JNATIA Volume 3, Nomor 2, Februari 2025 p-ISSN: 2986-3929
Jurnal Nasional Teknologi Informasi dan Aplikasinya e-ISSN: 3032-1948

Implementasi Ekstraksi Fitur VGG-16 dan Pemodelan LSTM untuk Pembangkitan Caption Gambar Otomatis

Made Pranajaya Dibyacita[a1], Luh Gede Astuti[a2]

[a]Program Studi Informatika, Fakultas Matematika dan Ilmu Pengetahuan Alam, Universitas Udayana
Jalan Raya Kampus Udayana, Bukit Jimbaran, Kuta Selatan, Badung, Bali Indonesia
[1]mdpranajaya@gmail.com
[2]lg.astuti@unud.ac.id

Abstract

Image captioning, the task of automatically generating descriptive captions for images, has gained significant attention due to its potential applications in various domains. This paper addresses the challenges associated with integrating computer vision and natural language processing techniques to develop an effective image caption generator. The proposed solution leverages the VGG-16 model for feature extraction from images and an LSTM (Long Short-Term Memory) model for caption generation. The Flickr8k dataset, containing approximately 8000 images with five different captions per image, is utilized for training and evaluation. The methodology encompasses several steps, including data preprocessing, feature extraction, model training, and evaluation. Data preprocessing involves cleaning captions by removing punctuations, single characters, and numerical values, while incorporating start and end sequences. Image features are extracted using the pre-trained VGG-16 model, and similar images are clustered to ensure accurate feature extraction. Subsequently, the captions and corresponding image features are merged and tokenized for model training. The LSTM model is designed with input layers for image features and captions, as well as an output layer for caption generation. Extensive hyperparameter tuning is conducted to optimize the model's performance, involving variations in the number of nodes and layers. The generated captions are evaluated using BLEU scores, where a score closer to 1 indicates higher similarity between predicted and actual captions. The proposed system demonstrates promising results in generating meaningful captions for images, with potential applications in assisting visually impaired individuals, medical image analysis, and advertising industry automation.

Keywords: *Image Captioning, Deep Learning, VGG-16, LSTM, NLP, BLEU.*

1. Pendahuluan

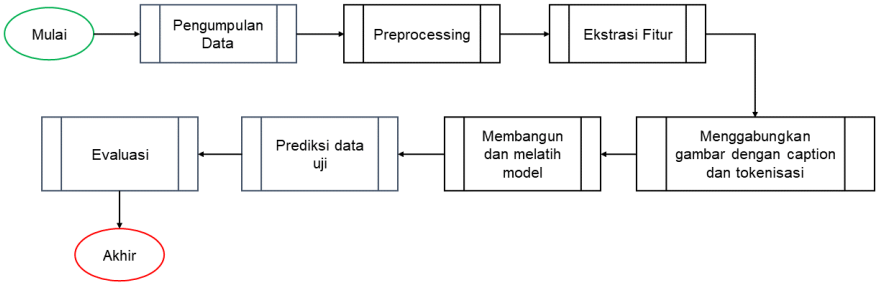
Dalam era digital yang semakin maju, gambar dan foto telah menjadi bagian integral dari kehidupan kita sehari-hari. Tak hanya sebagai media pengalaman visual semata, gambar juga menjadi alat komunikasi yang penting, yang membawa tantangan baru dalam hal pengelolaan dan pemahaman konten visual. Mengatasi tantangan ini memerlukan pendekatan yang inovatif, salah satunya adalah dengan menggabungkan teknologi *computer vision* dan pemrosesan bahasa alami. Dengan demikian, muncul bidang *Image Caption Generation*, yang menggabungkan kedua teknologi tersebut untuk menghasilkan deskripsi otomatis yang relevan dan bermakna untuk gambar. Bidang *Image Caption Generation* merupakan hasil dari perpaduan antara *computer vision* dan pemrosesan bahasa alami. Tujuannya adalah memanfaatkan model *deep learning* untuk secara otomatis menghasilkan deskripsi bahasa yang tepat dan koheren terhadap informasi visual. Keberadaan teknologi ini memiliki makna yang sangat penting, terutama dalam memberikan bantuan bagi orang awam terhadap dunia medis untuk menganalisis gambar hasil rontgen, dan membantu orang dengan gangguan penglihatan, memungkinkan mereka memahami konten gambar melalui deskripsi teks [1]. *Computer vision, machine translation, dan object detection* adalah domain yang dinamis dalam dunia *machine learning*, mengalami pertumbuhan yang signifikan dalam dekade terakhir [2]. Pertumbuhan ini telah menghasilkan berbagai kerangka kerja (framework) yang memfasilitasi implementasi

