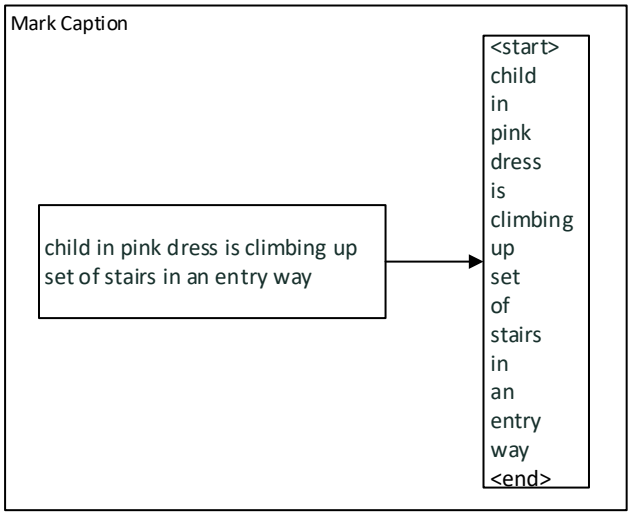


Gambar 6. Tahap Case Folding

Pada Gambar 6 ditunjukkan pemisahan setiap kata pada kalimat.

f. Mark Caption

Mark caption adalah proses menambahkan token baru.



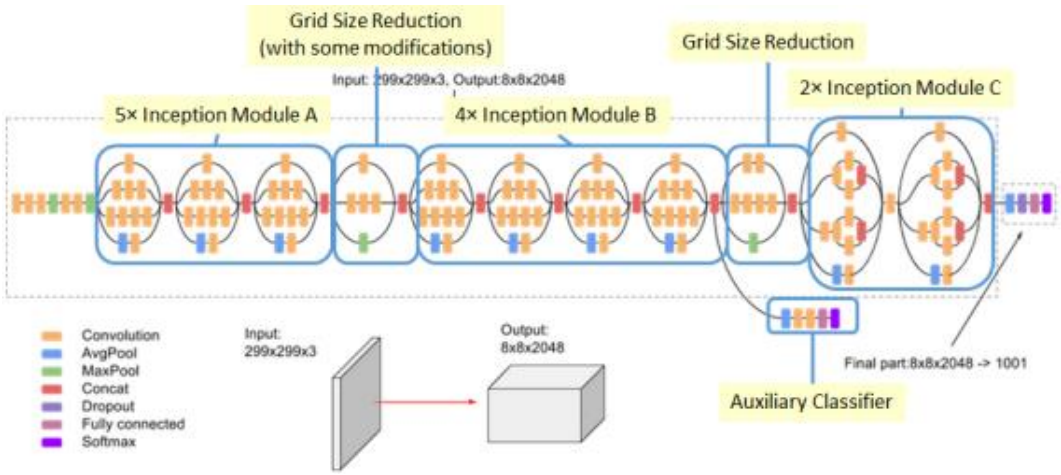
Gambar 7. Tahap Case Folding

Pada Gambar 7 ditunjukkan penambahan token "*<start>"* dan "*<end>".* Proses ini dilakukan agar sistem dapat mengenali bagian awal dan akhir kalimat *caption* dan

memutuskan untuk mulai menghasilkan kalimat dan berhenti untuk menghasilkan kalimat yang diprediksi.

