

**SISTEM *IMAGE CAPTIONING* UNTUK LUKA *DIABETIC FOOT ULCER*  
MENGUNAKAN VGG16 DAN *BIDIRECTIONAL LONG SHORT TERM*  
*MEMORY***

**LAPORAN TUGAS AKHIR**

Laporan ini Disusun untuk Memenuhi Salah Satu Syarat Memperoleh Gelar Sarjana  
Strata 1 (S1) pada Program Studi Teknik Informatika Fakultas Teknologi Industri  
Universitas Islam Sultan Agung Semarang



**DISUSUN OLEH:**

**DIMAS SURYA WIRASTAMA**

**NIM 32602100039**

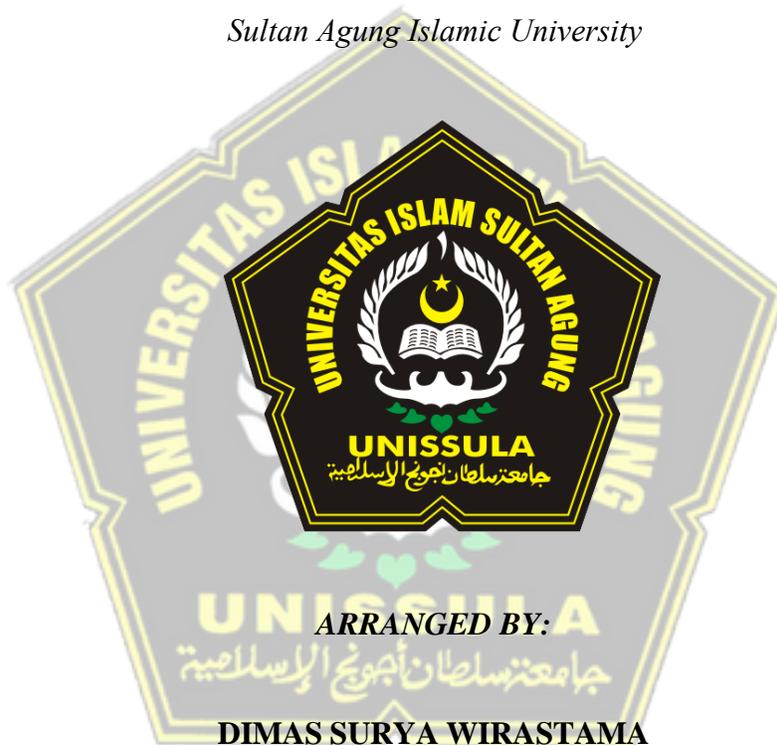
**PROGRAM STUDI TEKNIK INFORMATIKA  
FAKULTAS TEKNOLOGI INDUSTRI  
UNIVERSITAS ISLAM SULTAN AGUNG  
SEMARANG**

**2025**

***IMAGE CAPTIONING SYSTEM FOR DIABETIC FOOT ULCER WOUNDS  
USING VGG16 AND BIDIRECTIONAL LONG SHORT TERM MEMORY***

***FINAL PROJECT***

*Proposed to complete the requirement to obtain a bachelor's degree (SI) at  
Informatics Engineering Departement of Industrial Technology Faculty  
Sultan Agung Islamic University*



***ARRANGED BY:***

**DIMAS SURYA WIRASTAMA**

**32602100039**

***MAJORING OF INFORMATICS ENGINEERING***

***INDUSTRIAL TECHNOLOGY FACULTY***

***SULTAN AGUNG ISLAMIC UNIVERSITY***

**SEMARANG**

**2025**

**LEMBAR PENGESAHAN TUGAS AKHIR**

**SISTEM *IMAGE CAPTIONING* UNTUK LUKA *DIABETIC FOOT ULCER*  
MENGUNAKAN VGG16 DAN *BIDIRECTIONAL LONG SHORT TERM*  
*MEMORY***

**DIMAS SURYA WIRASTAMA**  
NIM 32602100039

Telah dipertahankan di depan tim penguji ujian sarjana tugas akhir  
Program Studi Teknik Informatika  
Universitas Islam Sultan Agung  
Pada tanggal : 27-02-2025

**TIM PENGUJI UJIAN SARJANA :**

**Bagus Satrio Waluyo Poetra,**

**S.kom, M.Cs**

NIK. 210616051

(Ketua Penguji)



21-02-2025

**Imam Much Ibnu Subroto,**

**ST, MSc.PhD**

NIK. 210600017

(Anggota Penguji)

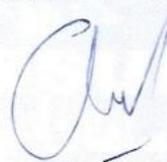


29-02-2025

**Andi Riansyah, ST, M.Kom**

NIK. 210616053

(Pembimbing)



27-02-2025

Semarang, 09-03-2025

Mengetahui,  
Kaprodik Teknik Informatika  
Universitas Islam Sultan Agung

**Moch. Taufik, ST, MIT**

NIK. 210604034

## SURAT PERNYATAAN KEASLIAN TUGAS AKHIR

Yang bertanda tangan dibawah ini :

Nama : Dimas Surya Wirastama

NIM : 32602100039

Judul Tugas Akhir : *SISTEM IMAGE CAPTIONING UNTUK LUKA DIABETIC FOOT ULCER MENGGUNAKAN VGG16 DAN BIDIRECTIONAL LONG SHORT TERM MEMORY.*

Dengan bahwa ini saya menyatakan bahwa judul dan isi Tugas Akhir yang saya buat dalam rangka menyelesaikan Pendidikan Strata Satu (S1) Teknik Informatika tersebut adalah asli dan belum pernah diangkat, ditulis ataupun dipublikasikan oleh siapapun baik keseluruhan maupun sebagian, kecuali yang secara tertulis diacu dalam naskah ini dan disebutkan dalam daftar pustaka, dan apabila di kemudian hari ternyata terbukti bahwa judul Tugas Akhir tersebut pernah diangkat, ditulis ataupun dipublikasikan, maka saya bersedia dikenakan sanksi akademis. Demikian surat pernyataan ini saya buat dengan sadar dan penuh tanggung jawab.

Semarang, 09 Mei 2025

Yang Menyatakan.

  
Dimas Surya Wirastama

## KATA PENGANTAR

Dengan mengucapkan syukur alhamdulillah atas kehadiran Allah SWT yang telah memberikan rahmat dan karunianya kepada penulis, sehingga dapat menyelesaikan Tugas Akhir dengan judul “Sistem *Image Captioning* Untuk Luka *Diabetic Foot Ulcer* Menggunakan VGG16 dan *Bidirectional Long Short Term Memory*” ini untuk memenuhi salah satu syarat menyelesaikan studi serta dalam rangka memperoleh gelar sarjana (S-1) pada Program Studi Teknik Informatika Fakultas Teknologi Industri Universitas Islam Sultan Agung Semarang.

Tugas Akhir ini disusun dan dibuat dengan adanya bantuan dari berbagai pihak, materi maupun teknis, oleh karena itu saya selaku penulis mengucapkan terima kasih kepada:

1. Rektor UNISSULA Bapak Prof. Dr. H. Gunarto, S.H., M.H yang mengizinkan penulis menimba ilmu di kampus ini.
2. Dekan Fakultas Teknologi Industri Ibu Dr. Novi Marlyana, S.T., M.T.
3. Dosen pembimbing I penulis Andi Riansyah, ST, M.Kom yang telah meluangkan waktu dan memberi ilmu. Serta memberikan banyak nasehat dan saran.
4. Orang tua penulis yang telah mengizinkan untuk menyelesaikan laporan ini,
5. Dan kepada semua pihak yang tidak dapat saya sebutkan satu persatu.

Dengan segala kerendahan hati, penulis menyadari masih terdapat banyak kekurangan dari segi kualitas atau kuantitas maupun dari ilmu pengetahuan dalam penyusunan laporan, sehingga penulis mengharapkan adanya saran dan kritikan yang bersifat membangun demi kesempurnaan laporan ini dan masa mendatang.

Semarang, 19 Februari 2024



Dimas Surya Wirastama

## DAFTAR ISI

<b>COVER .....</b>	<b>i</b>
<b>LEMBAR PENGESAHAN TUGAS AKHIR .....</b>	<b>iii</b>
<b>SURAT PERNYATAAN KEASLIAN TUGAS AKHIR.....</b>	<b>iv</b>
<b>PERNYATAAN PERSETUJUAN PUBLIKASI KARYA ILMIAH.....</b>	<b>v</b>
<b>KATA PENGANTAR .....</b>	<b>vi</b>
<b>DAFTAR ISI.....</b>	<b>vii</b>
<b>DAFTAR GAMBAR .....</b>	<b>ix</b>
<b>DAFTAR TABEL .....</b>	<b>x</b>
<b>ABSTRAK .....</b>	<b>xi</b>
<b>BAB I PENDAHULUAN .....</b>	<b>1</b>
1.1 Latar Belakang .....	1
1.2 Perumusan Masalah.....	2
1.3 Pembatasan Masalah.....	2
1.4 Tujuan.....	3
1.5 Manfaat .....	3
1.6 Sistematika Penulisan .....	4
<b>BAB II TINJAUAN PUSTAKA DAN DASAR TEORI .....</b>	<b>5</b>
2.1 Tinjauan Pustaka .....	5
2.2 Dasar Teori.....	8
2.2.1 <i>Diabetic Foot Ulcer (DFU)</i> .....	8
2.2.2 <i>Image Captioning</i> .....	9
2.2.3 <i>Convolutional Neural Network (CNN)</i> .....	10
2.2.4 <i>Visual Geometry Group (VGG16)</i> .....	14
2.2.5 <i>Bidirectional Long Short-Term Memory (BiLSTM)</i> .....	15
2.2.6 <i>Bilingual Evaluation Understudy (BLEU)</i> .....	18
<b>BAB III METODOLOGI PENELITIAN .....</b>	<b>21</b>
3.1 Metode Penelitian.....	21

3.1.1	Studi Literatur .....	22
3.1.2	Pengumpulan Data .....	22
3.1.3	<i>Preprocessing</i> Data .....	23
3.1.4	Training Model.....	24
3.1.5	Evaluasi Model.....	24
3.2	Analisa Kebutuhan .....	25
3.3	Penggunaan Sistem .....	28
3.4	Perancangan <i>User Interface</i> .....	29
3.4.1	Halaman Awal Sistem.....	29
3.4.2	Tampilan Hasil Sistem .....	30
<b>BAB IV HASIL DAN ANALISIS PENELITIAN .....</b>		<b>31</b>
4.1	Hasil Pengumpulan Data.....	31
4.2	Hasil <i>Preprocessing</i> Data.....	33
4.2.1	<i>Image Preprocessing</i> .....	33
4.2.2	Teks <i>Preprocessing</i> .....	34
4.3	Hasil <i>Modeling</i> .....	38
4.3.1	Model VGG16.....	38
4.3.2	Model <i>Image Captioning</i> .....	39
4.4	Hasil Training.....	40
4.5	Hasil Evaluasi.....	43
4.6	Hasil Implementasi.....	45
<b>BAB V KESIMPULAN DAN SARAN.....</b>		<b>48</b>
5.1	Kesimpulan .....	48
5.2	Saran.....	48
<b>DAFTAR PUSTAKA</b>		

## DAFTAR GAMBAR

Gambar 2. 1 Visualisai <i>Image Captioning</i> .....	9
Gambar 2. 2 Arsitektur dasar CNN (Satwikayana dkk., 2021) .....	10
Gambar 2. 3 <i>Convoluton layer</i> (Agustiono dkk., 2024) .....	11
Gambar 2. 4 <i>Max pooling</i> (Agustiono dkk., 2024) .....	12
Gambar 2. 5 <i>Fully connected layer</i> (Dabbo dan Bisilisin, 2024).....	13
Gambar 2. 6 Arsitektur VGG16 (Paneru dkk., 2024).....	14
Gambar 2. 7 Arsitektur BiLSTM (Ge dan Han, 2024) .....	15
Gambar 2. 8 Arsitektur LSTM (Lan dkk., 2024) .....	16
Gambar 3. 1 Tahap Penelitian .....	21
Gambar 3. 2 <i>Flowchart</i> alur pengumpulan data .....	23
Gambar 3. 3 Training VGG16 dan BiLSTM .....	24
Gambar 3. 4 <i>Flowchart</i> alur kerja sistem .....	28
Gambar 3. 5 Tampilan halaman awal sistem.....	29
Gambar 3. 6 Tampilan hasil sistem .....	30
Gambar 4. 1 Dataset gambar <i>diabetic foot ulcer</i> .....	31
Gambar 4. 2 <i>Source code Image preprocessing</i> .....	33
Gambar 4. 3 <i>Source code</i> untuk teks <i>preprocessing</i> .....	34
Gambar 4. 4 <i>Source code</i> tokenisasi dan <i>padding</i> .....	36
Gambar 4. 5 Hasil <i>source code</i> tokenisasi dan <i>padding</i> .....	37
Gambar 4. 6 <i>Source code</i> model VGG16 .....	38
Gambar 4. 7 <i>Source code modeling image captioning</i> .....	39
Gambar 4. 8 Halaman utama.....	46
Gambar 4. 9 Hasil <i>image captioning</i> .....	47

## DAFTAR TABEL

Table 4. 1 Dataset <i>caption diabetic foot ulcer</i> .....	32
Table 4. 2 Hasil <i>lowercasing</i> .....	35
Table 4. 3 Hasil penghapusan karakter <i>non-alfabet</i> .....	35
Table 4. 4 Hasil penambahan token .....	36
Table 4. 5 <i>Plot accuracy</i> dan <i>loss</i> .....	40
Table 4. 6 Hasil terbaik berdasarkan akurasi.....	42
Table 4. 7 Perbandingan hasil skor matrik BLEU .....	44



## ABSTRAK

Diabetes melitus merupakan penyakit kronis dengan gangguan metabolik yang ditandai oleh peningkatan kadar gula darah, yang pada akhirnya dapat menyebabkan berbagai komplikasi serius, salah satunya adalah *Diabetic Foot Ulcer* (DFU). DFU merupakan luka terbuka pada area kaki yang, apabila tidak ditangani dengan tepat, dapat menyebabkan infeksi parah hingga amputasi. Rendahnya tingkat pengetahuan pasien dan keluarga pasien mengenai perawatan dan pemantauan kondisi luka turut meningkatkan risiko terjadinya komplikasi tersebut. Penelitian ini bertujuan mengembangkan sistem *image captioning* berbasis *Deep Learning* untuk mendeskripsikan kondisi luka DFU secara objektif. Sistem dikembangkan dengan mengombinasikan model VGG16 sebagai *encoder* dan BiLSTM sebagai *decoder*. Dataset gambar DFU diperoleh dari *platform* Kaggle, sedangkan dataset *caption* DFU dikumpulkan melalui diskusi dengan ahli dokter. Hasil pelatihan menunjukkan bahwa model mencapai keseimbangan terbaik pada epoch 20 dengan akurasi sebesar 0.9466 dan loss yang relatif rendah 0.1837. Evaluasi model dilakukan dengan menggunakan metrik BLEU menunjukkan bahwa performa terbaik diperoleh pada epoch 100 dengan skor BLEU-1 sebesar 0.4511, BLEU-2 sebesar 0.3716, BLEU-3 sebesar 0.3455, dan BLEU-4 sebesar 0.3254. Hasil tersebut mengindikasikan bahwa sistem mampu menghasilkan *caption* yang mendekati struktur dan makna dari *caption* referensi. Penelitian ini menunjukkan potensi penerapan teknologi *Deep Learning* dalam meningkatkan pemahaman pasien dan keluarga pasien terhadap kondisi DFU, yang diharapkan dapat berkontribusi pada peningkatan kualitas diagnosis dan perawatan melalui penerapan *Generative AI* dalam bidang kesehatan.