p-ISSN: 2986-3929
e-ISSN: 3032-1948

caption generation. Metode tradisional sebagian besar mengandalkan *feature extractor* yang dirancang secara manual dan model bahasa berbasis aturan, tetapi efeknya terbatas [1]. Dalam konteks ini, penggunaan model VGG-16 dan LSTM *(Long Short-Term Memory)* telah menjadi pilihan yang populer dalam mengatasi tantangan *Image Caption Generation*. Model VGG-16 digunakan untuk ekstraksi fitur dari gambar, sementara LSTM digunakan untuk menghasilkan deskripsi bahasa yang relevan. Pendekatan ini, di mana model VGG-16 digunakan untuk ekstraksi fitur dari gambar dan LSTM untuk menghasilkan deskripsi bahasa yang relevan, telah membuka peluang baru dalam menangani kompleksitas visual dan linguistik. Tiga langkah utama dalam menciptakan model *image captioning* melibatkan ekstraksi fitur dari gambar dan caption untuk mendukung model, menggunakan fitur-fitur ini untuk melatih model, dan menggunakan model yang telah dilatih untuk menghasilkan caption berdasarkan atribut gambar input. Hasil dari penggunaan teknologi ini dievaluasi menggunakan skor *Bilingual Evaluation Understudy* (BLEU) sebagai tolak ukur untuk membandingkan dan menilai efektivitas model. Penelitian menunjukkan peningkatan yang signifikan dalam akurasi skor BLEU dibandingkan dengan pendekatan dasar, menegaskan keberhasilan dan relevansi pendekatan yang diusulkan. Selain itu, model yang dikembangkan juga memiliki kemampuan unik untuk menciptakan caption gambar secara otomatis, yang membedakannya dari metode dasar yang hanya meminjam caption yang sudah ada. Dengan demikian, teknologi *Image Caption Generation* terbukti menjadi alat yang berharga dalam memahami dan mengelola konten visual di era digital saat ini.

2. Metode Penelitian

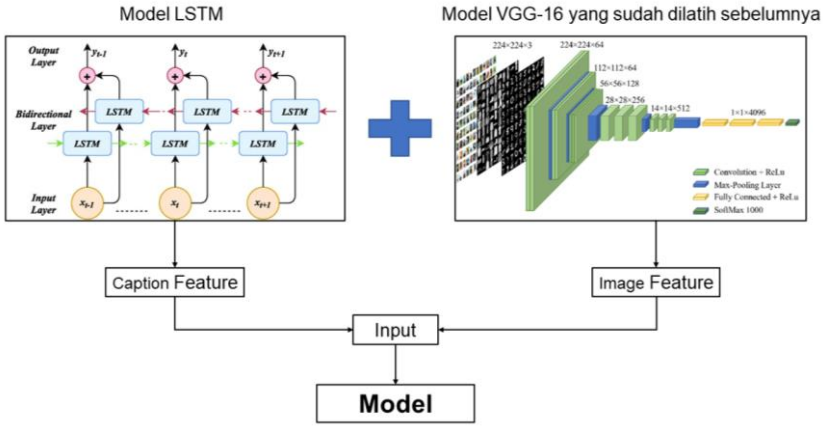
Penelitian ini bertujuan untuk mengeksplorasi integrasi antara ekstraksi fitur menggunakan model VGG-16 dan pemodelan LSTM dalam konteks pembangkitan caption gambar otomatis, dengan tools yang digunakan yaitu *Jupyter Notebook* dengan bahasa python. Langkah-langkah yang diterapkan dalam penelitian ini meliputi pengumpulan data, preprocessing, ekstraksi fitur menggunakan VGG-16, menggabungkan gambar dengan caption dan tokenisasi, membangun dan melakukan pelatihan pada model LSTM, prediksi data uji, dan evaluasi yang akan diuraikan sebagai berikut:



Gambar 1. Diagram Alur

Dalam pendekatan tersebut, penelitian ini juga melibatkan pelatihan model menggunakan dataset gambar Flickr8K [3]. Untuk prediksi, model ini menggunakan arsitektur jaringan saraf *Long Short-Term Memory* (LSTM), bersama dengan kombinasi caption dataset Flickr8K dan atribut gambar yang diekstraksi menggunakan VGG. Aspek inti dari proses prediksi gambar melibatkan ekstraksi fitur VGG dan penggunaan model LSTM yang telah dilatih untuk pembangkitan caption seperti yang dapat dilihat pada Gambar 2.