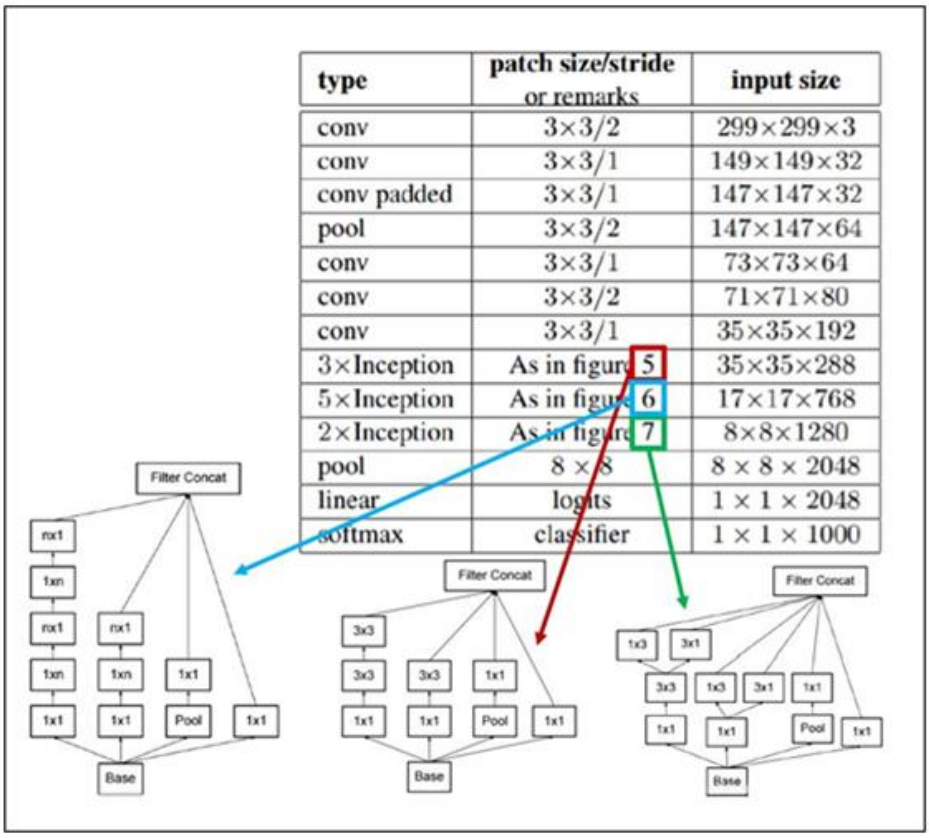
2.4 Inception-V3

Inception-V3 adalah sebuah arsitektur *deep convolutional* yang merupakan hasil dari pengembangan model GoogleNet atau Inception-v1 yang dikembangkan dari penelitian (**Szegedy et al., 2015**). Metode ini telah mengalami dua kali perubahan nama dan perkembangan pada arsitektur ini yaitu penambahan *batch normalization* (BN) **(Ioffe & Szegedy, 2015**) dan menambahkan faktorisasi tambahan pada tahap konvolusi untuk mengurangi jumlah koneksi atau parameter yang ada tanpa mengurangi jaringan yang digunakan dan dinamai Inception-V3 (**Szegedy et al., 2016**). Arsitektur dari Inception-V3 ditunjukkan pada Gambar 8.



Gambar 8. Arsitektur Inception-V3

Model Inception-V3 ini digunakan untuk melakukan *image feature extraction*. Ukuran *input* gambar yang dapat masuk ke dalam arsitektur ini berukuran 299 x 299 x 3 pixel. Untuk proses ekstraksi fitur, lapisan klasifikasi atau lapisan terakhir pada model dihapus karena penelitian ini tidak melakukan proses klasifikasi. Konfigurasi arsitektur Inception-v3 ditunjukkan pada Gambar 9.



Gambar 9. Konfigurasi Arsitektur Inception-V3

2.5 Transformer

Arsitektur Transformer pertama kali diperkenalkan oleh Ashish Vaswani pada tahun 2017 dalam penelitiannya yang berjudul "*Attention is All You Need*" oleh **(Vaswani et al., 2017**) dan Transformer menggunakan *self-attention mechanism* di dalam arsitekturnya. Transformer adalah sebuah model yang dapat memprediksi atau memulihkan kata kata secara berurutan dan juga dapat mengubah satu urutan berpindah ke urutan lainnya yang dibantu oleh *encoder-decoder*. Transformer ini menggunakan arsitektur *encoder-decoder* yang serupa dengan *Recurrent Neural Network* (RNN).

a. Key, Value, dan Query

Transformer memiliki bagian penting yang ada di dalamnya yaitu unit *multi-head self-attention mechanism*. Transformer menganggap *encoder* dari *input* sebagai satu yaitu pasangan *key* dan *value* (*K, V*) yang memiliki masing-masing dimensi *n* (panjang urutan *input*). Sedangkan pada *decoder*, *output* sebelumnya dikompres menjadi sebuah *query* (*Q*) dengan dimensi *m*. *Output* selanjutnya dihasilkan oleh pemetaan *query* tersebut dengan pasangan *key* dan *value* (**Vaswani et al., 2017**).