Kata Kunci : *Image captioning, Deep Learning, Diabetic foot ulcer, Generative AI*

## ABSTRACT

*Diabetes mellitus is a chronic disease with metabolic disorders characterized by increased blood sugar levels, which can eventually lead to various serious complications, one of which is Diabetic Foot Ulcer (DFU). DFU is an open wound in the leg area that, if not treated properly, can cause severe infection up to amputation. The low level of knowledge of patients and their families regarding the treatment and monitoring of wound conditions also increases the risk of these complications. This study aims to develop a Deep Learning-based image captioning system to objectively describe the condition of DFU wounds. The system was developed by combining the VGG16 model as an encoder and BiLSTM as a decoder. DFU image datasets were obtained from the Kaggle platform, while DFU caption datasets were collected through discussions with physician experts. The training results show that the model achieves the best equilibrium at epoch 20 with an accuracy of 0.9466 and a relatively low loss of 0.1837. The model evaluation was carried out using BLEU metrics showing that the best performance was obtained at epoch 100 with a BLEU-1 score of 0.4511, BLEU-2 of 0.3716, BLEU-3 of 0.3455, and BLEU-4 of 0.3254. The results indicate that the system is able to generate captions that are close to the structure and meaning of the reference captions. This study shows the potential of the application of Deep Learning technology in improving the understanding of patients and their families about the condition of DFU, which is expected to contribute to improving the quality of diagnosis and treatment through the application of Generative AI in the health sector.*

*Keywords : Image captioning, Deep Learning, Diabetic foot ulcer, Generative AI*

# BAB I

## PENDAHULUAN

### 1.1 Latar Belakang

Diabetes melitus adalah salah satu penyakit kronis yang ditandai oleh gangguan metabolik berupa peningkatan kadar gula darah akibat produksi insulin yang tidak mencukupi atau respon tubuh yang tidak efektif terhadap insulin (Said dkk., 2024). Diabetes melitus juga dikenal sebagai *the silent killer* karena dapat menyebabkan berbagai komplikasi serius di berbagai organ tubuh dalam jangka panjang (Rif'at dkk., 2023). Salah satu komplikasi penyakit diabetes melitus adalah *Diabetic Foot Ulcer* (DFU).

*Diabetic Foot Ulcer* (DFU) merupakan salah satu komplikasi serius yang ditandai dengan adanya luka terbuka pada kulit terutama di area kaki akibat kadar gula darah yang tidak terkontrol. DFU dapat menyebabkan berbagai dampak negatif seperti infeksi yang parah hingga amputasi (Vitniawati dkk., 2023). Salah satu faktor terhadap terjadinya DFU adalah rendahnya tingkat pengetahuan keluarga pasien mengenai perawatan luka dan pemantauan kondisi luka. Pengetahuan keluarga pasien sangat berperan dalam mendukung pemantauan diri pasien dalam perawatan kaki untuk mencegah adanya luka atau infeksi yang lebih parah hingga berujung amputasi (Agustari dkk., 2022).

Dalam penanganan DFU teknologi *image captioning* berbasis *Deep Learning* menawarkan solusi inovatif untuk meningkatkan akurasi diagnosis melalui *image captioning* pada gambar luka. Dengan menggunakan kombinasi *Convolutional Neural Network* (CNN) seperti arsitektur VGG16 dan *Bidirectional Long Short Term Memory* (BiLSTM), sistem ini memiliki potensi untuk menghasilkan *image captioning* pada luka yang objektif dan relevan. sistem ini memungkinkan dapat membantu pasien atau keluarga pasien untuk memahami kondisi luka DFU secara lebih baik dan memberikan perawatan yang tepat.

Teknologi *image captioning* ini sangat efektif untuk meningkatkan pemahaman pasien dan keluarga tentang kondisi luka yang dialami pasien, agar dapat mengurangi resiko infeksi yang lebih parah hingga amputasi. Pada penelitian sebelumnya oleh (Satria dan Pardede, 2022) dengan judul “*Image Captioning Menggunakan Metode ResNet50 dan Long Short Term Memory*” menunjukkan bahwa kombinasi ResNet50 dan LSTM mampu menghasilkan skor akurasi sebesar 85,74%. Tetapi dataset yang digunakan masih kurang relevan dengan luka *diabetic foot ulcer*.

Oleh karena itu, penulis bertujuan untuk mengembangkan sistem *image captioning* yang fokus pada dataset gambar luka *diabetic foot ulcer*, dengan menggunakan kombinasi model VGG16 dan BiLSTM. Sistem ini diharapkan dapat membantu pasien dan keluarga pasien dalam menganalisis kondisi luka *diabetic foot ulcer* secara objektif, sehingga memberikan pemahaman yang lebih baik mengenai kondisi luka yang dialami pasien. Dengan adanya sistem ini diharapkan resiko komplikasi serius hingga amputasi dapat diminimalkan dan kualitas hidup pasien dapat ditingkatkan.

## 1.2 Perumusan Masalah

Bagaimana mengimplementasikan *Deep Dearning* dengan metode VGG16 dan BiLSTM untuk menghasilkan *Image Captioning* pada luka *Diabetic Foot Ulcer* (DFU) yang dapat memberikan deskripsi objektif guna meningkatkan pemahaman pasien dan keluarga terhadap kondisi luka DFU.

## 1.3 Pembatasan Masalah

Batasan masalah ini bertujuan untuk memudahkan dan menghindari adanya kegiatan di luar sasaran, sehingga dalam pembuatan laporan ini perlu ditentukan suatu batasan masalah. Batasan masalah tersebut sebagai berikut :

1. Dataset yang digunakan pada penelitian ini diperoleh dari hasil wawancara dan diskusi dengan dokter spesialis *diabetic foot ulcer*, yang mencakup

deskripsi pada gambar yang relevan.

2. Penelitian ini berfokus pada penerapan model VGG16 dan BiLSTM untuk proses *image captioning*, tanpa mencakup analisis model atau metode lainnya.
3. *Caption* yang dihasilkan oleh model akan dibatasi pada penggunaan bahasa Indonesia. Penelitian ini tidak akan mencakup penerjemahan otomatis atau multi-bahasa.
4. Deskripsi yang dihasilkan oleh model akan difokuskan pada tingkat keparahan luka *diabetic foot ulcer*, penelitian ini tidak mencakup informasi medis lainnya.

#### 1.4 Tujuan

Adapun tujuan dari penelitian ini adalah untuk mengembangkan sistem *Image Captioning* dengan menggunakan kombinasi model VGG16 dan BiLSTM untuk menghasilkan *caption* otomatis pada gambar luka *Diabetic Foot Ulcer (DFU)*.

#### 1.5 Manfaat

Penelitian ini diharapkan dapat memberikan manfaat bagi Pasien dan Keluarga Pasien. Dengan adanya sistem *Image Captioning* yang dapat memahami mengenai kondisi luka pasien dan keluarga pasien dapat lebih memahami perawatan luka yang diperlukan, sehingga meningkatkan kesadaran dan kepatuhan terhadap penanganan medis.

## 1.6 Sistematika Penulisan

Untuk mempermudah penulisan tugas akhir ini, penulis membuat suatu sistematika yang terdiri dari:

**BAB I : PENDAHULUAN**

Bab ini menjelaskan mengenai latar belakang pemilihan judul, rumusan masalah, batasan masalah, tujuan penelitian, manfaat penelitian, metodologi penelitian, dan sistematika penulisan.

**BAB II : TINJAUAN PUSTAKA DAN DASAR TEORI**

Bab ini memuat penelitian-penelitian sebelumnya dan dasar teori yang berfungsi sebagai sumber atau alat dalam memahami permasalahan yang berkaitan dengan implementasi VGG16 dan BiLSTM.

**BAB III : METODOLOGI PENELITIAN**

Bab ini mengungkapkan proses tahapan - tahapan penelitian dimulai dari analisa kebutuhan sistem, kemudian perancangan sistem hingga *prototype*.

**BAB IV : HASIL DAN ANALISIS SISTEM**

Pada bab ini penulis mengungkapkan hasil penelitian yakni hasil dari pembuatan *image captioning*.

**BAB V : KESIMPULAN DAN SARAN**

Bab terakhir memuat kesimpulan isi dari keseluruhan uraian bab-bab sebelumnya dan saran-saran dari hasil yang diperoleh dan diharapkan dapat bermanfaat dalam pengembangan selanjutnya.

## BAB II

### TINJAUAN PUSTAKA DAN DASAR TEORI

#### 2.1 Tinjauan Pustaka

*Diabetic Foot Ulcer* (DFU) merupakan salah satu komplikasi serius pada penderita diabetes melitus (DM) yang dapat menyebabkan dampak seperti infeksi dan amputasi. Faktor resiko DFU meliputi gula darah yang buruk, obesitas, merokok, usia lanjut dan perawatan kaki yang tidak memadai. Oleh karena itu deteksi dini dan pemantauan luka DFU sangat penting untuk mencegah komplikasi lebih lanjut serta mengurangi resiko infeksi dan amputasi (Vitniawati dkk., 2023). Salah satu solusi yang dapat diimplementasikan adalah pengembangan sistem *image captioning* yang mampu menghasilkan deskripsi otomatis untuk gambar luka DFU. Sistem ini diharapkan dapat membantu pasien dalam memahami tingkat keparahan luka secara lebih efektif.

*Image captioning* adalah proses untuk menghasilkan *text* deskripsi pada gambar secara otomatis. Teknologi ini menggabungkan pengolahan gambar computer vision dengan pemrosesan bahasa alami seperti *Natural Language Processing* (NLP) untuk menghasilkan kalimat deskripsi yang sesuai dengan konten visual. Teknologi *Image Captioning* telah lama dikembangkan namun dalam skala publik masih terbatas (Taufiqur dan Hatta, 2022). Dari hasil penelitian oleh (Farhan dan Setiaji, 2023) ini menghasilkan deskripsi otomatis pada rambu lalu lintas Indonesia dengan menggunakan model *Inception V3* sebagai *ecoder* dan *Transformer* sebagai *decoder*, dataset yang digunakan gambar rambu lalu lintas berjumlah 9.594 gambar, model yang dikembangkan menunjukkan kinerja yang baik dengan tingkat akurasi mencapai 80% pada 50 *epoch*. Hal ini menunjukkan bahwa model yang dikembangkan dapat memberikan kontribusi signifikan dalam meningkatkan pemahaman terhadap lalu lintas Indonesia.

Dari hasil penelitian sebelumnya yang telah dilakukan oleh (Pratiwi dan

Pardede, 2022), dalam penelitiannya ini menggunakan dua metode yaitu *Inception-V3* dan *Transformer* untuk melakukan *image captioning*. Dataset yang digunakan adalah Flickr8k yang terdiri dari 8.000 gambar, kedua model dievaluasi menggunakan metrik BLEU (*Bilingual Evaluation Understudy*), yang terdiri dari empat parameter utama: BLEU-1 sebesar 0.306, BLEU-2 sebesar 0.184, BLEU-3 sebesar 0.123, dan BLEU-4 sebesar 0.048. Dari hasil penelitian di atas tentang *image captioning*, ini menunjukkan adanya peluang untuk penelitian *image captioning* terutama dalam konteks medis, seperti *image captioning* untuk gambar luka *diabetic foot ulcer* dengan menggunakan model VGG16 dan BiLSTM.

Dari penelitian sebelumnya oleh (Agustina dkk., 2022) dalam penelitiannya ini menggunakan model CNN dengan menggunakan arsitektur VGG16. Dataset yang digunakan terdiri dari 4000 citra jaringan kanker, yang diklasifikasi ke dalam empat kelas. Hasil dari pengujian ini menunjukkan bahwa sistem menghasilkan performa terbaik dengan konfigurasi 50 *epoch* dan *batch size* 32, dapat mencapai akurasi 99,70%. Hasil akurasi ini menunjukkan bahwa arsitektur VGG16 memiliki kemampuan tinggi dalam mengenali dan mengklasifikasikan citra kanker kulit dengan baik.

Dari penelitian sebelumnya oleh (Supirman dkk., 2023), pada penelitiannya ini memfokuskan pada klasifikasi penyakit kulit menjadi tujuh jenis, yaitu dermatitis, campak, herpes, psoriasis, cacar air, kurap dan kutil. Metode yang digunakan *Convolutional Neural Network* (CNN) dengan perbandingan dua arsitektur model dari CNN yaitu VGG16 dan MobileNet. Dengan hasil proses pelatihan pada 100 *epoch*, VGG16 memiliki akurasi pengujian sebesar 82% lebih tinggi dibandingkan MobileNet yang hanya memiliki akurasi pengujian 80%. Penelitian ini menunjukkan bahwa arsitektur VGG16, memberikan performa yang baik dalam klasifikasi penyakit kulit berbasis data citra digital.

Penelitian sebelumnya oleh (Noorizki dan Kusumawati, 2023), dalam

penelitiannya ini membandingkan kinerja dua arsitektur model dari algoritma CNN, yaitu VGG16 dan VGG19 menggunakan dataset dengan lima kelas varietas beras. Hasil penelitian ini menunjukkan bahwa VGG16 memiliki keunggulan akurasi sebesar 98%, sedangkan VGG19 mencapai akurasi 97%. Keunggulan VGG16 disebabkan oleh kompleksitas model yang lebih rendah, sehingga mampu menghindari *overfitting* dan jumlah lapisan yang lebih sedikit pada VGG16 memungkinkan proses pelatihan menjadi sepat dan optimal dalam dataset yang tersedia.

Dari hasil penelitian dari (Rolangon dkk., 2023), dalam penelitiannya ini membandingkan empat model yaitu LSTM, BiLSTM, GRU dan SimpleRNN. Data yang digunakan dari Twitter API dengan kata kunci “pelayanan RS” dan “layanan rumah sakit”, data yang dihasilkan sebanyak 13.321 tweet setelah melakukan tahap *preprocessing*. Hasil penelitian ini menunjukkan bahwa model BiLSTM lebih unggul daripada model lainnya dengan akurasi sebesar 86%, dibandingkan dengan model lainnya yaitu, model GRU dengan akurasi 86%, model LSTM dengan akurasi 85% dan model SimpleRNN dengan akurasi 75%. Keunggulan model BiLSTM terletak pada kemampuannya yang bisa menangkap konteks data teks secara lebih baik melalui pemrosesan informasi secara dua arah.

Penelitian sebelumnya oleh (Nugroho dkk., 2023), dalam penelitiannya ini membandingkan model *Bidirectional Long Short-Term Memory* (BiLSTM) dan *Long Short-Term Memory* (LSTM). Dataset yang digunakan dari tweet, dengan jumlah data 2.751 tweet berbahasa Indonesia yang dibedakan menjadi dua label. Hasil pengujian menunjukkan bahwa BiLSTM memiliki performa yang lebih baik dibandingkan dengan LSTM, akurasi dari BiLSTM sebesar 94.12% sedangkan akurasi dari LSTM adalah 84.91%. Hal ini menunjukkan bahwa BiLSTM mampu memanfaatkan konteks secara lebih efektif untuk klasifikasi teks seperti deteksi depresi dan kecemasan.

Kemudian dari hasil penelitian (Alghifari dkk., 2022), dalam hasil

pengujian pada penelitian ini membuktikan bahwa BiLSTM menghasilkan akurasi terbaik sebesar 91%, dibandingkan dengan LSTM hanya mencapai akurasi 76%, serta algoritma lainnya seperti Multinomial *Naive Bayes* akurasi 63%, *Logistic Regression* akurasi 67% dan SVC akurasi 65%. Keunggulan BiLSTM terletak pada kemampuannya yang dapat membaca informasi secara dua arah, sehingga menghasilkan representasi konteks yang lebih relevan.