p-ISSN: 2986-3929
e-ISSN: 3032-1948



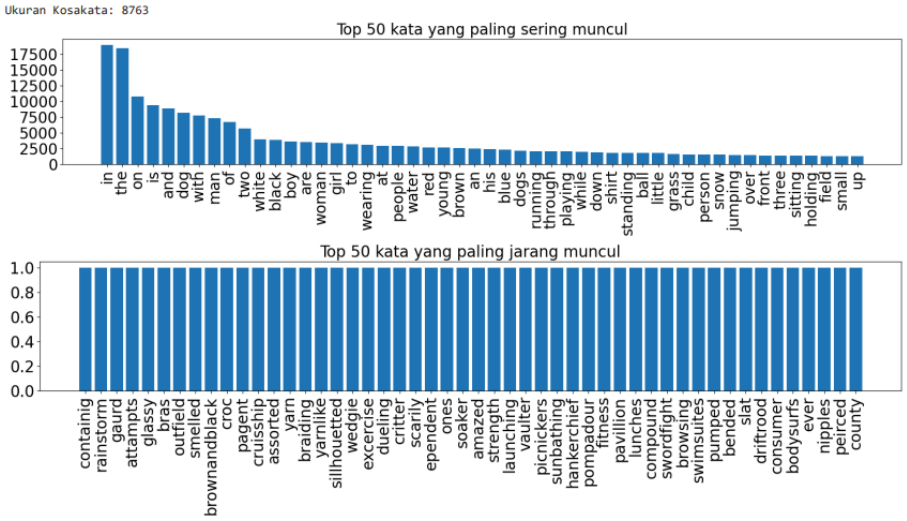
Gambar 2. Kerangka Model

2.1 Pengumpulan Data

Dataset yang digunakan adalah Flickr8k, terdiri dari 8.092 gambar dengan hingga lima keterangan deskriptif [4]. Data ini berasal dari layanan berbagi foto Flickr dengan lisensi yang sesuai. Setiap gambar dilengkapi dengan caption dalam bahasa Inggris, mendukung penelitian dalam pemrosesan bahasa alami dan visi komputer, termasuk *image captioning*, pengenalan objek, dan pemahaman konten gambar. Dataset ini digunakan sebagai sumber daya penting bagi pemula dalam machine learning dan AI, memberikan titik acuan dasar untuk proyek deskripsi gambar berbasis kalimat [5]. Selain itu, model juga menggunakan data dari VGG-16 yang telah dilatih sebelumnya untuk ekstraksi fitur gambar yang dapat diakses disini. Untuk pembagian data untuk latih, uji, dan validasi adalah 60%, 20%, dan 20% dari total dataset secara berturut-turut.

2.2 Tahap Preprocessing Data

Langkah preprocessing data melibatkan pembersihan caption untuk menghilangkan ekspresi reguler, angka, dan kata-kata tidak relevan *(stop words)*. Ini termasuk penghapusan tanda baca, karakter tunggal, dan nilai numerik. Setelah pembersihan, dilakukan analisis untuk menemukan kata-kata paling umum dan paling jarang dalam dataset. Data caption yang telah dibersihkan diperbarui dalam DataFrame dengan nama file gambar yang terkait. Selanjutnya, peneliti melakukan perhitungan frekuensi kemunculan kata-kata dan memvisualisasi 50 kata teratas dan 50 kata terendah.

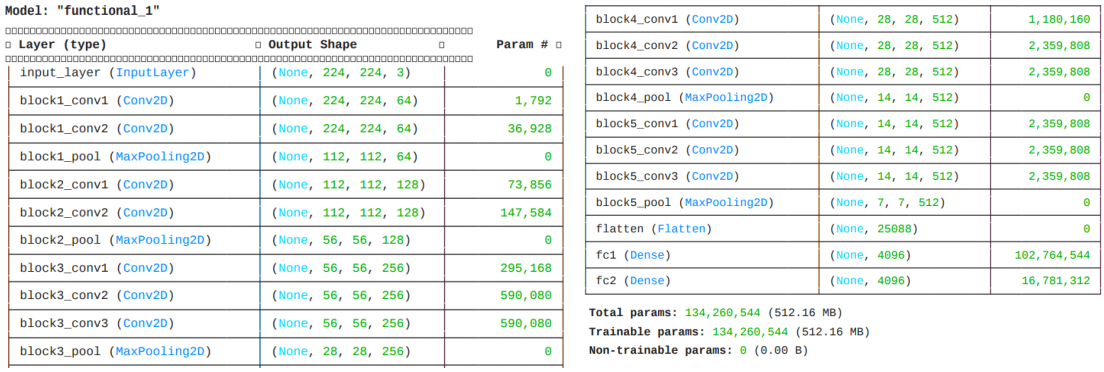


Gambar 3. Top 50 kata paling sering dan paling jarang muncul

p-ISSN: 2986-3929
e-ISSN: 3032-1948

2.3 Tahap Ekstraksi Fitur

Pada penelitian, dilakukan ekstraksi fitur menggunakan VGG-16 yang telah dilatih sebelumnya pada dataset besar yang berisi gambar-gambar bervariasi. Dalam konteks pembangkitan caption gambar, model ini digunakan untuk mengekstraksi fitur-fitur penting dari gambar yang akan digunakan sebagai input untuk model pembangkit caption [6]. Dalam penelitian ini, langkah pertama yang dilakukan adalah memuat model beserta bobot yang telah dilatih sebelumnya. Setelah itu, lapisan terakhir dari model yang digunakan untuk klasifikasi akan dihapus, sehingga model hanya akan menghasilkan vektor fitur dari gambar yang diberikan sebagai input. Dalam proses ini, VGG-16 akan menghasilkan vektor fitur dengan dimensi 4096 untuk setiap gambar yang diproses.



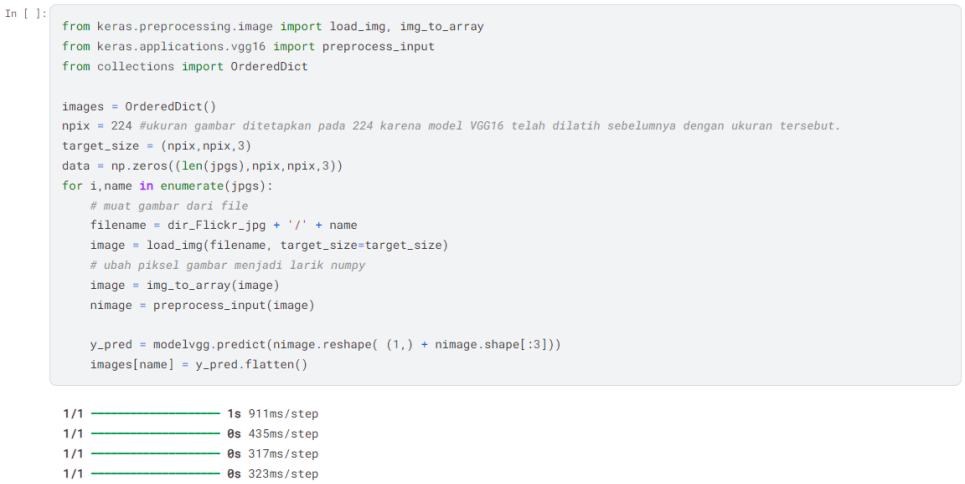
Model: "functional_1"