$$Attention(Q,K,V) = soft\,max(\ [\![QK]\!]\ \hat{}T/\sqrt{(d\_k\ )})V \qquad (1)$$

Keterangan :
Q = Matriks dari *query*
K = Matriks dari *key*
V = Matriks dari *value*
M = Optional mask
d_k = dimensi dari *key*

b. Multi-Head Self-Attention

Multi-head mechanism berjalan melalui *scaled dot-product attention* berulang kali sebanyak *h* secara paralel. *Output attention independent* akan digabungkan dan dipindahkan secara linier ke dalam dimensi yang ditentukan (**Weng, 2018**). *Multi-head self-attention* terdiri dari lapisan linier, *scaled dot-product attention*, *concat*, dan linier akhir.

$$MultiHead(Q,K,V) = Concat(head_1, \dots, head_h)W^O \quad (2)$$

$$Wherehead_i = Attention(QW_i^Q, KW_i^K, VW_i^V) \quad (3)$$

Di mana W_i^Q,W_i^K,W_i^V,dan W_i^O adalah parameter yang dipelajari.

c. Point-wise Feed Forward Network

Feed Forward Network (FFN) ditambahkan ke setiap sub-layer dari attention yang ada pada encoder dan decoder. Lapisan ini ditambahkan pada posisi yang terpisah tetapi memiliki lapisan yang identic antara satu sama lainnya. FFN memiliki dua transformasi linier dan menggunakan aktivasi ReLU didalamnya.

$$FFN(x) = max(0, xW_1 + b_1)W_2 + b_2 \quad (4)$$

d. Embedding dan Softmax

Embedding digunakan untuk mengubah token input dan token output menjadi vektor dimensi dmodel. Fungsi *softmax* digunakan untuk mengubah output decoder menjadi probabilitas token selanjutnya yang akan diprediksi (Vaswani et al., 2017).

e. Positional Encoding

Positional encoding digunakan untuk memberikan urutan pada token dan memberikan informasi mengenai posisi *relative* dan *absolute* dari token lainnya yang berada di dalam urutan, *Positional encoding* memiliki dimensi yang sama seperti *embedding* dimensi d_{model}. Untuk menentukan urutan token pada urutan, positional encoding memanfaatkan fungsi sinus dan cosinus dari frekuensi yang berbeda. Vektor dengan urutan genap akan menggunakan fungsi sinus, dan vektor urutan ganjil akan menggunakan fungsi cosinus (Vaswani et al., 2017)