Dari hasil penelitian penelitian terdahulu penulis dapat mengambil kesimpulan bahwa model VGG16 dan model BiLSTM dapat dikombinasikan untuk membangun sistem *image captioning*, khususnya untuk pemantauan luka *diabetic foot ulcer*. Dalam sistem yang akan dikembangkan, model VGG16 memiliki peran sebagai *encoder* yang bertugas untuk mengekstrak fitur visual dari gambar *diabetic foot ulcer*, sedangkan model BiLSTM memiliki peran sebagai *decoder* yang bertugas untuk mengolah fitur visual yang telah diekstraksi oleh *encoder* menjadi deskripsi teks. Kombinasi kedua model ini memberikan potensi untuk menghasilkan deskripsi otomatis yang sesuai dengan kondisi luka *diabetic foot ulcer*, agar dapat membantu tenaga medis maupun keluarga pasien dalam memahami kondisi luka yang dialami pasien.

## 2.2 Dasar Teori

### 2.2.1 *Diabetic Foot Ulcer* (DFU)

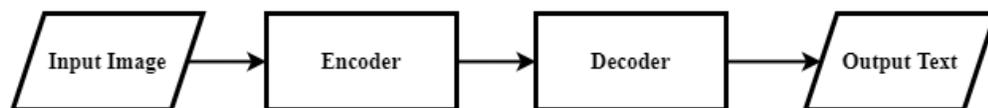
*Diabetic Foot Ulcer* (DFU) merupakan salah satu komplikasi pada penyakit diabetes melitus (DM) yang menjadi tantangan besar dalam bidang kesehatan masyarakat. DFU adalah kondisi kronis yang ditandai dengan adanya luka terbuka di area kaki akibat kontrol glukosa darah yang tidak terkontrol. Kondisi ini dapat menyebabkan berbagai dampak negatif seperti infeksi dan amputasi jika tidak ditangani dengan baik. Faktor resiko utama meliputi kontrol glukosa darah yang buruk, tekanan mekanis pada kaki, keterlambatan diagnosis dan perawatan. Pencegahan dan pengelolaan DFU memerlukan pendekatan yang

mencakup edukasi pasien, kontrol glukosa darah, kontrol infeksi dan penggunaan teknologi medis modern (Wang dkk., 2022).

### 2.2.2 *Image Captioning*

*Image captioning* adalah proses menghasilkan deskripsi teks otomatis dari gambar, *image captioning* dapat memberikan pemahaman tentang informasi yang ingin disampaikan oleh gambar tersebut. *Image captioning* dilakukan dengan cara mengintegrasikan dua teknologi utama yaitu *computer vision* dan *Natural Language Processing* (NLP). *Computer vision* digunakan untuk mengenal dan mengekstrak informasi visual gambar seperti objek, pola, dan fitur penting dari gambar, sedangkan NLP berfungsi untuk menyusun kalimat deskripsi teks berdasarkan informasi visual tersebut. Dalam penggunaan *image captioning* dapat diaplikasikan di berbagai bidang seperti virtual assistant, *image indexing*, interaksi manusia dan computer, dan berbagai macam aplikasi lainnya yang menggunakan NLP (Taufiqur dan Hatta, 2022).

*Image captioning* melibatkan dua tahap yaitu *encoder* dan *decoder*, pada bagian *encoder* bertugas untuk mengekstrak fitur visual dari gambar yang menggunakan model *Deep Learning* seperti *Convolutional Neural Networks* (CNN). *Encoder* berfungsi untuk memetakan data visual ke dalam representasi fitur. Sedangkan bagian *decoder* bertugas untuk menghasilkan teks deskripsi yang menggunakan model berbasis NLP seperti *Long Short-Term Memory* (LSTM). *Decoder* berfungsi untuk menerjemahkan representasi fitur menjadi kalimat yang bermakna (Pratiwi dan Pardede, 2022).



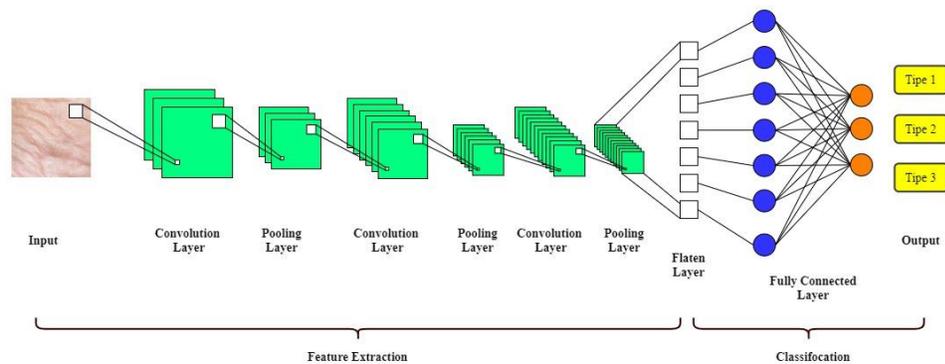
Gambar 2. 1 Visualisasi *Image Captioning*

Gambar 2. 1 adalah tampilan visualisasi *image captioning* yang melalui tahapan-tahapan dimulai dari *input image* memasukkan gambar ke sistem, *encoder* mengekstrak fitur visual dari gambar untuk menghasilkan *vector* fitur,

*decoder* menggunakan *vector* fitur untuk menghasilkan deskripsi teks, dan *output text* keluaran teks deskripsi otomatis dari gambar,

### 2.2.3 Convolutional Neural Network (CNN)

*Convolutional Neural Network* (CNN) dirancang khusus untuk mengolah data dua dimensi seperti gambar, sura, dan teks. CNN adalah salah satu jenis jaringan saraf tiruan yang khusus dirancang untuk memproses data seperti gambar. CNN sangat efektif dalam menyelesaikan berbagai masalah seperti pengenalan pola, klasifikasi citra dan deteksi objek. CNN mampu mengekstrak pola visual seperti tepi, tekstur, dan pola kompleks yang relevan dengan klasifikasi atau prediksi lainnya (Ferani dkk., 2022).



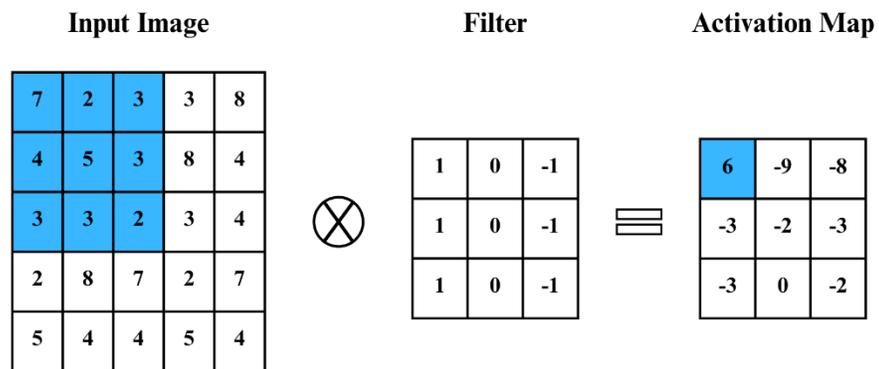
Gambar 2. 2 Arsitektur dasar CNN (Satwikayana dkk., 2021)

Pada Gambar 2. 2 adalah tampilan arsitektur CNN yang terdiri dari tiga jenis lapisan dasar yaitu, *Convolutional Layer*, *Pooling Layer* dan *Fully Connected Layer*.

#### 1. *Convolution Layer*

*Convolution layer* adalah lapisan pertama dalam CNN yang berfungsi untuk mengkonvolusi data menggunakan filter atau kernel dengan ukuran tetap. Filter ini bergerak melintasi *input* data seperti gambar dan melakukan operasi konvolusi yang mengarah pada ekstraksi fitur dari data tersebut. Hasil dari operasi konvolusi ini adalah *feature maps*, yang menggambarkan representasi dari fitur-fitur penting dalam data. *Convolution layer* bertujuan

untuk mengekstrak fitur yang relevan dari data, seperti tepi, tekstur, garis warna dan pola pada citra, yang digunakan untuk klasifikasi (Ferani dkk., 2022).



Gambar 2. 3 *Convolution layer* (Agustiono dkk., 2024)

Gambar 2. 3 menggambarkan konsep dasar *convolution layer* bekerja dengan menggunakan kernel untuk melakukan proses konvolusi terhadap gambar *input*.

Rumus *output convolution layer* :

$$Output = \frac{W - N - 2P}{S} + 1 \quad (1)$$

W = Input lebar atau tinggi citra

N = Ukuran filter untuk proses konvolusi misalnya (3 x 3)

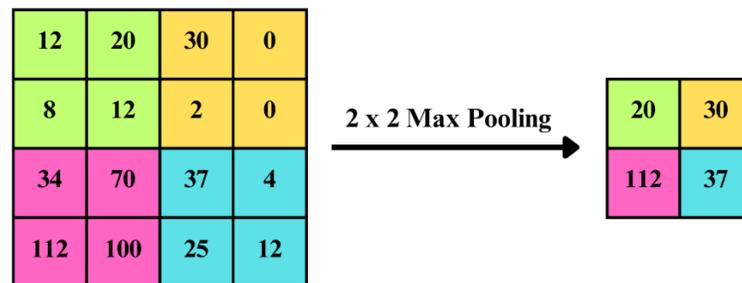
P = Padding untuk menambahkan ruang kosong disekitar citra.

S = Stride adalah langkah pergeseran filter saat menggeser filter di atas citra.

## 2. *Pooling Layer*

*Pooling Layer* digunakan untuk mengurangi jumlah parameter dalam feature map dengan mempertahankan informasi penting. Proses ini dilakukan dengan down-sampling, seperti *Max Pooling* untuk memilih nilai maksimum dalam area tertentu atau *Average Pooling* untuk menghitung rata-rata nilai dalam area tertentu. *pooling layer* mampu memilih fitur

dominan dan mempercepat proses pelatihan dengan mengurangi jumlah parameter yang harus diproses (Supirman dkk., 2023).



Gambar 2. 4 *Max pooling* (Agustiono dkk., 2024)

Pada Gambar 2. 4 menunjukkan proses *Max Pooling* dengan ukuran kernel 2x2 dan *stride* 2. Pada *Max Pooling*, setiap area berukuran 2x2 diproses untuk mengambil nilai maksimum di dalamnya, sehingga mengurangi dimensi *feature map*.

Rumus *max pooling*:

$$Output = \frac{W - N}{S} + 1 \quad (2)$$

W = Input lebar atau tinggi citra

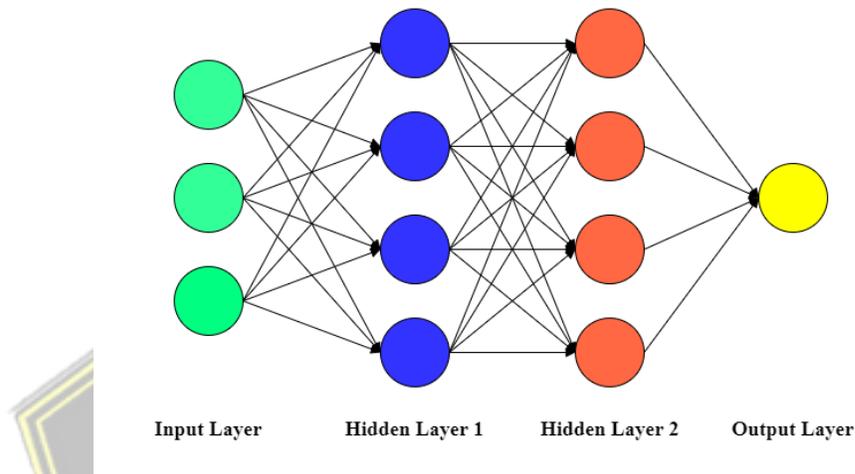
N = Ukuran filter untuk proses konvolusi misalnya (3 x 3)

S = Stride adalah langkah pergeseran filter saat menggeser filter di atas citra.

### 3. *Fully Connected Layer*

*Fully connected layer* adalah lapisan terakhir dalam arsitektur CNN yang menghubungkan semua neuron dari *layer* sebelumnya ke neuron saat ini. Pada *layer* ini semua informasi dari *layer* sebelumnya digabungkan untuk menghasilkan prediksi akhir. Tujuan *fully connected layer* digunakan untuk melakukan transformasi pada dimensi data yang telah diproses di *layer* sebelumnya, sehingga bisa diklasifikasikan secara linier. Fungsi aktifasi

sigmoid dapat digunakan untuk menghitung nilai probabilitas dari setiap kelas. fungsi sigmoid memberikan *output* antara 0 dan 1 yang dapat menunjukkan kemungkinan *input* termasuk kedalam kelas tertentu (Widi Wiguna dkk., 2023).



Gambar 2. 5 *Fully connected layer* (Dabbo dan Bisilisin, 2024)

Gambar 2. 5 adalah tampilan *fully connected layer* yang terdiri dari *input layer* untuk menerima data masukan, *hidden layer* untuk melakukan pemrosesan data dan *output layer* yang digunakan untuk menghasilkan prediksi.

Rumus *fully connected layer*:

$$Output = C \times K + C \quad (3)$$

C = Bobot dari koneksi antar neuron

K = Jumlah kernel

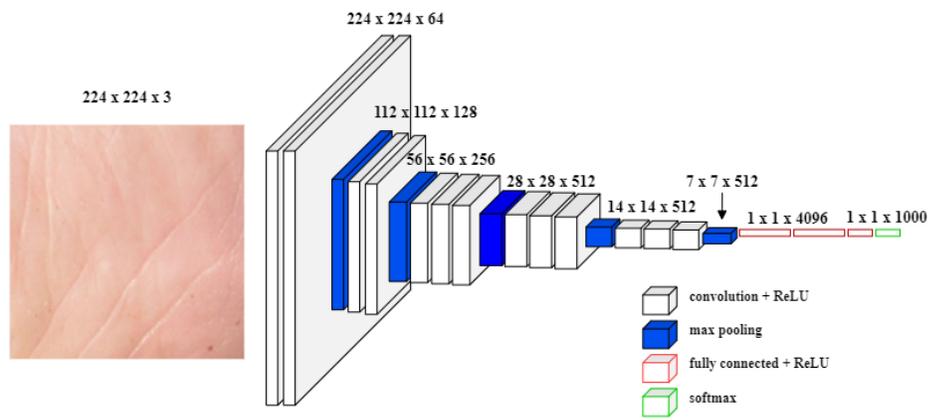
+C = Jumlah bias

Dalam beberapa tahun terakhir CNN mengalami banyak perkembangan seperti munculnya berbagai arsitektur jaringan seperti VGG16, VGG19, ResNet50, Inception V3 dan Xception. Arsitektur ini telah banyak digunakan dalam berbagai bidang seperti klasifikasi gambar hingga

pencitraan medis (Raharjo, 2022).

#### 2.2.4 Visual Geometry Group (VGG16)

VGG16 merupakan salah satu arsitektur *Convolutional Neural Network* (CNN) yang mampu mengekstrak fitur visual yang lebih kompleks. Arsitektur ini dikenal karena kesederhanaan modelnya yang mampu menghasilkan akurasi tinggi pada berbagai tugas pengenalan dan klasifikasi citra. VGG16 memiliki kernel berukuran kecil  $3 \times 3$  dengan *max pooling* untuk *feature extraction* dan *fully connected layer* untuk klasifikasi. VGG16 memiliki citra dengan dimensi  $224 \times 224$  piksel dan 3 channel R, G, B yang diolah menggunakan *convolution layer* dan *max pooling* (Ferani dkk., 2022).