Layer (type)	Output Shape	Param #
input_layer (InputLayer)	(None, 224, 224, 3)	0
block1_conv1 (Conv2D)	(None, 224, 224, 64)	1,792
block1_conv2 (Conv2D)	(None, 224, 224, 64)	36,928
block1_pool (MaxPooling2D)	(None, 112, 112, 64)	0
block2_conv1 (Conv2D)	(None, 112, 112, 128)	73,856
block2_conv2 (Conv2D)	(None, 112, 112, 128)	147,584
block2_pool (MaxPooling2D)	(None, 56, 56, 128)	0
block3_conv1 (Conv2D)	(None, 56, 56, 256)	295,168
block3_conv2 (Conv2D)	(None, 56, 56, 256)	590,080
block3_conv3 (Conv2D)	(None, 56, 56, 256)	590,080
block3_pool (MaxPooling2D)	(None, 28, 28, 256)	0
block4_conv1 (Conv2D)	(None, 28, 28, 512)	1,180,160
block4_conv2 (Conv2D)	(None, 28, 28, 512)	2,359,808
block4_conv3 (Conv2D)	(None, 28, 28, 512)	2,359,808
block4_pool (MaxPooling2D)	(None, 14, 14, 512)	0
block5_conv1 (Conv2D)	(None, 14, 14, 512)	2,359,808
block5_conv2 (Conv2D)	(None, 14, 14, 512)	2,359,808
block5_conv3 (Conv2D)	(None, 14, 14, 512)	2,359,808
block5_pool (MaxPooling2D)	(None, 7, 7, 512)	0
flatten (Flatten)	(None, 25088)	0
fc1 (Dense)	(None, 4096)	102,764,544
fc2 (Dense)	(None, 4096)	16,781,312

Total params: 134,260,544 (512.16 MB)
Trainable params: 134,260,544 (512.16 MB)
Non-trainable params: 0 (0.00 B)

Gambar 4. Menghapus Lapisan Terakhir Model

Sebelum gambar dapat diproses oleh VGG-16, gambar harus diubah ke dalam format yang sesuai dengan input yang diharapkan oleh model. Dalam kasus ini, gambar harus diubah ukurannya menjadi 224x224 piksel dan dinormalisasi agar nilai pikselnya berada dalam rentang 0 hingga 1. Proses ini dilakukan dengan menggunakan fungsi-fungsi dari library Keras seperti *load_img, img_to_array,* dan *preprocess_input.*



In [ ]:

```
from keras.preprocessing.image import load_img, img_to_array
from keras.applications.vgg16 import preprocess_input
from collections import OrderedDict

images = OrderedDict()
npix = 224 #ukuran gambar ditetapkan pada 224 karena model VGG16 telah dilatih sebelumnya dengan ukuran tersebut.
target_size = (npix,npix,3)
data = np.zeros((len(jpgs),npix,npix,3))
for i,name in enumerate(jpgs):
    # muat gambar dari file
    filename = dir_Flickr_jpg + '/' + name
    image = load_img(filename, target_size=target_size)
    # ubah piksel gambar menjadi larik numpy
    image = img_to_array(image)
    nimage = preprocess_input(image)

    y_pred = modelvgg.predict(nimage.reshape( (1,) + nimage.shape[:3]))
    images[name] = y_pred.flatten()
```

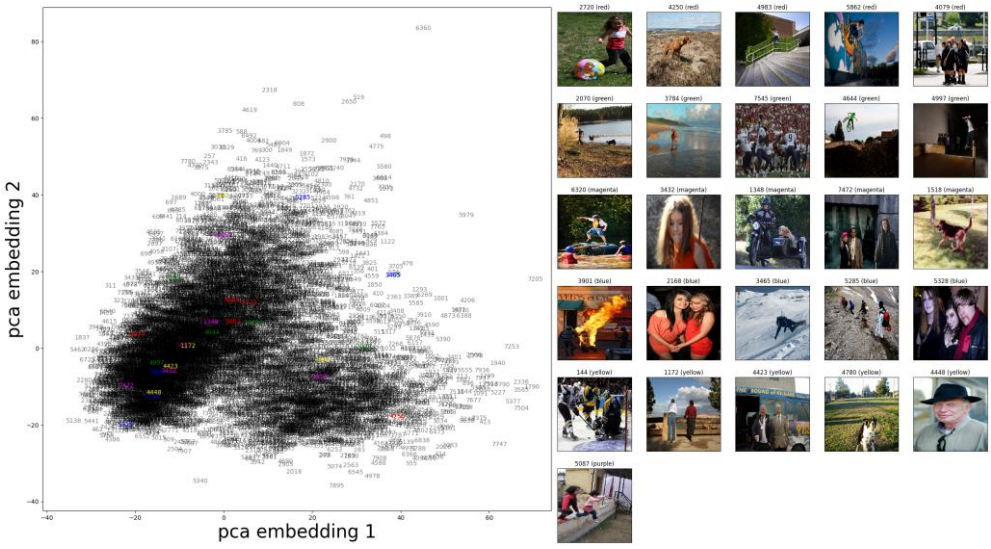
```
1/1 ━━━━━━━━━━━━━━━━━━━━ 1s 911ms/step
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 435ms/step
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 317ms/step
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 323ms/step
```

Gambar 5. Ekstraksi Fitur

Setelah fitur diekstraksi menggunakan model VGG-16, dilakukan visualisasi untuk mengelompokkan gambar dengan karakteristik visual serupa menggunakan teknik *Principal Component Analysis* (PCA). PCA digunakan untuk mengurangi dimensi fitur dari 4096 menjadi 2, memungkinkan visualisasi dalam ruang dua dimensi. Cluster gambar dibentuk berdasarkan kesamaan fitur visual, dan beberapa contoh gambar diambil dari setiap kelompok untuk ditampilkan dalam plot. Proses ini penting untuk memverifikasi keberhasilan ekstraksi fitur dan

p-ISSN: 2986-3929
e-ISSN: 3032-1948

memahami pola dalam dataset gambar.