$$PE_{(pos,2i)} = sin\left(\frac{pos}{10000^{2i/d_{model}}}\right) \quad (5)$$

$$PE_{(pos,2i+1)} = cos\left(\frac{pos}{10000^{2i/d_{model}}}\right) \quad (6)$$

f. Encoder

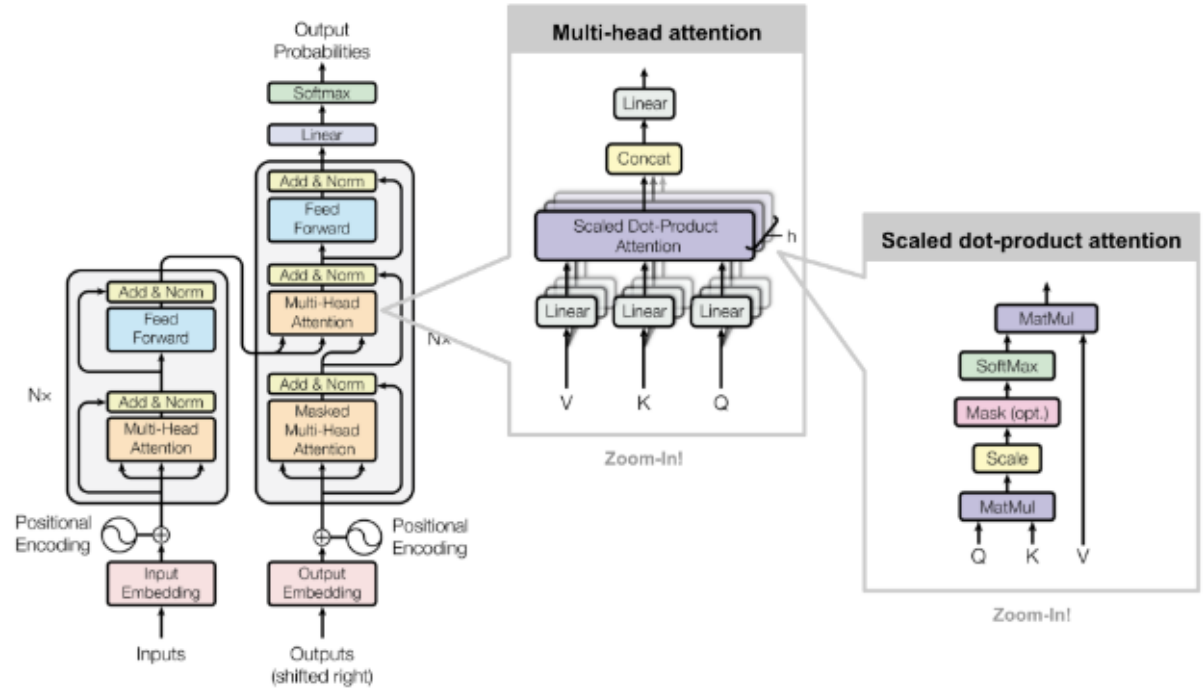
Encoder dapat menghasilkan sebuah representasi *attention-based* yang memiliki kemampuan untuk menempatkan bagian spesifik sebuah informasi dari konteks. *Encoder* terdiri dari beberapa bagian seperti memiliki 6 *layer encoder* yang identik, setiap *layer* memiliki memiliki *multi-head self-attention layer* dan *feed-forward layer*, dan setiap masing-masing *sub-layer* terdapat koneksi residual yang diikuti oleh *layer normalization*. Semua *output* data dari *sub-layer* memiliki dimensi yang sama yaitu dmodel = 512.

g. Decoder

Decoder dapat mengambil representasi yang telah dikodekan sebelumnya. *Decoder* terdiri dari beberapa bagian seperti memiliki 6 *layer encoder* yang identik, setiap *layer* memiliki memiliki *multi-head self-attention layer* dan *feed-forward layer*, setiap masing-masing sub-layer terdapat koneksi residual yang diikuti oleh *layer normalization* yang semua *output* data dari *sub-layer* memiliki dimensi yang sama yaitu dmodel = 512, dan *sub-layer masked multi-head self-attention* yang pertama dimodifikasi untuk mencegah posisi berpindah ke posisi berikutnya. Terjadi proses penggabungan *output* dengan *output* sebelumnya untuk menutupi *output* yang akan datang sehingga setiap *output* hanya akan memperhatikan *output-output* sebelumnya.

h. Transformer

Transformer memiliki tampilan arsitektur lengkap seperti urutan sumber dan urutan target pertama-tama memasuki bagian *embedding layer* untuk dapat menghasilkan data dengan dimensi yang sama yaitu d_{model} = 512, untuk mempertahankan informasi posisi dari input data maka diterapkan sebuah *sinusoid-wave-based positional encoding* dan dijumlahkan dengan *output embedding, dan softmax layer* dan *linear layer* ditambahkan ke *decoder output* akhir. Arsitektur lengkap dari Transformer ini ditunjukkan pada Gambar 10.



Gambar 10. Arsitektur Transformer.

2.6 BLEU (Bilingual Evaluation Understudy)

BLEU (*Bilingual Evaluation Understudy*) adalah sebuah algortima yang digunakan untuk mengevaluasi kualitas hasil dari mesin terjemahan (**Ardhi et al., 2018**) dan juga untuk model *image captioning* (**Al-faruq, 2021**). Terdapat sebuah ide yang mendasari BLEU ini yaitu "jika sebuah mesin dapat menerjemahkan sedekat terjemahan manusia, maka mesin itu akan semakin baik". Terdapat dua aspek yang mendasari dari ide tersebut, yaitu *adequacy* dan *fluency* (**Papineni et al., 2002**).

a. *Adequacy* adalah ukuran yang digunakan untuk mengetahui apakah arti dari target bahasa sudah diterjemahkan dari sumber bahasa. Terjemahan dengan kata-kata yang sama memenuhi kecukupan 1-gram atau unigram untuk *adequacy*.
b. *Fluency* mengukur bagaimana kalimat dapat dibentuk dengan benar secara tata bahasa dan mudah untuk ditafsirkan. Terjemahan yang lebih panjang memenuhi kecukupan N-gram untuk *fluecy*.