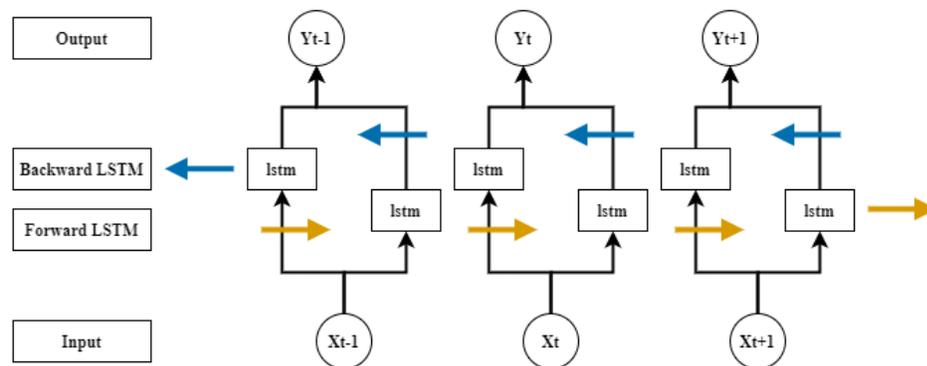
Gambar 2. 6 Arsitektur VGG16 (Paneru dkk., 2024)

Arsitektur VGG16 terdapat dua bagian yaitu *feature extraction* yang digunakan untuk mengekstrak fitur-fitur penting pada gambar seperti tepi, garis, warna, sudut, tekstur, bentuk dan objek yang lebih kompleks sedangkan *classification* untuk mengklasifikasi gambar berdasarkan fitur yang diekstrak oleh *feature extraction*. Dengan total 16 lapisan yang dapat disesuaikan, pada bagian *feature extraction* meliputi 13 lapisan *convolutional layer* dan bagian *classification* meliputi 3 lapisan *fully connected layer*. Selain itu terdapat 5

lapisan *max pooling* yang digunakan untuk mengurangi ukuran fitur (Noorizki dan Kusumawati, 2023).

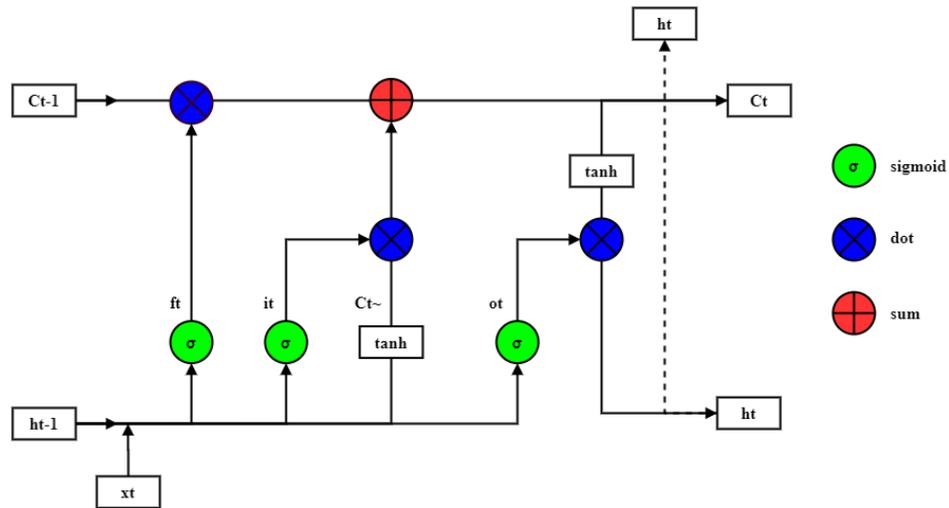
### 2.2.5 Bidirectional Long Short-Term Memory (BiLSTM)

*Bidirectional Long Short-Term Memory* (BiLSTM) merupakan pengembangan dari model *Long Short Term Memory* (LSTM). Model ini dirancang untuk memproses data urutan dengan lebih baik dengan memanfaatkan arsitektur dari model LSTM. Pada model LSTM data hanya diproses dalam satu arah yaitu arah maju, sedangkan model BiLSTM dapat memproses data dari dua arah yaitu arah maju dan mundur (Rolangon dkk., 2023).



Gambar 2. 7 Arsitektur BiLSTM (Ge dan Han, 2024)

Pada Gambar 2. 7 Arsitektur BiLSTM terdapat dua lapisan utama yaitu *Forward LSTM* dan *Backward LSTM*. *Forward LSTM* digunakan untuk memproses urutan data dari awal ke akhir, *Backward LSTM* digunakan untuk memproses urutan data dari akhir ke awal, hasil dari kedua lapisan ini akan digabung agar menghasilkan *output* yang dapat memahami konteks penuh dari data (Ge dan Han, 2024).



Gambar 2. 8 Arsitektur LSTM (Lan dkk., 2024)

Pada Gambar 2. 8 arsitektur LSTM memiliki *cell state* dan tiga gerbang utama *forget gate*, *input gate* dan *output gate*. *Forget gate* ( $f_t$ ) berfungsi untuk memutuskan informasi mana yang harus dibuang atau disimpan, *input gate* ( $i_t$ ) bertanggung jawab untuk menangkap informasi baru mana yang harus ditambahkan pada langkah waktu saat ini, *output gate* ( $o_t$ ) untuk mengontrol bagian mana dari status sel yang harus dikeluarkan sebagai status saat ini, *cell state* ( $C_{t-1}$ ) digunakan sebagai memori jangka panjang untuk menyimpan dan mengalirkan informasi penting yang dibutuhkan untuk menghasilkan *output* (Lan dkk., 2024).

1. *Forget gate* ( $f_t$ )

$$f_t = \sigma(W_f \cdot [h_{t-1}, x_t] + b_f) \quad (4)$$

$f_t$  = Nilai *output forget gate* .

$\sigma$  = adalah fungsi sigmoid yang menghasilkan nilai antara 0 sampai 1.

$W_f$  = matriks bobot untuk *forget gate* .

$h_{t-1}$  = adalah *hidden state* sebelumnya.

$x_t$  = adalah *input* saat ini.

$b_f$  = merupakan bias dari *forget gate* , dan  $\sigma$  adalah fungsi sigmoid yang menghasilkan nilai antara 0 sampai 1.

2. *Input gate* (it)

$$i_t = \sigma(W_i \cdot [h_{t-1}, x_t] + b_i) \quad (5)$$

$$\tilde{c}_t = \tanh(W_c \cdot [h_{t-1}, x_t] + b_c) \quad (6)$$

$i_t$  = adalah *output* dari fungsi sigmoid untuk menentukan besar informasi baru yang diterima dan menghasilkan nilai 0 sampai 1.

$\tilde{c}_t$  = nilai *cell state* baru, dihitung menggunakan fungsi tanh yang menghasilkan nilai antara -1 sampai 1.

3. *Cell state* (ct)

$$c_t = f_t \odot c_{t-1} + i_t \odot \tilde{c}_t \quad (7)$$

$c_t$  = merupakan *output* dari *cell state*.

$f_t \odot c_{t-1}$  = adalah informasi yang dipertahankan dari *cell state* sebelumnya.

$i_t \odot \tilde{c}_t$  = adalah informasi baru yang ditambahkan dari *input gate* .

$\odot$  = operasi perkalian elemen yang digunakan untuk memastikan hanya informasi relevan yang dipertahankan atau ditambahkan.

4. *Output gate* (ot)

$$o_t = \sigma(W_o \cdot [h_{t-1}, x_t] + b_o) \quad (8)$$

$o_t$  = *Output* dari *output gate* , menentukan informasi mana dari *cell state* yang akan diteruskan sebagai *hidden state*.

$W_o$  = matriks bobot untuk *output gate* .

$[h_{t-1}, x_t]$  = gabungan status tersembunyi sebelumnya dan *input* saat ini.

$b_o$  = bias untuk *output gate* .

$\sigma$  = adalah fungsi sigmoid yang menghasilkan nilai antara 0 sampai 1.

$$h_t = o_t \odot \tanh(c_t) \quad (9)$$

$h_t$  = output akhir

$\tanh(c_t)$  = aktivasi non-linier dari cell state untuk menghasilkan nilai dari -1 sampai 1

### 2.2.6 Bilingual Evaluation Understudy (BLEU)

Bilingual Evaluation Understudy (BLEU) adalah metrik evaluasi otomatis yang digunakan untuk mengukur teks yang dihasilkan oleh model dibandingkan dengan referensi. Metrik ini dapat mengevaluasi sistem penerjemah mesin di berbagai tugas *Natural Language Processing* (NLP) termasuk *image captioning* (Fudholi dan Nayoan, 2022).

Skor BLEU memiliki rentang antara 0 hingga 1, di mana skor 1 menunjukkan kecocokan sempurna dengan teks referensi, sedangkan skor 0 menandakan tidak adanya kemiripan sama sekali. BLEU menggunakan pendekatan berbasis n-gram, yaitu rangkaian kata yang tersusun secara berurutan. Dalam perhitungan BLEU, panjang maksimum n-grams yang digunakan adalah 4, karena nilai ini terbukti memiliki korelasi tinggi dengan penilaian manusia terhadap kualitas teks (Farhan dan Setiaji, 2023).

Skor BLEU dihitung dengan membandingkan jumlah n-gram yang sesuai antara teks hasil prediksi dan teks referensi. Selanjutnya, nilai tersebut dinormalisasi agar tetap berada dalam rentang 0 hingga 1, sehingga skor tetap terjaga dalam skala yang konsisten. Semakin tinggi nilai BLEU tersebut, maka semakin akurat model tersebut. Skor BLEU di atas 0.3 dapat mencerminkan terjemahan yang dapat dimengerti dan skor BLEU di atas 0.5 dapat mencerminkan terjemahan yang baik dan fasih. Jika semakin banyak n-gram

dalam prediksi yang cocok dengan referensi maka skor BLEU akan semakin tinggi. Perhitungan dalam BLEU didasarkan pada *precision* n-gram dengan mempertimbangkan panjang keluaran menggunakan *brevity penalty* (Luo dkk., 2022).

### 1. *Precision* n-gram ( $\mathcal{P}_n$ )

*Precision* dalam BLEU dihitung dengan membandingkan jumlah n-gram dalam prediksi yang cocok dengan referensi.

$$\mathcal{P}_n = \frac{\sum_{C \in \text{Candidates}} \sum_{n\text{-gram} \in C} \text{Count}_{clip}(n\text{-gram})}{\sum_{C' \in \text{Candidates}} \sum_{n\text{-gram} \in C'} \text{Count}(n\text{-gram})} \quad (10)$$

Di mana :

$\mathcal{P}_n$  = *Precision* n-gram, digunakan untuk mengukur seberapa akurat model menghasilkan n-gram yang sesuai dengan referensi. *Precision* dihitung sebagai rasio antara jumlah n-gram yang cocok dengan total n-gram yang dihasilkan.

$\text{Count}_{clip}(n\text{-gram})$  = Jumlah n-gram yang muncul, baik dalam hasil prediksi maupun dalam referensi, tetapi dibatasi oleh jumlah kemunculan dalam referensi.

$\text{Count}(n\text{-gram})$  = Total n-gram dalam teks yang dihasilkan model

C = Prediksi yang dihasilkan model dan dibandingkan dengan referensi

*Precision* ini adalah bagian dari perhitungan BLEU *Score*, yang menggabungkan *precision* untuk unigram hingga 4-gram serta menerapkan *brevity penalty* untuk menangani prediksi yang terlalu pendek.

### 2. *Brevity Penalty* (BP)

BLEU menggunakan *brevity penalty* untuk menghukum keluaran yang terlalu pendek dibandingkan referensi dan memastikan model menghasilkan teks yang cukup panjang.

$$BP = \begin{cases} 1, & \text{Jika } c > r \\ e^{(1-r/c)}, & \text{Jika } c \leq r \end{cases} \quad (11)$$

Di mana :

e = eurel konstanta matematika yang bernilai sekitar 2.71828.

c = Panjang teks yang dihasilkan model.

r = Panjang referensi.

Dalam rumus ini jika BP bernilai kurang dari 1 akan terjadi penalti terhadap keluaran yang lebih pendek dari referensi. Sebaliknya jika BP bernilai 1 maka tidak ada penalti.

### 3. Rumus BLEU

Skor BLEU dihitung dengan mengkombinasikan *precision* n-gram dan *brevity penalty*.

$$BLEU = BP \times \exp \left( \sum_{n=1}^N w_n \log P_n \right) \quad (12)$$

Di mana :

BP = *Brevity Penalty*.

exp = Eksponen, untuk memastikan bahwa perhitungan menggunakan log *precision* tetap dalam skala probabilitas yang *valid*. Memastikan nilai tetap dalam rentang 0 hingga 1.

n = order n-gram untuk menentukan tingkan n-gram yang digunakan.

$w_n$  = bobot untuk setiap n-gram, biasanya sama rata seperti 0.25 untuk BLEU-4.

log = Logaritma natural, untuk menghindari nilai *precision* mendekati 0, logaritma menjaga stabilitas perhitungan BLEU

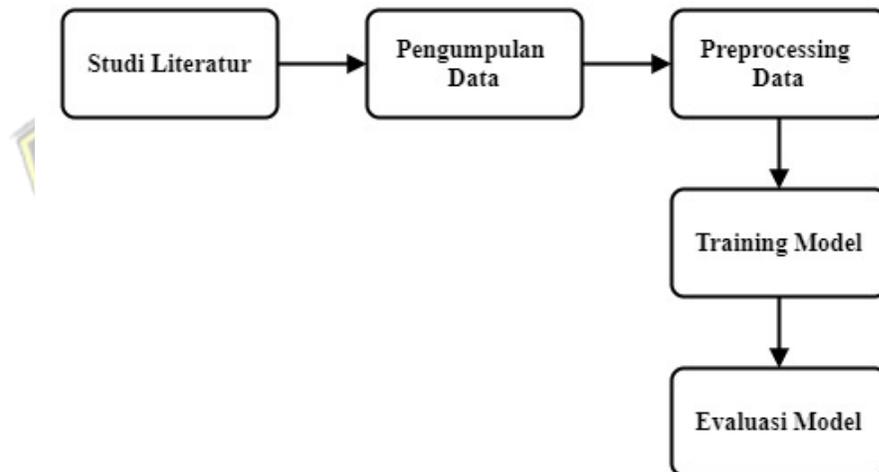
$P_n$  = *Precision* n-gram.

BLEU dihitung dengan menggabungkan *brevity penalty* (BP), n-gram *precision* ( $P_n$ ), bobot ( $w_n$ ), serta logaritma dan eksponen untuk menghasilkan skor BLEU dalam rentang 0-1.

## BAB III METODOLOGI PENELITIAN

### 3.1 Metode Penelitian

Dalam penelitian ini metode yang digunakan adalah VGG16 dan *Bidirectional Long Short Term Memory*, dimana dengan mengkombinasikan kedua metode ini akan menghasilkan sistem *image captioning* yang mampu mendeskripsikan gambar secara akurat. Adapun tahapan yang harus dilakukan dalam penelitian ini dapat dilihat pada Gambar 3. 1 :



Gambar 3. 1 Tahap Penelitian

Pada Gambar 3. 1 merupakan diagram alur Tahap Penelitian dalam pengembangan model *image captioning* berbasis VGG16 dan BiLSTM. Berikut adalah penjelasan setiap tahap yang terdapat pada diagram:

1. Studi Literatur

Tahap awal dalam penelitian ini adalah melakukan studi literatur untuk memahami konsep dan metode yang digunakan dalam *image captioning*, termasuk arsitektur VGG16 dan BiLSTM.