Gambar 6. Memplot Gambar-gambar yang Mirip dari Dataset

2.4 Tahap Menggabungkan Gambar dengan Caption dan Tokenisasi

Penelitian ini, melakukan penggabungan gambar yang dimana proses ini dimulai dengan pemilihan caption pertama dari setiap gambar dalam dataset. Dalam konteks ini, pemilihan hanya dilakukan terhadap caption pertama untuk menghindari kompleksitas yang mungkin timbul dari penggunaan seluruh caption. Selanjutnya, caption dan gambar yang terkait digabungkan bersama untuk membentuk pasangan data yang akan digunakan dalam pelatihan.

Out[34]:

	namafile	index	caption
0	1000268201_693b08cb0e.jpg	0	startseq child in pink dress is climbing up s...
5	1001773457_577c3a7d70.jpg	0	startseq black dog and spotted dog are fighti...
10	1002674143_1b742ab4b8.jpg	0	startseq little girl covered in paint sits in...
15	1003163366_44323f5815.jpg	0	startseq man lays on bench while his dog sits...
20	1007129816_e794419615.jpg	0	startseq man in an orange hat starring at som...

Gambar 7. Menggabungkan Gambar dengan Caption

Setelah proses penggabungan, selanjutnya dilakukan tokenisasi caption. Tokenisasi ini diperlukan karena model yang akan digunakan tidak dapat menerima teks mentah sebagai input, melainkan memerlukan representasi vektor kata-kata. Tokenisasi dilakukan menggunakan Tokenizer dari TensorFlow, di mana setiap kata dalam caption dikonversi menjadi urutan angka yang merepresentasikan kata-kata dalam kamus. Pada tahap ini, juga ditetapkan jumlah kata maksimum dalam kamus, yang dalam kasus ini disetel sebesar 6000 kata.

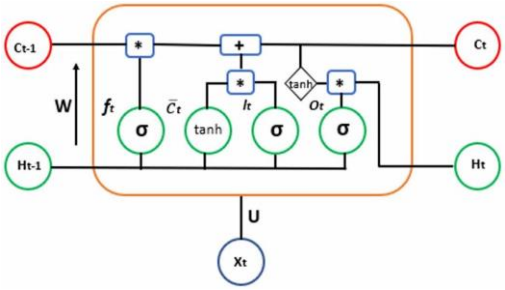
```
ukuran kosakata : 4476
[[1, 38, 3, 66, 144, 7, 124, 52, 406, 9, 367, 3, 24, 2351, 522, 2], [1, 12, 8, 5, 752,
8, 17, 368, 2], [1, 48, 15, 170, 3, 584, 101, 3, 41, 9, 551, 1198, 11, 55, 213, 3, 1076,
2], [1, 10, 621, 6, 150, 27, 23, 8, 101, 46, 112, 2], [1, 10, 3, 24, 82, 96, 1199, 19, 1
62, 2]]
```

Gambar 8. Tokenisasi Caption

p-ISSN: 2986-3929
e-ISSN: 3032-1948

2.5 Tahap Membangun dan Melatih Model

Penelitian ini, akan membangun model LSTM yang merupakan tahap krusial dalam proses pembangkitan caption untuk gambar. Model LSTM dipilih karena mampu memperhitungkan keadaan keluaran sel sebelumnya dan masukan sel saat ini untuk menghasilkan keluaran saat ini [7]. Hal ini sangat berguna saat menghasilkan caption untuk gambar-gambar.



Gambar 9. LSTM *Network*

$*$ = *Elementwise multiplicant* (perkalian elemen demi elemen)
$+$ = *Element-wise addition* (penambahan elemen demi elemen)

$$f_t = \sigma\ (X_t\ *\ U_f +\ H_{t-1} * W_f)$$

$$\underline{C_t} = tanh\ (X_t\ *\ U_c +\ H_{t-1} * W_c)$$

$$I_t\ = \sigma\ (X_t\ *\ U_i +\ H_{t-1} * W_i)$$

$$O_t = \sigma\ (X_t\ *\ U_o +\ H_{t-1} * W_o)$$

$$C_t = f_t * C_{t-1} + I_t * \underline{C_t}$$

$$H_t = O_t * tanh\ (C_t)$$

Keterangan:

X_t = Input Vector
H_{t-1} = Sel Output Sebelumnya
C_{t-1} = Memori Output Sebelumnya
H_t = Sel Output Saat Ini
C_t = Sel Memori Saat Ini
W, U = Weight vector untuk gerbang forget (f), candidate (c), gerbang input (i), gerbang output (o)