Tugas utama dari BLEU adalah untuk membandingkan *n-grams* dari kalimat kandidat dengan *n-grams* dari referensi terjemahan dan banyaknya jumlah *n*, dimana *n-grams* merupakan sekuensial kata yang muncul pada kalimat dan *n* merupakan ukuran teks. Semakin banyak jumlah *n*, maka semakin baik juga terjemahan kandidatnya. Pada Tabel 1 ditunjukkan contoh dari macam-macam *n-grams* seperti *unigram (1-gram)*, *bigram (2-gram)*, *trigram* (3-gram) dan *4-gram* dengan menggunakan kalimat "*A man wears an orange hat and glasses*".

Tabel 1. Contoh n-gram

1-gram	2-gram	3-gram	4-gram
A	A man	A man wears	A man wears an
Man	Man wears	Man wears an	Man wears an orange
Wears	Wears an	Wears an orange	Wears an orange hat
An	An Orange	An orange hat	An orange hat and
Orange	Orange Hat	Orange hat and	Orange hat and glasses
Hat	Hat and	Hat and glasses	
And	And Glasses		
Glasses			

Nilai yang dihasilkan dari BLEU ini berada diantara 0 sampai 1. Semakin tinggi nilai BLEU tersebut, maka semakin akurat model tersebut. Menurut (Lavie, 2010) skor BLEU diatas 0.3 dapat mencerminkan terjemahan yang dapat dimengerti dan skor BLEU diatas 0.5 dapat mencerminkan terjemahan yang baik dan fasih. Berikut merupakan rumus untuk menghitung BLEU:

$$BP_{BLEU} = \{1 \quad if\ c > r\ e^{(1-\frac{r}{c})} \quad if\ c \leq r \tag{4}$$

$$P_n = \frac{\sum_{C\in\ corpus\ n-gram\in C} \sum count_{clip(n-gram)}}{\sum_{C\in\ corpus\ n-gram\in C} \sum count_{(n-gram)}} \tag{5}$$

$$BLEU = BP_{BLEU}.e\ \sum_{n-1}^{N} w_n log P_n \tag{6}$$

Keterangan:

BP = *brevity penalty*

c = jumlah kata dari hasil terjemahan otomatis

r = jumlah kata rujukan

Pn = *modified precission score*

Wn = 1/N (standar nilai N untuk BLEU adalah 4)

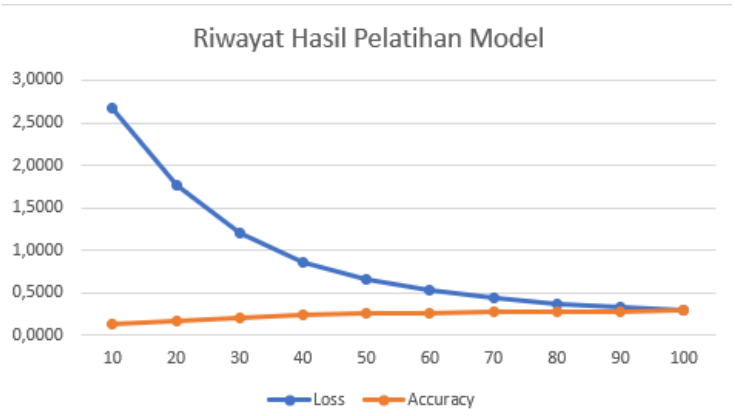
pn = jumlah n-gram hasil terjemahan yang sesuai dengan rujukan dibagi jumlah n-gram hasil terjemahan

3. HASIL DAN PEMBAHASAN

Berdasarkan pelatihan yang digunakan menggunakan model Inception-V3 dan Transformer terhadap data latih menghasilkan riwayat berupa *loss* atau nilai kesalahan dan *accuracy* atau nilai akurasi antara hasil prediksi dengan data asli dari model yang telah dibuat. Dari total pelatihan yang dilakukan menggunakan 100 kali epoch, dihasilkan riwayat pada *epoch* 1 *loss* = 5.8589 dan *accuracy* = 0.518. Kemudian nilai *loss* semakin menurun seiring bertambahnya *epoch*, dan nilai *accuracy* semakin menaik seiring bertambahnya *epoch*. Nilai *loss* terendah didapatkan pada *epoch* ke 100 dengan nilai *loss* = 0.2949, dan nilai *accuracy* tertinggi didapatkan pada *epoch* ke 100 dengan nilai *accuracy* = 0.2874. Adapun detail riwayat pelatihan model ditunjukkan pada Tabel 2 dengan ilustrasi grafik yang ditunjukkan pada Gambar 11.