2. Pengumpulan Data

Data yang digunakan dalam penelitian ini dikumpulkan dari dataset gambar yang memiliki deskripsi teks (*captions*). Dataset ini kemudian digunakan untuk melatih dan menguji model.

### 3. *Preprocessing* Data

Data gambar dan teks diproses sebelum dimasukkan ke dalam model. Beberapa langkah *preprocessing* meliputi:

- a. Normalisasi gambar agar sesuai dengan input VGG16.
- b. *Case folding* untuk pemrosesan teks seperti normalisasi teks.
- c. Tokenisasi teks dan konversi kata ke dalam bentuk numerik.
- d. Padding untuk menyamakan panjang input teks.

### 4. Training Model

Pada tahap ini, model *image captioning* dilatih menggunakan kombinasi VGG16 sebagai ekstraksi fitur gambar dan BiLSTM sebagai pemroses teks. Model akan mempelajari hubungan antara fitur gambar dan teks deskriptif.

### 5. Evaluasi Model

Model yang telah dilatih dievaluasi menggunakan metrik seperti BLEU *score* untuk mengukur kualitas hasil *captioning* terhadap data uji.

#### 3.1.1 Studi Literatur

Dalam penelitian ini akan dilakukan tinjauan terhadap beberapa *e-book*, artikel, jurnal, dan skripsi terdahulu, serta mengunjungi berbagai situs website untuk mempelajari teori dan berlatih mengenai metode VGG16, BiLSTM dan Penerapan *TensorFlow Lite* untuk membuat sistem *Image Captioning*.

#### 3.1.2 Pengumpulan Data

Pada tahap proses pengumpulan data pada penelitian ini memiliki dua data yaitu gambar dan *caption* gambar, untuk data gambar diambil dari *platform* Kaggle, untuk data *caption* gambar dilakukan dengan cara diskusi langsung dengan ahli dokter dari Fakultas Kedokteran Universitas Islam Sultan Agung Semarang. Dengan acuan diskusi mengidentifikasi gambar luka *diabetic foot*

*ulcer* kemudian penulis mengumpulkan jawaban dari diskusi untuk memberikan *caption* pada gambar luka. Proses pengumpulan data digambarkan dengan floechart pada gambar 3. 2.



Gambar 3. 2 *Flowchart* alur pengumpulan data

Pada Gambar 3. 2 adalah *flowchart* proses diskusi dan pengumpulan data yang akan digunakan untuk dataset sistem *image captioning*. Dimulai dari mencari dataset gambar dari *platform* Kaggle, kemudian dilanjutkan diskusi dengan ahli dokter terkait informasi gambar luka (*caption*).

### 3.1.3 Preprocessing Data

#### 1. *Resize* dan normalisasi gambar

Gambar dari *platform* Kaggle di *resize* dengan target ukuran 224x224 piksel, ini diperlukan karena *input* model vgg16 adalah 244x244 piksel. Kemudian setelah di *resize* dikonversi ke *array* numerik dan dinormalisasikan ke rentang antara [0, 1] dengan melalui pembagian 255.

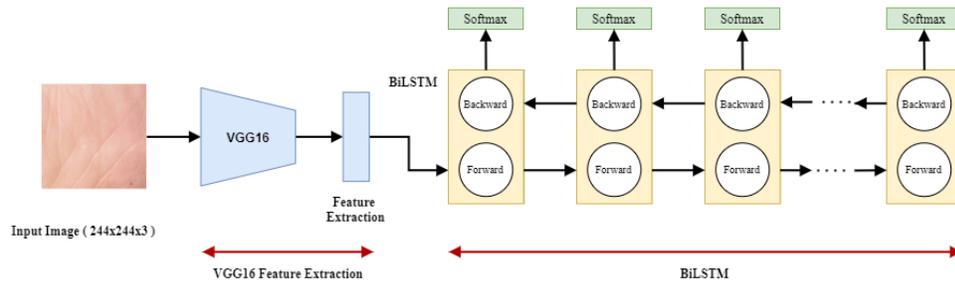
#### 2. *Case Folding*

*Case folding* adalah tahap dimana akan dilakukan untuk pemrosesan teks seperti normalisasi teks setiap *caption* pada dataset diubah yang awalnya huruf besar menjadi huruf kecil dan *caption* akan dibersihkan dari karakter *non-alfabet* misalnya titik dan koma. Kemudian dilakukan penambahan token *startseq* dan *endseq* yang ditambahkan pada awal *caption* dan akhir *caption* untuk menandai batas kalimat untuk membantu model dalam proses pelatihan dapat mengenali batas awal dan akhir kalimat.

#### 3. Tokenisasi dan *Padding*

*Caption* diubah menjadi urutan bilangan integer menggunakan Tokenizer dari Keras. Setiap urutan dilakukan *padding* berdasarkan Panjang maksimum *caption*.

### 3.1.4 Training Model



Gambar 3. 3 Training VGG16 dan BiLSTM

Pada Gambar 3. 3 adalah tampilan diagram pada proses pelatihan *image captioning*. Proses pelatihan model *image captioning* dilakukan dua tahap yaitu:

1. Ekstraksi Fitur

Pada tahap ini, gambar dimasukkan ke dalam model VGG16 yang berfungsi sebagai *encoder*. Jaringan konvolusional pada VGG16 mengekstraksi fitur visual dari gambar, menghasilkan vektor fitur yang merepresentasikan informasi visual. Vektor fitur inilah yang nantinya akan digunakan sebagai *input* untuk tahap pemrosesan berikutnya.

2. Pemrosesan Teks melalui BiLSTM

Pada tahap ini, *caption* yang sesuai dengan gambar diubah menjadi representasi numerik melalui proses *embedding*, di mana kata-kata diubah menjadi vektor. Selanjutnya, proses pemahaman urutan kata dilakukan oleh BiLSTM (*Bidirectional Long Short-Term Memory*), yang memproses informasi dari kedua arah kiri ke kanan dan kanan ke kiri untuk menangkap konteks yang lebih baik. *Output* dari BiLSTM kemudian digabungkan dengan vektor fitur gambar dari tahap sebelumnya dan dipetakan ke ruang probabilitas kata menggunakan fungsi *softmax*, sehingga dapat memprediksi kata berikutnya dalam urutan *caption*.

### 3.1.5 Evaluasi Model

Evaluasi model dilakukan untuk mengukur kinerja sistem *image captioning* dalam menghasilkan *caption* yang sesuai dengan gambar. Evaluasi ini

menggunakan metrik BLEU (*Bilingual Evaluation Understudy*) sebagai indikator utama untuk menilai kualitas *caption* yang dihasilkan. Dalam proses evaluasi, *caption* referensi dan *caption* prediksi ditokenisasi dipisahkan menjadi kata-kata menggunakan metode tokenisasi. Nilai BLEU dihitung menggunakan fungsi `corpus_bleu` dari *library* NLTK.

### 3.2 Analisa Kebutuhan

Pada tahap ini penulis menganalisa apa saja kebutuhan selama pembangunan sistem, seperti software, tools, bahasa pemrograman, dan *library* yang digunakan. Berikut adalah apa saja yang digunakan dalam pembangunan sistem :

#### 1. Bahasa Pemrograman

##### a. Python

Python merupakan bahasa pemrograman utama yang digunakan dalam pengembangan sistem ini. Python dipilih karena memiliki sintaks yang sederhana dan didukung oleh berbagai pustaka *Deep Learning* yang sangat powerful. Dengan fitur-fitur seperti kemudahan dalam pemrograman berorientasi objek, dukungan untuk pengolahan data, serta kemampuan untuk bekerja dengan data besar, Python adalah bahasa yang sangat ideal untuk pengembangan sistem berbasis *Deep Learning*.

#### 2. Perangkat Lunak (*Software*)

##### a. *Jupyter Notebook*

*Jupyter Notebook* digunakan sebagai lingkungan pengembangan untuk menulis dan menjalankan kode secara interaktif. *Platform* ini memudahkan penulis untuk menguji dan memodifikasi kode, serta mendokumentasikan eksperimen.

##### b. *Visula Studio Code*

*Visual Studio Code* dipilih sebagai text editor pada pengembangan aplikasi dalam penelitian ini, *Visual Studio Code* dipilih dikarenakan mendukung banyak bahasa pemrograman dan *framework*, *multi platform*, performa yang sangat cepat, mempunyai banyak *extensions* yang dapat mempermudah proses pengembangan website.

c. *Figma*

*Figma* adalah *tools* untuk memfasilitasi *designer* UI dan UX yang berbasis website yang digunakan untuk mendesain sebuah aplikasi. Dalam penelitian ini digunakan untuk membuat *design user interface* pada sistem *image captioning*.

3. *Library* dan *Framework*

a. *TensorFlow*

*TensorFlow* adalah *framework open-source* yang digunakan untuk membangun, melatih, dan menerapkan model *Deep Learning*. *TensorFlow* dipilih karena efisiensinya dalam menangani perhitungan matriks dan operasi tensor, serta kemampuan untuk melakukan pelatihan model menggunakan CPU/GPU.

b. *Keras*

*Keras* adalah antarmuka tinggi untuk *TensorFlow* yang memudahkan dalam membangun dan melatih model *neural network*. *Keras* menyediakan API yang sangat intuitif dan mudah dipahami, memungkinkan penulis untuk fokus pada pengembangan model tanpa terjebak dalam implementasi teknis rendah.

c. *NumPy*

*NumPy* digunakan untuk manipulasi *array* dan matriks numerik. Pustaka ini sangat penting untuk pengolahan data numerik, yang sering digunakan dalam model *Deep Learning* untuk pemrosesan *batch* data.

d. *Pandas*

*Pandas* digunakan untuk manipulasi dan analisis data. Dalam proyek ini, *Pandas* digunakan untuk mengelola dataset dalam format tabel, memudahkan proses pembersihan, transformasi, dan analisis data sebelum digunakan dalam model.

e. *Matplotlib*

*Matplotlib* digunakan untuk visualisasi hasil eksperimen dan model. Visualisasi ini sangat penting untuk menganalisis proses pelatihan, seperti menggambar grafik *loss* dan akurasi selama pelatihan model, serta untuk memverifikasi performa model.

f. NLTK (*Natural Language Toolkit*)

NLTK digunakan untuk memproses dan menganalisis data teks. Pustaka ini mendukung tugas-tugas seperti tokenisasi, penghilangan stopwords, dan perhitungan skor BLEU yang digunakan untuk mengevaluasi hasil penerjemahan dalam sistem.

g. *Scikit-learn*

*Scikit-learn* digunakan untuk beberapa tugas pembelajaran mesin klasik dan evaluasi model. Pustaka ini sangat berguna untuk membangun dan menilai model-model selain neural *network*, serta melakukan pre-processing data seperti normalisasi dan pemisahan data pelatihan dan pengujian.

h. TQDM

TQDM digunakan untuk menampilkan progress bar pada iterasi pelatihan model. Ini memudahkan pengembang untuk memonitor kemajuan pelatihan, terutama ketika bekerja dengan dataset yang besar.

i. *TensorBoard*

Digunakan untuk visualisasi pelatihan model dalam hal grafik akurasi, *loss*, dan metrik lainnya. *TensorBoard* membantu dalam pemantauan pelatihan model secara *realtime*.

j. *Pickle*

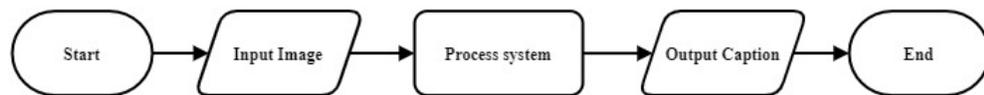
Pustaka *Pickle* digunakan untuk serialisasi objek python, memungkinkan penyimpanan model yang telah dilatih ke dalam file dan memuatnya kembali di lain waktu tanpa perlu melatih model dari awal.

k. *Textwrap*

Digunakan untuk membungkus atau merapikan teks agar mudah dibaca, terutama saat menampilkan hasil terjemahan atau teks yang dihasilkan model.

### 3.3 Penggunaan Sistem

Pada tahap ini dilakukan analisa untuk menentukan alur kerja penggunaan sistem *image captioning* yang akan dilakukan oleh *user* dalam bentuk *flowchart*. Alur kerja sistem dapat dilihat pada gambar 3.2



Gambar 3. 4 *Flowchart* alur kerja sistem

Gambar 3. 4 adalah *flowchart image captioning* yang menggambarkan proses sistem saat digunakan oleh *user*. Pada gambar 3. 3 memiliki beberapa tahapan, sebagai berikut :

1. *User* membuka aplikasi kemudian *user* akan melihat tampilan awal website *image captioning*.
2. *User* menekan tombol browser files untuk memasukkan gambar ke sistem atau bisa melakukan drag and drop file gambar.
3. Kemudian sistema akan memproses gambar tersebut dengan cara mengklasifikasi dengan model yang telah dibangun.
4. *User* akan mendapatkan *output* dari sistem berupa hasil *caption* dari gambar yang telah *user* masukkan ke sistem.

### 3.4 Perancangan *User Interface*

#### 3.4.1 Halaman Awal Sistem



Gambar 3. 5 Tampilan halaman awal sistem

Gambar 3. 5 merupakan tampilan rancangan antar muka pada bagian halaman awal sistem. Pada halaman ini, pengguna akan diminta untuk mengunggah gambar yang ingin dianalisis menggunakan sistem. Dengan cara menekan tombol *Browses files* untuk memilih gambar melalui *file explorer*. Setelah gambar diunggah, sistem akan memproses gambar tersebut dan menghasilkan *caption* otomatis menggunakan model VGG16 untuk ekstraksi fitur gambar dan BiLSTM untuk menghasilkan deskripsi dalam bentuk teks.

### 3.4.2 Tampilan Hasil Sistem



Gambar 3. 6 Tampilan hasil sistem

Gambar 3. 6 merupakan tampilan yang muncul setelah berhasil mengunggah gambar. Dalam tampilan ini pengguna dapat melihat informasi gambar yang telah diunggah dibagian kanan halaman. Pada bagian generate *caption* akan menampilkan hasil *caption* yang dihasilkan untuk gambar yang telah diunggah. Bagian *reference caption* menampilkan *caption* dari dataset yang digunakan untuk evaluasi. *Reference caption* ini akan dibandingkan dengan generate *caption* untuk mengukur akurasi *caption* yang dihasilkan oleh model. Pada bagian bawah sistem menampilkan skor BLEU yang digunakan untuk menilai kualitas *caption* yang dihasilkan. Skor BLEU mengukur kesamaan antara *caption* yang dihasilkan dan *caption* referensi dengan menggunakan perhitungan n-gram. Terdapat beberapa level BLEU (BLEU-1, BLEU-2, BLEU-3, dan BLEU-4) yang menunjukkan seberapa baik *caption* yang dihasilkan sesuai dengan referensi.

## BAB IV

### HASIL DAN ANALISIS PENELITIAN

#### 4.1 Hasil Pengumpulan Data



Gambar 4. 1 Dataset gambar *diabetic foot ulcer*

Pada gambar 4. 1 merupakan kumpulan dataset gambar luka diabetic foot ulcer. *Preprocessing* data menggunakan algoritma VGG16 melibatkan langkah-langkah umum seperti *resize* gambar dan normalisasi nilai piksel. *Resize* gambar dilakukan untuk menyesuaikan *input* pada model vgg16 dan normalisasi dilakukan untuk mengubah nilai piksel gambar menjadi skala tertentu, biasanya antara 0 dan 1 guna meningkatkan stabilitas dan kecepatan proses pelatihan.