Pada model LSTM ini, peneliti akan membangun lapisan input dan output untuk menghasilkan caption dengan variasi jumlah simpul dan lapisan, mulai dari 256 hingga 1024, dengan penyetelan *hyperparameter* yang cermat. Proses ini melibatkan penggunaan fitur gambar dan teks yang telah di-tokenisasi sebelumnya, dengan teks melewati lapisan *embedding* dan LSTM untuk mengekstraksi pola urutan. Dua lapisan LSTM digunakan dengan *dropout* untuk mencegah *overfitting*, diikuti dengan penggabungan hasilnya menggunakan operasi penambahan. Output dari model adalah distribusi probabilitas kata-kata dalam kamus, dikompilasi dengan fungsi *loss categorical_crossentropy* dan *optimizer adam*.

p-ISSN: 2986-3929
e-ISSN: 3032-1948

4476
Model: "functional_9"

Layer (type)	Output Shape	Param #	Connected to
input_layer_8 (InputLayer)	(None, 30)	0	-
embedding_3 (Embedding)	(None, 30, 64)	286,464	input_layer_8[0]…
not_equal_3 (NotEqual)	(None, 30)	0	input_layer_8[0]…
FiturCaption (LSTM)	(None, 30, 256)	328,704	embedding_3[0][0… not_equal_3[0][0]
dropout_3 (Dropout)	(None, 30, 256)	0	FiturCaption[0][…
input_layer_7 (InputLayer)	(None, 4096)	0	-
FiturCaption2 (LSTM)	(None, 256)	525,312	dropout_3[0][0], not_equal_3[0][0]
FiturGambar (Dense)	(None, 256)	1,048,832	input_layer_7[0]…
add_3 (Add)	(None, 256)	0	FiturCaption2[0]… FiturGambar[0][0]
dense_6 (Dense)	(None, 256)	65,792	add_3[0][0]
dense_7 (Dense)	(None, 4476)	1,150,332	dense_6[0][0]

Total params: 3,405,436 (12.99 MB)
Trainable params: 3,405,436 (12.99 MB)
Non-trainable params: 0 (0.00 B)

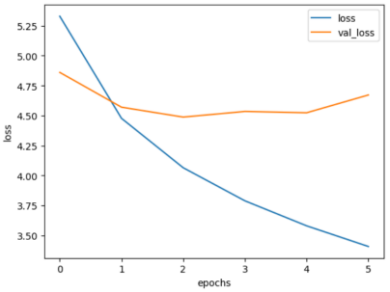
Gambar 10. Membangun Model LSTM

Setelah model dibangun, pelatihan dilakukan menggunakan metode *fit ()* dari Keras dengan data latih dan validasi. Selama pelatihan, evaluasi kerugian *(loss)* dipantau untuk mengukur kinerja model. Proses pelatihan dilakukan hingga model mencapai kinerja yang diharapkan atau tidak lagi meningkat secara signifikan. Grafik *loss* dan *val_loss* digunakan untuk memvisualisasikan kinerja model selama pelatihan, menunjukkan perubahan *loss* pada data latih dan validasi seiring dengan jumlah *epochs*. Perubahan ini membantu memahami apakah model cenderung *underfitting* atau *overfitting*.

```
Melatih pada 49631 sampel, dan memvalidasi pada 16353 sampel.

Epoch 1/6
1551/1551 - 308s - 199ms/step - loss: 5.3285 - val_loss: 4.8601
Epoch 2/6
1551/1551 - 307s - 198ms/step - loss: 4.4770 - val_loss: 4.5699
Epoch 3/6
1551/1551 - 307s - 198ms/step - loss: 4.0648 - val_loss: 4.4874
Epoch 4/6
1551/1551 - 290s - 187ms/step - loss: 3.7888 - val_loss: 4.5345
Epoch 5/6
1551/1551 - 290s - 187ms/step - loss: 3.5803 - val_loss: 4.5234
Epoch 6/6
1551/1551 - 322s - 207ms/step - loss: 3.4080 - val_loss: 4.6717

 Waktu yang dibutuhkan: 30.48 Menit.
```



Gambar 11. Melatih Model LSTM dan Menampilkan Grafik Loss dan Val_loss

2.6 Tahap Prediksi Data Uji

Penelitian ini, akan menguji pada dataset uji untuk melihat bagaimana kinerjanya dalam menghasilkan caption untuk beberapa gambar. Jika caption yang dihasilkan sudah dapat diterima, langkah selanjutnya adalah menghasilkan caption untuk seluruh dataset uji. Proses ini melibatkan perbandingan antara caption yang dihasilkan oleh model dengan caption sebenarnya dari dataset.

Tabel 1. Hasil Prediksi Data Uji

Gambar	Prediksi
	startseq man in blue shirt is standing on the street endseq
	startseq black and white dog is running in the grass endseq

Gambar	Prediksi
	startseq black and white dog is running through the snow endseq
	startseq man in blue shirt is riding on the air endseq
	startseq boy is riding up on the ocean endseq

2.7 Tahap Evaluasi

Setelah model dilatih, penting untuk menguji kemampuan prediksi pada dataset uji. Dalam konteks evaluasi teks, metrik tradisional seperti akurasi tidak relevan. Sebagai gantinya, peneliti menggunakan skor BLEU *(Bilingual Evaluation Understudy)*, sebuah metrik untuk membandingkan teks kandidat dengan satu atau lebih teks referensi. Sebagai contoh, pada dua hipotesis yang berbeda, peneliti menggunakan skor BLEU untuk mengevaluasi kesamaan antara teks prediksi dan teks referensi. Misalnya, skor BLEU untuk hipotesis pertama adalah 0.603 dan untuk hipotesis kedua adalah 0.544. Berikut adalah rumus yang dapat digunakan untuk menghitung skor BLEU [8]:

$$BLEU = BP \times exp\left(\sum_{n=1}^{N} w_n \times log(P_n)\right)$$

Keterangan:

BP = Faktor penyusutan *Brevity Penalty* yang mengkompensasi kecenderungan sistem untuk menghasilkan teks yang terlalu pendek.
N = Urutan maksimum n-gram yang dievaluasi (biasanya 4).
w_n = Bobot yang diberikan pada setiap urutan n-gram.
P_n = Presisi urutan n-gram, yaitu rasio jumlah urutan n-gram yang cocok dalam hasil sistem terhadap jumlah urutan n-gram dalam teks referensi.

Tabel 2. Hasil Evaluasi

Hipotesis	Referensi	Skor BLEU
I like dog	I do like dog	0.603
I love dog!	I do like dog	0.544

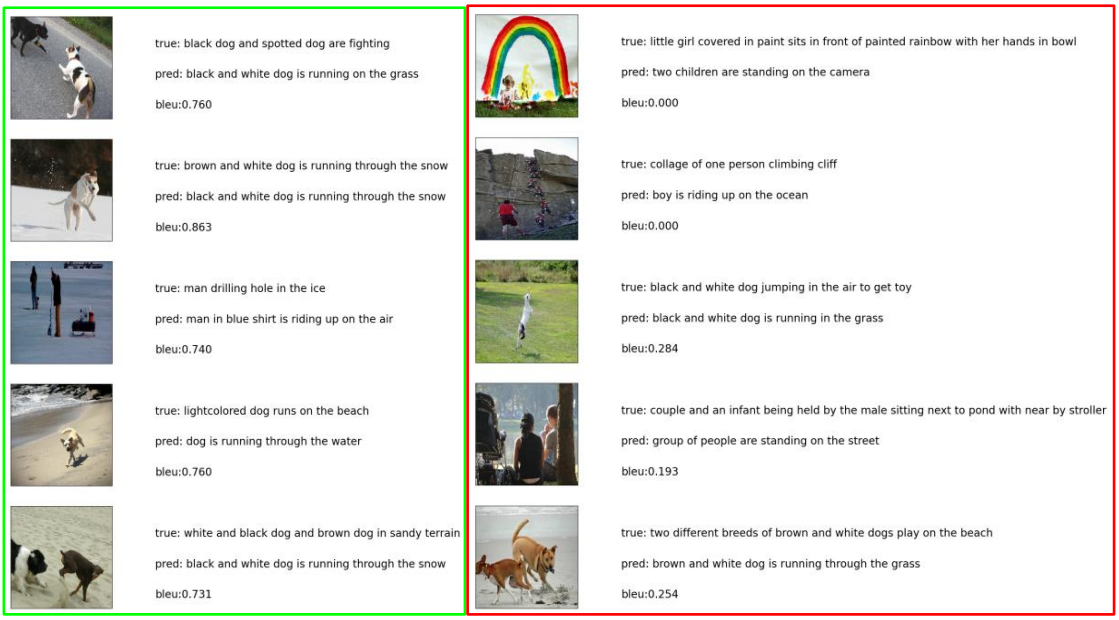
Metrik BLEU digunakan sebagai alat untuk mengevaluasi sejauh mana terjemahan yang dihasilkan oleh model (*hypothesis*) cocok dengan terjemahan referensi (*reference*). Skor BLEU berkisar dari 0 hingga 1, dengan nilai yang lebih tinggi menunjukkan kesamaan yang lebih baik antara terjemahan hypothesis dan referensi. Dalam contoh yang diberikan, hipotesis pertama "I like dog" memiliki skor BLEU sebesar 0.603 ketika dibandingkan dengan referensi "I do like dog". Sementara itu, hipotesis kedua "I love dog!" mendapat skor BLEU sebesar 0.544 terhadap referensi yang sama. Dari skor BLEU ini, dapat disimpulkan bahwa hipotesis pertama lebih mirip

p-ISSN: 2986-3929
e-ISSN: 3032-1948

dengan referensi daripada hipotesis kedua.

3. Hasil dan Pembahasan

Setelah semua dilakukan semua proses pada tahap sebelumnya, penelitian ini akan menghasilkan caption untuk seluruh data uji dan mengevaluasinya menggunakan skor BLEU. Dalam proses ini, setiap gambar dari data uji diproses secara individual. Untuk setiap gambar, caption yang dihasilkan oleh model dibandingkan dengan caption referensi yang sebenarnya menggunakan metrik BLEU. Skor BLEU yang dihasilkan memberikan gambaran tentang seberapa baik caption yang dihasilkan oleh model cocok dengan caption referensi. Peneliti melakukan iterasi melalui setiap gambar dari data uji. Untuk setiap gambar, caption yang dihasilkan oleh model dibandingkan dengan caption referensi yang sebenarnya. Jika skor BLEU lebih tinggi dari 0.7, caption tersebut dianggap baik dan dimasukkan ke dalam kategori "caption baik". Sebaliknya, jika skor BLEU kurang dari 0.3, caption tersebut dianggap buruk dan dimasukkan ke dalam kategori "caption buruk". Contoh-caption baik dan buruk juga diberikan untuk memberikan pemahaman visual tentang kualitas caption yang dihasilkan oleh model. Dengan melihat contoh-caption baik dan buruk, peneliti dapat memahami secara lebih konkret tentang kekuatan dan kelemahan model dalam menghasilkan caption untuk gambar-gambar tertentu.