Tabel 2. Riwayat Hasil Pelatihan Model

Epoch Ke-	***Loss***	***Accuracy***
10	2.6768	0.1253
20	1.7625	0.1675
30	1.2005	0.2050
40	0.8641	0.2324
50	0.6602	0.2510
60	0.5252	0.2640
70	0.4429	0.2720
80	0.3763	0.2789
90	0.3285	0.2839
100	0.2949	0.2874



Gambar 11. Grafik Riwayat Hasil Pelatihan Model

Gambar yang digunakan untuk mengevaluasi model adalah *data validation* sebanyak 1600 gambar yang sudah dipisahkan dari *data training*. Data gambar tidak melewati tahapan *model training*, sehingga dengan menggunakan gambar tersebut dapat mengetahui apakah model dapat belajar dengan baik atau tidak. Pengujian dilakukan terhadap 40 gambar. Setiap gambar yang diberikan akan menghasilkan kalimat yang didapatkan dari model sebagai kalimat referensi untuk menghitung skor BLEU dengan menggunakan *cumulative 4-gram* yang terdiri dari skor BLEU-1, BLEU-2, BLEU-3 dan BLEU-4. Pada tabel 3 ditunjukkan kinerja *n-gram* yang dilakukan pada salah satu kalimat hasil diprediksi terhadap pengujian gambar.

Tabel 3. Tabel Kinerja *n-gram*

1-gram	2-gram	3-gram	4-gram
dog	dog runs	dog runs toward	dog runs toward the
runs	runs toward	runs toward the	runs toward the camera
toward	toward the	toward the camera	toward the camera with
the	the camera	the camera with	the camera with stick
camera	camera with	camera with stick	camera with stick in
with	with stick	with stick in	with stick in its
stick	stick in	stick in its	stick in its mouth
in	in its	in its mouth	
its	its mouth		
mouth			

Setiap *n-gram* yang dihasilkan lalu dihitung skor *precision* dengan formula (5), lalu dihitung skor BLEU untuk masing-masing *n-gram* dengan formula (6) yang diimplementasikan pada program yang dibuat. Pada Tabel 4 ditunjukkan hasil skor BLEU-1, BLEU-2, BLEU-3, dan BLEU-4 serta prediksi kalimat yang dihasilkan oleh sistem.

Tabel 4. Hasil Skor BLEU dan Prediksi Kalimat

Gambar	Skor BLEU	*Caption* Asli	*Caption* Prediksi
	BLEU-1 score: 0.6 BLEU-2 score: 0.4472135954999577 BLEU-3 score: 0.4070905315369044 BLEU-4 score: 0.2907153684841096	the dog carrying long stick in its mouth	dog runs toward the camera with stick in its mouth

Berdasarkan hasil *caption* yang diprediksi, hasil *caption* memiliki makna yang sama dengan *caption* asli dan dapat mendeskripsikan gambar serta konteks dari gambar yang dihasilkan. Tetapi masih terdapat beberapa kata yang berbeda dari hasil *caption* yang diprediksi dengan *caption* asli.

4. KESIMPULAN

Pada penelitian ini telah diimplementasikan model Inception-V3 dan Transformer dalam melakukan *image captioning* sebagai *image feature extraction* dan *caption generation*. Model yang diusulkan mampu untuk mengekstraksi fitur gambar dan mengubahnya menjadi kalimat untuk mendeskripsikan gambar.

Model yang telah dibangun lalu diukur kinerjanya dengan menggunakan BLEU *scoring*. Terdapat empat parameter skor yaitu BLEU-1, BLEU-2, BLEU-3, dan BLEU-4. Berdasarkan model tersebut, nilai rata-rata skor BLEU-1, BLEU-2, BLEU-3, dan BLEU-4 yang didapatkan adalah (0.306, 0.184, 0.123, 0.084). Dengan skor BLEU tersebut dapat disimpulkan bahwa model yang dibangun masih belum cukup baik untuk dapat menghasilkan kalimat *caption* dari suatu gambar.