Table 4. 1 Dataset *caption diabetic foot ulcer*

<i>Image</i>	<i>Caption</i>
 <p data-bbox="477 709 646 747">tipe 1 (1).jpg</p>	<p data-bbox="764 411 1382 663">Kondisi luka pada Stage satu. Luka hanya pada permukaan kulit pada perubahan warna dan nyeri pada kaki. Tidak ada tanda-tanda infeksi. Cukup jaga kebersihan dan kondisi kering pada luka. Lanjutkan perawatan mandiri secara rutin.</p>
 <p data-bbox="477 1071 646 1108">tipe 2 (2).jpg</p>	<p data-bbox="764 772 1382 972">Kondisi luka pada Stage dua. Luka melibatkan jaringan subkutan tanpa tanda infeksi berat. Disarankan konsultasi ke fasilitas kesehatan untuk pencegahan komplikasi.</p>
 <p data-bbox="477 1432 646 1470">tipe 3 (1).jpg</p>	<p data-bbox="764 1134 1382 1333">Kondisi luka pada Stage tiga. Luka dengan infeksi serius, melibatkan jaringan dalam seperti otot atau tulang. Perawatan medis mendesak diperlukan. Segera kunjungi fasilitas kesehatan.</p>
 <p data-bbox="477 1793 646 1831">tipe 4 (1).jpg</p>	<p data-bbox="764 1495 1382 1694">Kondisi luka pada Stage empat. Luka dengan gangrene luas atau nekrosis jaringan, melibatkan otot dan tulang. Kondisi serius memerlukan penanganan dokter spesialis sesegera mungkin.</p>

Pada Tabel 4. 1 merupakan kumpulan dataset *caption* luka diabetic foot ulcer yang telah dikumpulkan melalui diskusi dengan ahli dokter spesialis. Yang kemudian akan dilakukan *preprocessing* data dan dilatih menggunakan BiLSTM.

## 4.2 Hasil *Preprocessing* Data

### 4.2.1 *Image Preprocessing*

```
# Preprocess Images
from tensorflow.keras.preprocessing.image import load_img,
import numpy as np

def preprocess_image(image_path, target_size=(224, 224)):
    img = load_img(image_path, target_size=target_size)
    img = img_to_array(img) / 255.0

    return np.expand_dims(img, axis=0)
```

Gambar 4. 2 *Source code Image preprocessing*

Pada Gambar 4. 2 merupakan tampilan potongan kode yang diberikan adalah sebuah fungsi dalam Python yang dirancang untuk memproses gambar sebelum dimasukkan ke dalam model. Fungsi ini mengimpor fungsi yang diperlukan dari *TensorFlow*, seperti *load\_img* untuk memuat gambar dari jalur yang ditentukan dan *img\_to\_array* untuk mengonversi gambar menjadi *array* numerik. Fungsi *preprocess\_image* menerima jalur gambar dan ukuran target 224x224 piksel sebagai *input*. Fungsi ini memuat gambar, mengubah ukurannya sesuai ukuran target, kemudian mengonversinya menjadi *array* dan menormalkan nilai piksel dengan membaginya dengan 255 untuk menskalakan nilai antara 0 dan 1. Terakhir, fungsi ini menambahkan dimensi ekstra pada *array* gambar untuk menyesuaikan bentuk *input* yang diharapkan oleh model, memastikan gambar dapat diproses oleh model pembelajaran mendalam seperti jaringan saraf konvolusional (CNN).

## 4.2.2 Teks *Preprocessing*

```
import pandas as pd
import re

def text_preprocessing(data_caption):
    data_caption['caption'] = data_caption['caption'].apply(lambda x: x.lower())

    data_caption['caption'] = data_caption['caption'].apply(lambda x: re.sub(r"^[^a-z\s]", "", x))

    data_caption['caption'] = "startseq "+data_caption['caption']+" endseq"

    return data_caption
```

Gambar 4. 3 *Source code* untuk teks *preprocessing*

Pada Gambar 4. 3 merupakan tampilan *source code* untuk pemrosesan teks. Seperti *lowercasing*, penghapusan karakter *non-alfabeth* dan penambahan token. Kode pada gambar 4. 3 adalah sebuah fungsi *text\_preprocessing* yang digunakan untuk memproses teks dalam sebuah kolom pada dataset. Fungsi ini pertama-tama mengubah semua huruf kapital dalam kolom '*caption*' menjadi huruf kecil dengan menggunakan method *.lower()*. Selanjutnya, karakter *non-alfabet*, seperti angka dan simbol, dihapus dengan menggunakan *regular expression* (*re.sub*), yang hanya membiarkan karakter huruf dari 'a' hingga 'z' dan spasi. Terakhir, fungsi ini menambahkan token "*startseq*" di awal dan "*endseq*" di akhir setiap *caption*, yang biasanya digunakan dalam model *captioning* gambar untuk menandai awal dan akhir dari sebuah kalimat. Fungsi ini mengembalikan dataset yang sudah diproses.

### 1. *Lowercasing*

*Lowercasing* digunakan untuk mengubah huruf besar menjadi huruf kecil agar tidak ada perbedaan antara huruf besar dan huruf kecil saat pelatihan model.

Table 4. 2 Hasil *lowercasing*

Sebelum	Sesudah
"Luka basah dengan keluarnya cairan (eksudat). Cairan dapat berwarna kuning, kehijauan, atau bercampur darah. Luka basah dengan keluarnya cairan (eksudat). Cairan dapat berwarna kuning, kehijauan, atau bercampur darah. Kulit di sekitar luka tampak merah muda atau merah terang."	"luka basah dengan keluarnya cairan (eksudat). cairan dapat berwarna kuning, kehijauan, atau bercampur darah. luka basah dengan keluarnya cairan (eksudat). cairan dapat berwarna kuning, kehijauan, atau bercampur darah. kulit di sekitar luka tampak merah muda atau merah terang."

## 2. Penghapusan Karakter *Non-Alfabet*

Pada proses ini dilakukan penghapusan karakter selain huruf seperti titik dan koma.

Table 4. 3 Hasil penghapusan karakter *non-alfabet*

Sebelum	Sesudah
"luka basah dengan keluarnya cairan (eksudat). cairan dapat berwarna kuning, kehijauan, atau bercampur darah. luka basah dengan keluarnya cairan (eksudat). cairan dapat berwarna kuning, kehijauan, atau bercampur darah. kulit di sekitar luka tampak merah muda atau merah terang."	luka basah dengan keluarnya cairan eksudat cairan dapat berwarna kuning kehijauan atau bercampur darah luka basah dengan keluarnya cairan eksudat cairan dapat berwarna kuning kehijauan atau bercampur darah kulit di sekitar luka tampak merah muda atau merah terang

### 3. Penambahan Token

Pada proses ini dilakukan penambahan token “*startseq*” di awal dan “*endseq*” di akhir *caption* untuk membantu model dalam proses pelatihan dapat mengenali batas awal dan akhir kalimat.

Table 4. 4 Hasil penambahan token

Sebelum	Sesudah
luka basah dengan keluarnya cairan eksudat cairan dapat berwarna kuning kehijauan atau bercampur darah luka basah dengan keluarnya cairan eksudat cairan dapat berwarna kuning kehijauan atau bercampur darah kulit di sekitar luka tampak merah muda atau merah terang	<i>startseq</i> luka basah dengan keluarnya cairan eksudat cairan dapat berwarna kuning kehijauan atau bercampur darah luka basah dengan keluarnya cairan eksudat cairan dapat berwarna kuning kehijauan atau bercampur darah kulit di sekitar luka tampak merah muda atau merah terang <i>endseq</i>

### 4. Tokenisasi dan *Padding*

```
# Tokenizer and Padding
from tensorflow.keras.preprocessing.text import Tokenizer

tokenizer = Tokenizer()
tokenizer.fit_on_texts(captions)
vocab_size = len(tokenizer.word_index) + 1
max_length = max(len(caption.split()) for caption in captions)
```

Gambar 4. 4 *Source code* tokenisasi dan *padding*

Pada Gambar 4. 4 adalah tampilan *source code* untuk proses tokenisasi dan *padding*. Setelah melalui proses *lowercasing*, penghapusan karakter *non-alfabet* dan penambahan token Langkah selanjutnya yaitu tokenisasi. *caption-caption* tersebut dikonversi menjadi urutan bilangan, setiap kata diubah menjadi angka unik berdasarkan frekuensi kemunculan dalam seluruh

dataset. Pada proses *padding* panjang maksimum *caption* dihitung untuk memastikan semua urutan teks dipadding ke panjang yang sama sehingga dapat diproses oleh model secara konsisten.

```

Vocabulary size: 201
Word index: {'luka': 1, 'atau': 2, 'dan': 3, 'startseq': 4, 'endseq': 5, 'di': 6, 'yang': 7, 'dengan': 8, 'sekitar': 9, 'infeksi':
10, 'kulit': 11, 'kondisi': 12, 'jaringan': 13, 'sering': 14, 'segera': 15, 'disertai': 16, 'sangat': 17, 'hitam': 18, 'memerlukan':
19, 'kaki': 20, 'tampak': 21, 'merah': 22, 'pada': 23, 'bau': 24, 'terang': 25, 'basah': 26, 'cairan': 27, 'menyebar': 28, 'bengkok':
29, 'busuk': 30, 'terdapat': 31, 'gangren': 32, 'berwarna': 33, 'kering': 34, 'darah': 35, 'serius': 36, 'terjadi': 37, 'berat': 38,
'penanganan': 39, 'akut': 40, 'cepat': 41, 'kehitaman': 42, 'lunak': 43, 'bernanah': 44, 'aliran': 45, 'pucat': 46, 'lin': 47,
'keras': 48, 'untuk': 49, 'mencegah': 50, 'kebiruan': 51, 'buruk': 52, 'nekrotik': 53, 'gelap': 54, 'tidak': 55, 'mati': 56,
'karena': 57, 'tindakan': 58, 'kali': 59, 'dingin': 60, 'cokelat': 61, 'dalam': 62, 'tulang': 63, 'lebih': 64, 'seperti': 65,
'ulkus': 66, 'perawatan': 67, 'lutut': 68, 'nekrosis': 69, 'amputasi': 70, 'nanah': 71, 'diabetikum': 72, 'kemerahan': 73, 'akibat':
74, 'melibatkan': 75, 'sebagai': 76, 'ditandai': 77, 'bisa': 78, 'permukaan': 79, 'tepi': 80, 'teratur': 81, 'kadang': 82,
'berupa': 83, 'lokal': 84, 'selulitis': 85, 'ke': 86, 'osteomielitis': 87, 'keuangan': 88, 'peradangan': 89, 'tanda-tanda': 90,
'adalah': 91, 'gangrene': 92, 'ada': 93, 'otot': 94, 'adanya': 95, 'tempat': 96, 'keterlibatan': 97, 'umum': 98, 'jari': 99, 'satu':
100, 'tumor': 101, 'ditumpangkan': 102, 'eksudat': 103, 'medis': 104, 'mendesak': 105, 'diperlukan': 106, 'kunjungi': 107,
'fasilitas': 108, 'kesehatan': 109, 'diabetes': 110, 'hanya': 111, 'perubahan': 112, 'warna': 113, 'nyeri': 114, 'cukup': 115,
'jaga': 116, 'kebersihan': 117, 'lanjutkan': 118, 'mandiri': 119, 'secara': 120, 'rutin': 121, 'atas': 122, 'bawah': 123,
'kerusakan': 124, 'keluarnya': 125, 'dapat': 126, 'kuning': 127, 'kehijauan': 128, 'bercampur': 129, 'ulserasi': 130, 'menunjukkan':
131, 'yaitu': 132, 'parah': 133, 'neuropatik': 134, 'muda': 135, 'didefinisikan': 136, 'sendi': 137, 'pergelangan': 138, 'melalui':
139, 'penyebab': 140, 'termasuk': 141, 'rasa': 142, 'sakit': 143, 'menyiksa': 144, 'ekstrim': 145, 'area': 146, 'eritema': 147,
'munculnya': 148, 'biasanya': 149, 'plantar': 150, 'dorsum': 151, 'luas': 152, 'disebabkan': 153, 'oleh': 154, 'minim': 155, 'tanpa':
156, 'perlu': 157, 'penyebaran': 158, 'eksudasi': 159, 'proses': 160, 'inflamasi': 161, 'sedap': 162, 'purulen': 163, 'kekuningan':
164, 'tepi': 165, 'aktif': 166, 'signifikan': 167, 'sebelumnya': 168, 'penderita': 169, 'melitus': 170, 'mengalami': 171, 'gangguan':
172, 'penyembuhan': 173, 'sirkulasi': 174, 'mungkin': 175, 'memiliki': 176, 'faktor': 177, 'risiko': 178, 'neuropati': 179,
'sensorik': 180, 'iskemia': 181, 'kelainan': 182, 'bentuk': 183, 'kapalan': 184, 'kombinasi': 185, 'antara': 186, 'iskemik': 187,
'dangkal': 188, 'dasar': 189, 'abu-abu': 190, 'penyebabnya': 191, 'saraf': 192, 'sulit': 193, 'sembuh': 194, 'berisiko': 195,
'terhadap': 196, 'sedikit': 197, 'dokter': 198, 'spesialis': 199, 'sesegera': 200}
Maximum caption length: 58

```

Gambar 4. 5 Hasil *source code* tokenisasi dan *padding*

Pada Gambar 4. 5 adalah tampilan hasil dari tokenisasi dan *padding* pada teks untuk aplikasi *image captioning*. Daftar yang ditampilkan berisi kamus kata yang telah ditokenisasi, dengan setiap kata diberikan indeks numerik yang unik. Kamus ini berfungsi untuk memetakan kata-kata dalam *caption* gambar menjadi angka-angka yang dapat diproses oleh model pembelajaran mesin. Selain itu gambar ini juga menunjukkan panjang maksimal *caption* yang telah ditentukan yaitu 58. Proses *padding* di sini memastikan bahwa semua *caption* memiliki panjang yang konsisten, dengan menambahkan token khusus jika *caption* lebih pendek dari panjang maksimal. Tokenisasi dan *padding* merupakan langkah penting dalam mempersiapkan data teks agar siap digunakan dalam model *Deep Learning*, seperti BiLSTM yang digunakan dalam *image captioning*.

## 4.3 Hasil Modeling

### 4.3.1 Model VGG16

```

from tqdm import tqdm
from tensorflow.keras.applications import VGG16
from tensorflow.keras.models import Model
from tensorflow.keras.layers import GlobalAveragePooling2D
from tensorflow.keras.preprocessing.image import load_img, img_to_array
import os
import numpy as np

vgg_model = VGG16(weights='imagenet', include_top=False, input_shape=(224, 224, 3))
model_vgg16 = Model(inputs=vgg_model.input, outputs=GlobalAveragePooling2D()(vgg_model.output))

features = {}
for image in tqdm(data_caption['image'].unique().tolist()):
    img = load_img(os.path.join(data_image, image), target_size=(224, 224))
    img = img_to_array(img) / 255.0
    img = np.expand_dims(img, axis=0)
    features[image] = model_vgg16.predict(img, verbose=0)

vgg_model.summary()

```

100%|██████████| 1460/1460 [02:15<00:00, 10.76it/s]  
Model: "vgg16"

Gambar 4. 6 Source code model VGG16

Gambar 4. 6 adalah tampilan *source code* yang menunjukkan implementasi model VGG16 untuk ekstraksi fitur dari gambar. Pertama model VGG16 diimport dengan bobot yang telah dilatih sebelumnya pada dataset *ImageNet*, dan bagian atas (*top*) dari model ini dihilangkan menggunakan *code* `include_top = False`, sehingga hanya lapisan konvolusinya yang digunakan. Model ini juga disesuaikan untuk menerima *input* gambar berukuran 224x224 piksel dengan tiga saluran warna (RGB). Selanjutnya, sebuah objek model baru dibuat menggunakan *input* dari VGG16 dan *output* dari lapisan `GlobalAveragePooling2D`, yang mengubah fitur spasial menjadi satu vektor fitur untuk setiap gambar. Dalam *loop* setiap gambar dalam dataset diproses dengan memuat gambar, mengubahnya menjadi *array*, melakukan normalisasi dengan membaginya dengan 255, dan menambah dimensi agar sesuai dengan *input* model. Fitur hasil ekstraksi untuk setiap gambar disimpan dalam *dictionary features*. Setelah proses ekstraksi fitur selesai, `vgg_model.summary()` digunakan untuk menampilkan ringkasan struktur model.