Gambar 12. Pembangkitan Caption Baik dan Buruk

Keterangan:

Hijau = Hasil pembangkitan caption yang baik.
Merah = Hasil Pembangkitan caption yang buruk.

Peneliti juga menghitung rata-rata skor BLEU dari semua caption setelah semua tahapan selesai dilakukan. Dalam kasus ini, rata-rata skor BLEU adalah 0.415. Ini mengindikasikan bahwa secara keseluruhan, model memiliki tingkat kesesuaian yang cukup baik dengan caption referensi. Untuk menghitung rata-rata BLEU dari sejumlah teks yang dievaluasi, dapat digunakan rumus berikut [8]:

$$Rata - rata\ BLEU = \frac{\sum_{i=1}^{N} BLEU_i}{N}$$

Keterangan:

$BLEU$ = Skor BLEU untuk teks ke-i.
N = Jumlah teks yang dievaluasi.

```
In [71]: print("Rata-rata BLEU {:4.3f}".format(np.mean(bleus)))

Rata-rata BLEU 0.415
```

Gambar 13. Rata-rata Skor BLEU dari Semua Caption

3.1 Dokumentasi

Hasil penelitian tertuang dalam bentuk file PDF yang merupakan konversi dari *Jupyter Notebook* dengan bahasa python yang dapat diakses disini. Dokumen ni mencakup langkah-langkah eksperimen, analisis, dan temuan utama dalam pengembangan model secara lengkap. Penelitian ini diharapkan dapat menjadi sumber wawasan bagi peneliti lain dalam bidang Pembangkitan Caption Gambar Otomatis, memberikan inspirasi untuk eksplorasi lebih lanjut, serta mendorong pengembangan model yang lebih canggih dan akurat.

4. Kesimpulan

Berdasarkan paparan penelitian yang telah dilakukan sebelumnya, penelitian ini telah berhasil menerapkan model LSTM pada dataset Flickr8k untuk menghasilkan caption gambar yang memuaskan, meskipun masih terdapat ruang untuk peningkatan. Evaluasi menggunakan skor BLEU menunjukkan bahwa model mencapai rata-rata skor BLEU sebesar 0.415 atau 41.5% untuk seluruh dataset uji yang digunakan untuk mengevaluasi kinerja model. Untuk pengembangan lebih lanjut, disarankan untuk mengeksplorasi arsitektur model yang lebih mutakhir, seperti CNN, RNN, dll. Selain itu, penerapan teknik data augmentasi dapat membantu meningkatkan kinerja model. Integrasi informasi kontekstual tambahan, seperti objek dan aktivitas dalam gambar, juga dapat diperluas untuk meningkatkan keakuratan caption. Dengan penelitian dan pengembangan lebih lanjut, diharapkan dapat tercipta model yang lebih unggul dalam menghasilkan caption gambar yang informatif dan bermanfaat dalam berbagai aplikasi praktis.

Daftar Pustaka

[1] H. Wang, Y. Zhang, and X. Yu, “An overview of image caption generation methods,” Computational Intelligence and Neuroscience, vol. 2020, pp. 1–13, Jan. 2020. doi:10.1155/2020/3062706.
[2] A. R. GRIGOREV, Tensorflow Deep Learning Projects: 10 Real-World Projects on Computer Vision, Machine Translation, Chatbots, and Reinforcement Learning;10 Real-World. PACKT Publishing, 2018.
[3] K. Anitha Kumari, C. Mouneeshwari, R. B. Udhaya, and R. Jasmitha, “Automated image captioning for Flickr8k dataset,” Proceedings of International Conference on Artificial Intelligence, Smart Grid and Smart City Applications, pp. 679–687, 2020. doi:10.1007/978-3-030-24051-6_62.
[4] adityajn105, “Flickr 8K dataset,” *Kaggle*, 27-Apr-2020. [Online]. Available: https://www.kaggle.com/datasets/adityajn105/flickr8k. [Accessed: 01-May-2024].
[5] B. Jawade, D. D. Mohan, N. M. Ali, S. Setlur, and V. Govindaraju, “NAPReg: Nouns as proxies’ regularization for semantically aware cross-modal embeddings,” 2023 IEEE/CVF Winter Conference on Applications of Computer Vision (WACV), Jan. 2023. doi:10.1109/wacv56688.2023.00119.
[6] T. V. Sneha and Dr. S. J. Rani, LSTM-VGG-16: A Novel and Modular Model for Image Captioning Using Deep Learning Approachesge captioning for Flickr8k dataset, vol. 12, no. 11, pp. 131–141, 2021.
[7] C. Wang, H. Yang, C. Bartz, and C. Meinel, “Image captioning with deep bidirectional lstms,” *Proceedings of the 24th ACM international conference on Multimedia*, Oct. 2016.

doi:10.1145/2964284.2964299.
[8] GeeksforGeeks, “NLP - Bleu score for Evaluating Neural Machine Translation - Python,” *GeeksforGeeks*, 08-Mar-2024. [Online]. Available: https://www.geeksforgeeks.org/nlp-bleu-score-for-evaluating-neural-machine-translation-python/. [Accessed: 01-May-2024].

p-ISSN: 2986-3929
e-ISSN: 3032-1948

Halaman ini sengaja dibiarkan kosong