UCAPAN TERIMA KASIH

Penulis mengucapkan terima kasih kepada Program Studi Informatika Institut Teknologi Nasional (ITENAS) Bandung. Penulis juga berterima kasih kepada semua pihak yang telah membantu dan mendukung sehingga penelitian ini dapat berjalan dengan baik.

DAFTAR PUSTAKA

Al-faruq, U. A. A. (2021). Implementasi Arsitektur Transformer pada Image Captioning dengan Bahasa Indonesia. *Automata*, *2*(2).

Ardhi, H., Sujaini, H., & Putra, A. B. (2018). Analisis Penggabungan Korpus dari Hadits Nabi dan Alquran untuk Mesin Penerjemah Statistik. *Jurnal Linguistik Komputasional*, *1*(1), 31–37. https://doi.org/10.26418/jlk.v1i1.1

Devlin, J., Cheng, H., Fang, H., Gupta, S., Deng, L., He, X., Zweig, G., & Mitchell, M. (2015). Language models for image captioning: The quirks and what works. *ACL-IJCNLP 2015 - 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing of the Asian Federation of Natural Language Processing, Proceedings of the Conference*, *2*, 100–105. https://doi.org/10.3115/v1/p15-2017

Ioffe, S., & Szegedy, C. (2015). Batch normalization: Accelerating deep network training by reducing internal covariate shift. *32nd International Conference on Machine Learning, ICML 2015*, *1*, 448–456.

Kadhim, A. I. (2018). An Evaluation of Preprocessing Techniques for Text Classification. *International Journal of Computer Science and Information Security*, *16*(6), 22–32. https://sites.google.com/site/ijcsis/

Lavie, A. (2010). Evaluating the output of machine translation systems. *AMTA 2010 - 9th Conference of the Association for Machine Translation in the Americas*.

Liu, S., Bai, L., Hu, Y., & Wang, H. (2018). Image Captioning Based on Deep Neural Networks. *MATEC Web of Conferences*, *232*, 1–7. https://doi.org/10.1051/matecconf/201823201052

Malik, U. (2022). *Text Classification with Python and Scikit-Learn*. StackAbuse. https://stackabuse.com/text-classification-with-python-and-scikit-learn/

Nursikuwagus, A., Munir, R., & Khodra, M. L. (2020). Image Captioning menurut Scientific Revolution Kuhn dan Popper. *Jurnal Manajemen Informatika (JAMIKA)*, *10*(2), 110–121. https://doi.org/10.34010/jamika.v10i2.2630

Papineni, K., Roukos, S., Ward, T., & Zhu, W.-Z. (2002). BLEU: a Method for Automatic

Evaluation of Machine Translation. *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics (ACL)*, 311–318. https://doi.org/10.1002/andp.19223712302

Srinivasan, L., Sreekanthan, D., & Amutha, A. L. (2018). *Image Captioning - A Deep Learning Approach*. *13*(9), 7239–7242.

Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., Erhan, D., Vanhoucke, V., & Rabinovich, A. (2015). Going Deeper with Convolutions. *Research Methods in Applied Settings*. https://doi.org/10.4324/9781410605337-29

Szegedy, C., Vanhoucke, V., Ioffe, S., Shlens, J., & Wojna, Z. (2016). Rethinking the Inception Architecture for Computer Vision. *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, *2016-Decem*, 2818–2826. https://doi.org/10.1109/CVPR.2016.308

Tammina, S. (2019). Transfer learning using VGG-16 with Deep Convolutional Neural Network for Classifying Images. *International Journal of Scientific and Research Publications (IJSRP)*, *9*(10).

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., & Polosukhin, I. (2017). Attention is all you need. *Advances in Neural Information Processing Systems*, *2017-Decem*(Nips), 5999–6009.

Vinyals, O., Toshev, A., Bengio, S., & Erhan, D. (2015). Show and tell: A neural image caption generator. *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, *07-12-June*, 3156–3164. https://doi.org/10.1109/CVPR.2015.7298935

Weng, L. (2018). *Attention? Attention!* https://lilianweng.github.io/posts/2018-06-24-attention/

Yadav, D. (2020). *NLP: Building Text Cleanup and PreProcessing Pipeline*. Towards Data Science. https://towardsdatascience.com/nlp-building-text-cleanup-and-preprocessing-pipeline-eba4095245a0