### 4.3.2 Model *Image Captioning*

```

from tensorflow.keras.layers import Input, Dense, Embedding, Dropout, LSTM, Bidirectional, RepeatVector, concatenate, add
from tensorflow.keras.models import Model
from tensorflow.keras.optimizers import Adam

# Input
input1 = Input(shape=(model_vgg16.output_shape[-1],)) # Fitur gambar
input2 = Input(shape=(max_length,)) # Fitur teks (sekuensial)

# Proses fitur gambar
img_features = Dense(256, activation='relu')(input1)
img_features_expanded = RepeatVector(max_length)(img_features) # (None, max_Length, 256)

# Proses fitur teks
sentence_features = Embedding(vocab_size, 256, mask_zero=False)(input2)
sentence_features = Dropout(0.5)(sentence_features) # (None, max_Length, 256)

# Gabungkan fitur gambar & teks
merged = concatenate([img_features_expanded, sentence_features], axis=2) # (None, max_Length, 512)

# BiLSTM Processing
sentence_features = Bidirectional(LSTM(256, return_sequences=False))(merged) # (None, 512)
sentence_features = Dropout(0.5)(sentence_features)

# Refinement Layer untuk fitur gambar
img_features_resized = Dropout(0.5)(Dense(512, activation='relu')(img_features)) # (None, 512)

# Combine Image + Text Features
x = add([sentence_features, img_features_resized])
x = Dense(128, activation='relu')(x)
x = Dropout(0.5)(x)
output = Dense(vocab_size, activation='softmax')(x)

# Build Model
caption_model = Model(inputs=[input1, input2], outputs=output)
caption_model.compile(loss='categorical_crossentropy', optimizer=Adam(learning_rate=0.001), metrics=['accuracy'])

# Model Summary
caption_model.summary()

```

Gambar 4. 7 Source code modeling image captioning

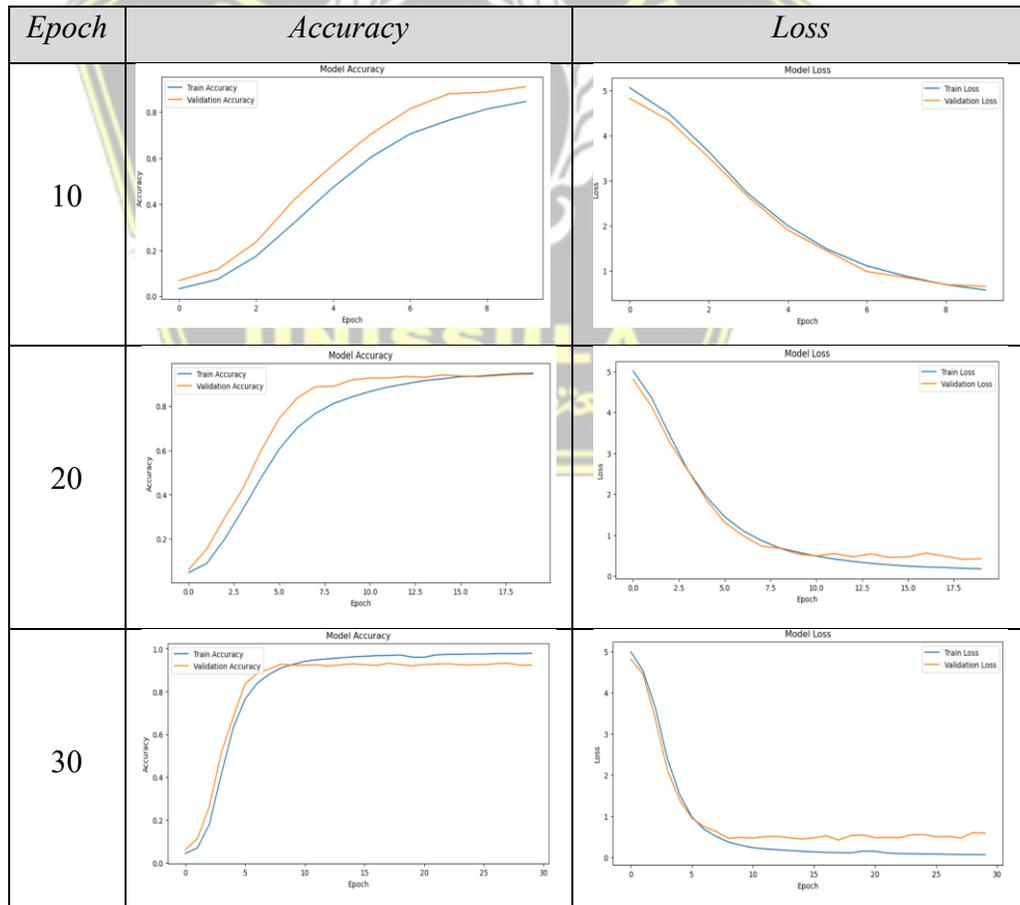
Pada Gambar 4. 7 adalah tampilan *source code* yang menunjukkan implementasi model *image captioning* yang menggabungkan fitur gambar dan teks menggunakan arsitektur BiLSTM. Model dimulai dengan dua *input*, satu untuk fitur gambar yang dihasilkan dari model VGG16 dan satu lagi untuk teks yang sudah diproses. Fitur gambar diproses dengan lapisan Dense dan Dropout, kemudian diperluas menggunakan RepeatVector agar bisa digabungkan dengan fitur teks. Fitur teks diproses melalui lapisan *Embedding* yang mengubah teks menjadi representasi vektor, kemudian dilakukan mask untuk mengabaikan *padding*. BiLSTM digunakan untuk menangkap dependensi dalam urutan teks, dengan `return_sequences = True` agar *outputnya* memiliki bentuk yang sesuai dengan panjang urutan. Refinement layer digunakan untuk fitur gambar agar memiliki dimensi yang cocok sebelum digabungkan dengan fitur teks. Kedua

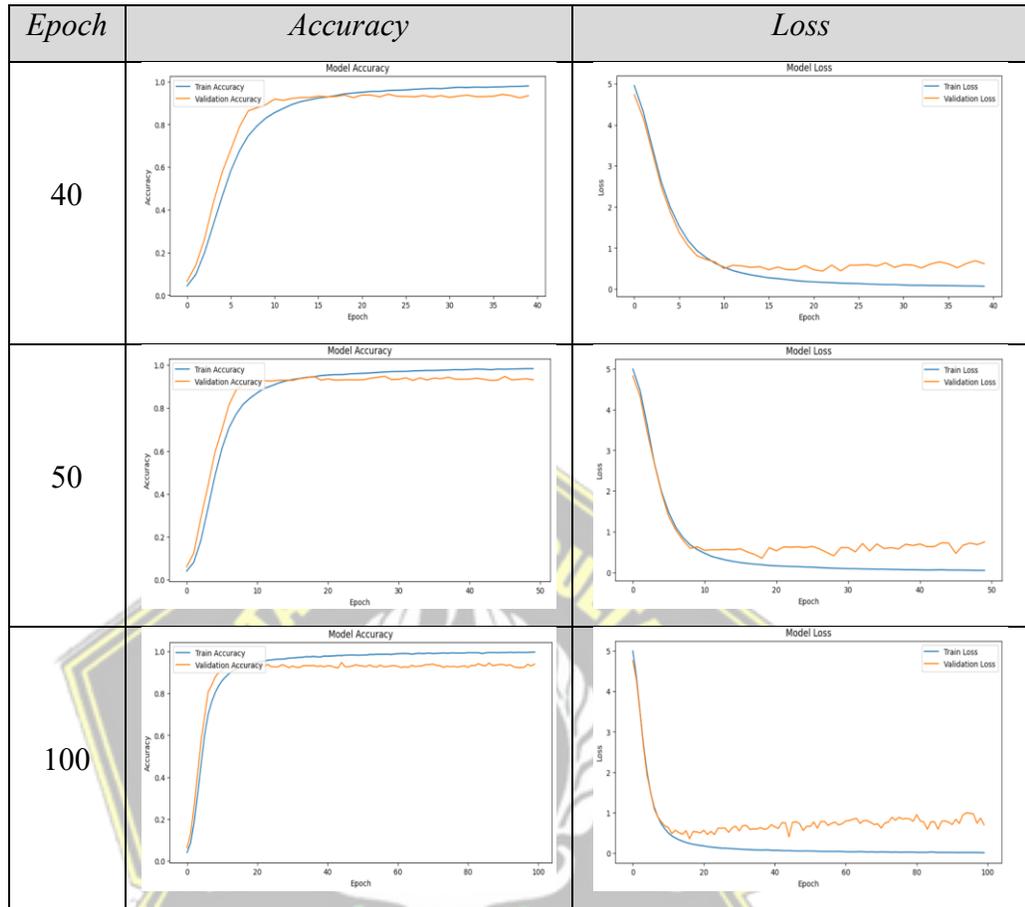
fitur gambar dan teks kemudian digabungkan dan diproses lebih lanjut dengan BiLSTM dan lapisan Dense untuk menghasilkan prediksi kata demi kata. Model ini dibangun menggunakan fungsi  $loss = categorical\_crossentropy$  dan optimisasi menggunakan Adam, dengan learning rate yang disesuaikan sebesar 0.001, yang merupakan pengaturan standar untuk mendapatkan pembaruan parameter yang stabil.

#### 4.4 Hasil Training

Setelah proses pemodelan selesai selanjutnya proses training model. Dimana pada hasil pelatihan ini dilakukan dengan mengkombinasi model VGG16 dengan BiLSTM berdasarkan *epoch*.

Table 4. 5 Plot accuracy dan loss





Tabel 4. 5 merupakan grafik dari *plot accuracy* dan *loss* pada *epoch* yang berbeda. Semakin naik grafik pada *plot accuracy* dan semakin turun grafik pada *loss*, serta grafik train dan validasi yang rapat dan saling mendekati, maka model akan semakin baik.

Table 4. 6 Hasil terbaik berdasarkan akurasi

<i>epoch</i>	<i>accuracy</i>	<i>val-accuracy</i>	<i>loss</i>	<i>val-loss</i>
10	0.8371	0.9094	0.6084	0.6529
20	0.9466	0.9439	0.1837	0.5285
30	0.9787	0.9244	0.0638	0.5864
40	0.9809	0.9346	0.0639	0.6152
50	0.9834	0.9318	0.0525	0.7483
100	0.9972	0.9401	0.0115	0.6999

Pada Tabel 4. 6 terlihat bahwa model menunjukkan perkembangan yang sangat baik selama pelatihan, dengan akurasi yang terus meningkat seiring bertambahnya jumlah *epoch*.

Pada *epoch* 10, akurasi validasi mencapai 0.9094, yang sudah tergolong tinggi. Namun, nilai *val-loss* sebesar 0.6529 lebih tinggi dibandingkan *loss* pada data pelatihan (0.6084). Hal ini menunjukkan bahwa model mulai belajar, tetapi masih ada ruang untuk perbaikan.

Pada *epoch* 20, akurasi pelatihan meningkat hingga 0.9466, sementara akurasi validasi mencapai 0.9439. Ini menunjukkan bahwa model dapat menggeneralisasi dengan baik terhadap data validasi. Selain itu, *loss* pada pelatihan turun secara signifikan menjadi 0.1837, meskipun *val-loss* masih berada di angka 0.5285.

Pada *epoch* 30, akurasi pelatihan terus meningkat hingga 0.9787, tetapi akurasi validasi justru sedikit menurun menjadi 0.9244. Meskipun demikian, selisihnya masih cukup kecil. Nilai *loss* pada pelatihan mengalami penurunan yang baik menjadi 0.0638, tetapi *val-loss* tetap cukup tinggi di angka 0.5864.

Pada *epoch* 40, akurasi pelatihan mencapai 0.9809, sedangkan akurasi validasi meningkat menjadi 0.9346. Namun, *val-loss* justru mengalami kenaikan menjadi 0.6152. Selisih antara akurasi pelatihan dan validasi yang mulai

membesar menunjukkan tanda awal *overfitting*, di mana model mulai lebih fokus pada data pelatihan daripada mampu menggeneralisasi ke data validasi.

Pada *epoch* 50, akurasi pelatihan meningkat lagi menjadi 0.9834, tetapi akurasi validasi mengalami sedikit penurunan menjadi 0.9318. Selain itu, *val-loss* naik lebih tinggi menjadi 0.7483. Peningkatan *val-loss* dan semakin besarnya selisih antara akurasi pelatihan dan validasi merupakan indikasi bahwa model semakin mengalami *overfitting*, sehingga performanya pada data yang belum pernah dilihat cenderung menurun.

Pada *epoch* 100, akurasi pelatihan mencapai 0.9972, menandakan bahwa model sangat baik dalam mengenali pola dalam data pelatihan. Namun, akurasi validasi hanya 0.9401, yang meskipun masih tinggi, tetap lebih rendah dibandingkan akurasi pelatihan. *Val-loss* sedikit menurun menjadi 0.6999, tetapi tidak cukup signifikan untuk menunjukkan perbaikan yang besar. Meskipun ada sedikit perbaikan dalam *loss* validasi, selisih yang cukup besar antara akurasi pelatihan dan validasi tetap menunjukkan adanya potensi *overfitting*, meskipun tidak terlalu parah.

Berdasarkan hasil tabel 4. 6 dapat disimpulkan bahwa, *epoch* 20 memberikan keseimbangan terbaik antara akurasi pelatihan 0.9466 dan akurasi validasi 0.9439, dengan *loss* yang relatif rendah baik pada pelatihan 0.1837 maupun validasi 0.5285. Model pada *epoch* ini menunjukkan performa yang optimal tanpa tanda-tanda *overfitting* yang signifikan. Oleh karena itu, *epoch* 20 dapat dipilih sebagai titik terbaik dalam pelatihan model ini.

#### 4.5 Hasil Evaluasi

Evaluasi performa model dilakukan dengan menggunakan metrik BLEU untuk mengukur tingkat kesamaan antara *caption* yang dihasilkan dengan *caption* referensi yang dikumpulkan. Tabel 4. 7 dibawah ini hasil perbandingan skor evaluasi metrik BLEU:

Table 4. 7 Perbandingan hasil skor matrik BLEU

<i>Epoch</i>	BLEU-1	BLEU-2	BLEU-3	BLEU-4
10	0.3286	0.2143	0.1722	0.1414
20	0.4307	0.3421	0.3125	0.2896
30	0.4067	0.3084	0.2751	0.2498
40	0.4044	0.3210	0.2937	0.2730
50	0.4189	0.3383	0.3111	0.2908
100	0.4511	0.3716	0.3455	0.3254

Pada Tabel 4. 7 terlihat hasil evaluasi model berdasarkan skor matriks BLEU pada berbagai *epoch*. Skor BLEU digunakan untuk mengukur kualitas keluaran model dengan membandingkan kesamaan antara hasil prediksi dan referensi yang benar. Semakin tinggi nilai BLEU, semakin baik performa model dalam menghasilkan teks yang sesuai dengan referensi.

Pada *epoch* 10, skor BLEU-1 sebesar 0.3286, BLEU-2 sebesar 0.2143, BLEU-3 sebesar 0.1722, dan BLEU-4 sebesar 0.1414. Skor ini menunjukkan bahwa model mulai belajar, tetapi kualitas hasil prediksi masih tergolong rendah.

Pada *epoch* 20, skor BLEU meningkat signifikan, dengan BLEU-1 sebesar 0.4307, BLEU-2 sebesar 0.3421, BLEU-3 sebesar 0.3125, dan BLEU-4 sebesar 0.2896. Peningkatan ini menunjukkan bahwa model semakin baik dalam memahami pola dalam data pelatihan dan menghasilkan prediksi yang lebih mendekati referensi.

Pada *epoch* 30, skor BLEU sedikit menurun dibandingkan *epoch* 20, dengan BLEU-1 sebesar 0.4067, BLEU-2 sebesar 0.3084, BLEU-3 sebesar 0.2751, dan BLEU-4 sebesar 0.2498. Ini menunjukkan bahwa model tetap mampu menghasilkan keluaran yang relatif baik, meskipun lebih rendah dibandingkan *epoch* 20.

Pada *epoch* 40, skor BLEU mengalami sedikit peningkatan dibandingkan *epoch* 30, terutama pada BLEU-2 hingga BLEU-4. Skor BLEU-1 berada di

0.4044, BLEU-2 meningkat menjadi 0.3210, BLEU-3 menjadi 0.2937, dan BLEU-4 menjadi 0.2730. Ini menunjukkan bahwa model tetap mampu menghasilkan keluaran yang relatif baik, meskipun masih sedikit lebih rendah dibandingkan *epoch* 20.

Pada *epoch* 50, skor BLEU kembali naik dengan BLEU-1 sebesar 0.4189, BLEU-2 sebesar 0.3383, BLEU-3 sebesar 0.3111, dan BLEU-4 sebesar 0.2908. Ini menunjukkan bahwa model tetap mampu menghasilkan keluaran yang relatif baik, meskipun masih sedikit lebih rendah dibandingkan *epoch* 20.

Pada *epoch* ke-100, skor BLEU mencapai nilai tertinggi dalam seluruh percobaan, dengan BLEU-1 sebesar 0.4511, BLEU-2 sebesar 0.3716, BLEU-3 sebesar 0.3455, dan BLEU-4 sebesar 0.3254. Ini menunjukkan bahwa model memiliki pemahaman terbaik dalam menghasilkan keluaran yang sesuai dengan referensi.

Berdasarkan hasil tabel 4. 7 dapat disimpulkan bahwa *epoch* ke-100 memberikan skor BLEU tertinggi di semua level dari BLEU-1 hingga BLEU-4, ini menandakan bahwa model menghasilkan prediksi yang paling akurat dengan referensi. Oleh karena itu, model pada *epoch* ke-100 dapat dipilih sebagai yang terbaik berdasarkan evaluasi BLEU.

#### 4.6 Hasil Implementasi

Setelah tahap pemodelan selesai, selanjutnya implementasi sistem *image captioning* ke dalam *platform* website dengan menggunakan framework streamlit. Berikut ini merupakan hasil implementasi sistem :

## Image Captioning with VGG16 and BiLSTM

Choose an image...

 Drag and drop file here  
Limit 200MB per file • JPG, JPEG, PNG

Gambar 4. 8 Halaman utama

Pada Gambar 4. 8 merupakan tampilan halaman utama yang akan ditampilkan pengguna saat pertama kali menjalankan sistem. Pada halaman ini terdapat judul utama "*Image Captioning Diabetic Foot Ulcer Menggunakan VGG16 dan BiLSTM*" yang menjelaskan fungsi utama aplikasi, yaitu menghasilkan deskripsi gambar menggunakan kombinasi model VGG16 dan BiLSTM. Pada halaman ini juga terdapat tombol "Browser files", tombol ini digunakan untuk mengunggah gambar dengan format JPG, JPEG dan PNG dengan batas ukuran file 200MB.

## Image Captioning with VGG16 and BiLSTM

Choose an image...

Drag and drop file here  
Limit 200MB per file • JPG, JPEG, PNG

Browse files

1460.jpg 16.9KB



Uploaded Image.

Generated Caption:

luka tampak basah jaringan hitam lunak disertai bau busuk terdapat cairan bernanah kondisi akut dengan infeksi berat yang menyebar cepat sangat memerlukan penanganan segera kulit di sekitar luka merah terang atau kehitaman sering bengkak

Reference Caption:

Luka disertai nanah, tepian luka tidak teratur, dan kadang disertai bau busuk. Sering terdapat bengkak dan kemerahan di sekitar luka. Infeksi bisa berupa lokal (selulitis) atau menyebar ke jaringan yang lebih dalam seperti tulang (osteomielitis). Kulit di sekitar luka tampak merah terang atau keunguan akibat peradangan.

	BLUE-1	BLUE-2	BLUE-3	BLUE-4
0	0.3100	0.1990	0.1224	0.0611

Gambar 4. 9 Hasil *image captioning*

Pada Gambar 4. 9 merupakan hasil tampilan setelah pengguna mengunggah gambar ke dalam sistem. Pada halaman ini sistem menampilkan beberapa informasi mulai dari gambar yang telah diunggah pengguna, generate *caption* yang merupakan hasil dari gambar yang diunggah, reference *caption* diambil dari dataset dan menampilkan skor BELU sebagai hasil evaluasi kualitas generate *caption*.

## BAB V

### KESIMPULAN DAN SARAN

#### 5.1 Kesimpulan

Berdasarkan hasil penelitian ini berhasil mengembangkan sistem *image captioning* untuk luka *Diabetic Foot Ulcer* (DFU) menggunakan kombinasi model VGG16 sebagai *encoder* dan BiLSTM sebagai *decoder*. Dataset gambar diperoleh dari *platform* Kaggle, sementara dataset *caption* disusun berdasarkan diskusi dengan ahli medis. Hasil pelatihan menunjukkan bahwa model mencapai keseimbangan terbaik pada epoch 20 dengan akurasi sebesar 0.9466 dan loss yang relatif rendah 0.1837. Evaluasi model menggunakan metrik BLEU menunjukkan bahwa performa terbaik diperoleh pada epoch 100 dengan skor BLEU-1 sebesar 0.4511, BLEU-2 sebesar 0.3716, BLEU-3 sebesar 0.3455, dan BLEU-4 sebesar 0.3254. Skor tersebut mengindikasikan bahwa model mampu menghasilkan *caption* yang lebih mendekati struktur dan makna dari *caption* referensi. Diharapkan, sistem ini memiliki potensi untuk membantu pasien dan keluarga dalam memahami kondisi luka DFU secara lebih objektif, sehingga dapat meningkatkan kualitas perawatan dan mengurangi risiko komplikasi lebih lanjut.

#### 5.2 Saran

Berdasarkan penelitian yang sudah dilakukan, saran untuk sistem *image captioning* pada luka *diabetic foot ulcer* (DFU) menggunakan VGG16 dan BiLSTM. Diharapkan dapat mengumpulkan dataset gambar luka DFU lebih banyak dan beragam, serta melibatkan banyak ahli medis dalam pembuatan *caption*. Guna meningkatkan akurasi model. Kemudian mengeksplorasi arsitektur model *Deep Learning* lain yang lebih canggih, seperti *Transformer* atau model *attention-based*, dapat menghasilkan *caption* yang lebih deskriptif dan informatif.

## DAFTAR PUSTAKA

- Agustari, F., Novitasari, D., dan Sembayang, S. M. (2022). Pencegahan Komplikasi Kaki Diabetik Melalui Edukasi Perawatan Kaki Pada Penderita Diabetes Melitus Tipe 2. *Jurnal Pengabdian Kepada Masyarakat (JPKM) - Aphelion*, 4(Desember), 603–608. <http://jurnal.globalhealthsciencegroup.com/index.php/JPM>
- Agustina, R., Magdalena, R., dan Pratiwi, N. K. C. (2022). Klasifikasi Kanker Kulit menggunakan Metode Convolutional Neural Network dengan Arsitektur VGG-16. *ELKOMIKA: Jurnal Teknik Energi Elektrik, Teknik Telekomunikasi, & Teknik Elektronika*, 10(2), 446. <https://doi.org/10.26760/elkomika.v10i2.446>
- Agustiono, W., Informasi, S., Madura, U. T., Inda, P. T., Bangkalan, K., Learning, D., Penerjemah, M., Network, C. N., dan Qur, A. (2024). *Model Penerjemah Bahasa Isyarat Huruf Hijaiyah*. 8(6), 12335–12342.
- Alghifari, D. R., Edi, M., dan Firmansyah, L. (2022). Implementasi Bidirectional LSTM untuk Analisis Sentimen Terhadap Layanan Grab Indonesia. *Jurnal Manajemen Informatika (JAMIKA)*, 12(2), 89–99. <https://doi.org/10.34010/jamika.v12i2.7764>
- Dabbo, P., dan Bisilisin, F. Y. (2024). Klasifikasi Motif Kain Tenun Sabu Rajjua Menggunakan Convolutional Neural Network (CNN) Berbasis Citra. *KETIK: Jurnal Informatika*, 1(06), 11–18.
- Farhan, N. M., dan Setiaji, B. (2023). Indonesian Journal of Computer Science. *Indonesian Journal of Computer Science*, 12(2), 284–301. <http://ijcs.stmikindonesia.ac.id/ijcs/index.php/ijcs/article/view/3135>
- Ferani, C., #1, K., Ratri, R., Wardani, S., dan Kom, M. T. (2022). Penerapan VGG16 untuk Klasifikasi Pneumonia pada Citra X-ray. *Institut Teknologi Harapan Bangsa*.
- Fudholi, D. H., dan Nayoan, R. A. N. (2022). The Role of Transformer-based *Image Captioning* for Indoor Environment Visual Understanding. *International Journal*

- of Computing and Digital Systems*, 12(1), 479–488.  
<https://doi.org/10.12785/ijcnds/120138>
- Ge, K., dan Han, Q. B. (2024). Hidden code vulnerability detection: A study of the Graph-BiLSTM algorithm. *Information and Software Technology*, 175(March 2024). <https://doi.org/10.1016/j.infsof.2024.107544>
- Lan, W., Tao, Y., Liang, B., Zhu, R., Wei, Y., dan Shen, B. (2024). Shear wave velocity prediction based on bayesian-optimized multi-head attention mechanism and CNN-BiLSTM. *Computers and Geosciences*, 195(November 2024), 105787. <https://doi.org/10.1016/j.cageo.2024.105787>
- Luo, G., Cheng, L., Jing, C., Zhao, C., dan Song, G. (2022). A thorough review of models, evaluation metrics, and datasets on *image captioning*. *IET Image Processing*, 16(2), 311–332. <https://doi.org/10.1049/ipr2.12367>
- Noorizki, A. Z., dan Kusumawati, W. I. (2023). Perbandingan Performa Algoritma VGG16 Dan VGG19 Melalui Metode CNN Untuk Klasifikasi Varietas Beras. *Journal of Computer, Electronic, and Telecommunication*, 4(2). <https://doi.org/10.52435/complete.v4i2.387>
- Nugroho, K. S., Akbar, I., Suksmawati, A. N., dan Istiadi. (2023). *Deteksi Depresi dan Kecemasan Pengguna Twitter Menggunakan Bidirectional LSTM*. *Ciastech*, 287–296. <http://arxiv.org/abs/2301.04521>
- Paneru, B., Paneru, B., dan Poudyal, K. N. (2024). Advancing human-computer interaction: AI-driven translation of American Sign Language to Nepali using convolutional neural networks and text-to-speech conversion application. *Systems and Soft Computing*, 6(October), 200165. <https://doi.org/10.1016/j.sasc.2024.200165>
- Pratiwi, V. R., dan Pardede, J. (2022). *Image Captioning Menggunakan Metode Inception-V3 dan Transformer*. *Prosiding Diseminasi FTI*, 1–14.
- Raharjo, B. (2022). *Deep Learning dengan Python*.
- Rif'at, I. D., Hasneli N, Y., dan Indriati, G. (2023). Gambaran Komplikasi Diabetes Melitus Pada Penderita Diabetes Melitus. *Jurnal Keperawatan Profesional*, 11(1),

52–69. <https://doi.org/10.33650/jkp.v11i1.5540>

- Rolangon, A., Weku, A., dan Sandag, G. A. (2023). Perbandingan Algoritma LSTM Untuk Analisis Sentimen Pengguna Twitter Terhadap Layanan Rumah Sakit Saat Pandemi Covid-19. *TeIka*, 13(01), 31–40. <https://doi.org/10.36342/teika.v13i01.3063>
- Said, M., Haskas, Y., dan Restika, I. (2024). Hubungan Perilaku Pencarian Pengobatan dan Perawatan Kaki Terhadap Resiko Ulkus Kaki Diabetes. *JIMPK : Jurnal Ilmiah Mahasiswa & Penelitian Keperawatan*, 4(1), 3.
- Satria, M. R., dan Pardede, J. (2022). *Image Captioning* Menggunakan Metode ResNet50 Dan Long Short-Term Memory. *Jurnal Tera*, 2(2), 84–94. <http://jurnal.undira.ac.id/index.php/jurnaltera/>
- Satwikayana, S., Adi Wibowo, S., dan Vendyansyah, N. (2021). Sistem Presensi Mahasiswa Otomatis Pada Zoom Meeting Menggunakan Face Recognition Dengan Metode Convolutional Neural Network Berbasis Web. *JATI (Jurnal Mahasiswa Teknik Informatika)*, 5(2), 785–793. <https://doi.org/10.36040/jati.v5i2.3762>
- Supirman, S., Lubis, C., Yulianto, D., dan Perdana, N. J. (2023). Klasifikasi Penyakit Kulit Menggunakan Convolutional Neural Network (Cnn) Dengan Arsitektur Vgg16. *Simtek : jurnal sistem informasi dan teknik komputer*, 8(1), 135–140. <https://doi.org/10.51876/simtek.v8i1.217>
- Taufiqur, A., dan Hatta, D. (2022). *Image Captioning* For Mobile Application. *Automata*, 3(2), 1–8.
- Vitniawati, V., Rahayu, S. M., dan Tambunan, I. (2023). Analysis of Risk Factors of Diabetic Foot Ulcer in Diabetes Mellitus Patients. *Jurnal Ilmu Keperawatan dan Kebidanan*, 14(2), 458–463. <https://doi.org/10.26751/jikk.v14i2.2051>
- Wang, X., Yuan, C.-X., Xu, B., dan Yu, Z. (2022). Diabetic foot ulcers: Classification, risk factors and management. *World Journal of Diabetes*, 13(12), 1049–1065. <https://doi.org/10.4239/wjd.v13.i12.1049>
- Widi Wiguna, C., Dedy Irawan, J., dan Orisa, M. (2023). Penerapan Metode

Convolutional Neural Network Pada Aplikasi Deteksi Wajah Buronan Berbasis Web. *JATI (Jurnal Mahasiswa Teknik Informatika)*, 6(2), 1051–1058.  
<https://doi.org/10.36040/jati.v6i2.5438>

