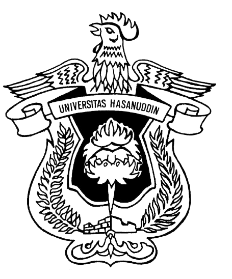
**PENERAPAN CONVOLUTIONAL NEURAL NETWORK DALAM KLASIFIKASI BERDASARKAN TINGKAT KONTAMINASI AFLATOKSIN PADA SAMPEL JAGUNG**

****

**INNAR FAIZA SYAHRANI**

**D121211021**

**PROGRAM STUDI TEKNIK INFORMATIKA**

**FAKULTAS TEKNIK**

**UNIVERSITAS HASANUDDIN**

**MAKASSAR**

**2025**

**PENERAPAN CONVOLUTIONAL NEURAL NETWORK DALAM KLASIFIKASI BERDASARKAN TINGKAT KONTAMINASI AFLATOKSIN PADA SAMPEL JAGUNG**

**INNAR FAIZA SYAHRANI**

**D121211021**



**PROGRAM STUDI TEKNIK INFORMATIKA**

**FAKULTAS TEKNIK**

**UNIVERSITAS HASANUDDIN**

**MAKASSAR**

**2025**

**PENERAPAN CONVOLUTIONAL NEURAL NETWORK DALAM KLASIFIKASI BERDASARKAN TINGKAT KONTAMINASI AFLATOKSIN PADA SAMPEL JAGUNG**

INNAR FAIZA SYAHRANI

D121211021

Skripsi

sebagai salah satu syarat untuk mencapai gelar sarjana

Program Studi Teknik Informatika

pada

**PROGRAM STUDI TEKNIK INFORMATIKA**

**DEPARTEMEN TEKNIK INFORMATIKA**

**FAKULTAS TEKNIK**

**UNIVERSITAS HASANUDDIN**

**MAKASSAR**

**2025**

**SKRIPSI**

**PENERAPAN CONVOLUTIONAL NEURAL NETWORK DALAM KLASIFIKASI BERDASARKAN TINGKAT KONTAMINASI AFLATOKSIN PADA SAMPEL JAGUNG**

**INNAR FAIZA SYAHRANI**

**D121211021**

Skripsi,

telah dipertahankan di depan Panitia Ujian Sarjana Teknik Informatika pada (tanggal bulan tahun) dan dinyatakan telah memenuhi syarat kelulusan pada

Program Studi Teknik Informatika

Departemen Teknik Informatika

Fakultas Teknik

Universitas Hasanuddin

Makassar

|  |  |
| --- | --- |
| Mengesahkan: | Mengetahui: |
| Pembimbing Tugas Akhir, | Ketua Program Studi, |
|  |  |
| Prof. Dr. Ir. Amil Ahmad Ilham, S.T., M.IT.  NIP. 19731010 199802 1 001 | Prof. Dr. Ir. Indrabayu, S.T., M.T., M.Bus.Sys., IPM., ASEAN Eng.  NIP. 197507162002121004 |

**PERNYATAAN KEASLIAN SKRIPSI**

**DAN PELIMPAHAN HAK CIPTA**

Dengan ini saya menyatakan bahwa, skripsi berjudul “Penerapan *Convolutional Neural Network*  dalam Klasifikasi Berdasarkan Tingkat Kontaminasi Aflatoksin pada Sampel Jagung*l*” adalah benar karya saya dengan arahan dari pembimbing Prof. Dr. Ir. Amil Ahmad Ilham, S.T., M.IT... Karya ilmiah ini belum diajukan dan tidak sedang diajukan dalam bentuk apa pun kepada perguruan tinggi mana pun. Sumber informasi yang berasal atau dikutip dari karya yang diterbitkan maupun tidak diterbitkan dari penulis lain telah disebutkan dalam teks dan dicantumkan dalam Daftar Pustaka skripsi ini. Apabila di kemudian hari terbukti atau dapat dibuktikan bahwa sebagian atau keseluruhan skripsi ini adalah karya orang lain, maka saya bersedia menerima sanksi atas perbuatan tersebut berdasarkan aturan yang berlaku.

Dengan ini saya melimpahkan hak cipta (hak ekonomis) dari karya tulis saya berupa skripsi ini kepada Universitas Hasanuddin.

Makassar, 17-06-2025

*INNAR FAIZA SYAHRANI*

NIM D121211021

# ABSTRAK

INNAR FAIZA SYAHRANI. **Penerapan *Convolutional Neural Network* dalam Klasifikasi Berdasarkan Tingkat Kontaminasi Aflatoksin pada Sampel Jagung** (dibimbing oleh Amil Ahmad Ilham).

**Latar belakang**. Mekanisme *attention* dalam *Deep Learning* (DL) dapat meningkatkan fokus model pada bagian penting dari teks input. Meskipun keunggulannya terbukti dalam klasifikasi teks, dampak variasi jumlah *head* *attention* terhadap kinerja dan efisiensi model masih perlu diteliti. **Tujuan**. Penelitian ini bertujuan menganalisis pengaruh jumlah *head* pada *layer attention* model DL untuk klasifikasi teks, serta membandingkannya dengan model tanpa *attention*. **Metode**. Model *Bidirectional Long Short*-*Term Memory* (BiLSTM) dengan *multi*-*head* *attention* diimplementasikan menggunakan data pertanyaan Text REtrieval Conference (TREC) dari University of Southern California (USC) dengan jumlah 5,942 data. Data melalui tahap *pre*-*processing* dengan menghapus kategori ABBR (*abbreviation*) karena jumlahnya tidak seimbang dengan kategori lainnya. Eksperimen meliputi skenario *grid search* untuk optimasi *hyperparameter* dan skenario pengujian variasi jumlah *head attention* (0 hingga 150). Metrik akurasi, *confusion matrix*, dan F1-*score* digunakan untuk evaluasi. **Hasil**. Model tanpa *attention* dengan akurasi 93.27% mengalahkan beberapa variasi model dengan *attention*. Namun, akurasi tertinggi diperoleh oleh model dengan 50 *head attention* dengan akurasi 94.09%. Seluruh model dengan *attention* memiliki waktu *training* yang lebih efisien dibandingkan model tanpa *attention*. **Kesimpulan**. Penambahan *head* *attention* tidak selalu memberikan peningkatan akurasi yang signifikan. Namun, dapat meningkatkan efisiensi waktu pelatihan.

Kata kunci: klasifikasi teks; *deep learning*; *attention*; *multi*-*head* *attention*; BiLSTM

# ABSTRACT

INNAR FAIZA SYAHRANI. **Application of Convolutional Neural Network in Classification Based on Aflatoxin Contamination Level in Corn Samples** (supervised by Amil Ahmad Ilham).

**Background.** The attention mechanism in Deep Learning (DL) can enhance a model's focus on important parts of input text. Although its advantages are proven in text classification, the impact of varying the number of attention heads on model performance and efficiency still needs to be investigated. **Aim.** This research aims to analyze the effect of the number of heads in the attention layer of a DL model for text classification, and to compare it with models without attention. **Method.** A Bidirectional Long Short-Term Memory (BiLSTM) model with multi-head attention was implemented using Text REtrieval Conference (TREC) question data from the University of Southern California (USC), comprising 5,942 data points. As part of the pre-processing stage, the ABBR (abbreviation) category was excluded due to its imbalanced distribution relative to other categories. Experiments included a grid search scenario for hyperparameter optimization and a testing scenario for varying the number of attention heads (0 to 150). Accuracy, confusion matrix, and F1-score metrics were used for evaluation. **Results.** The model without attention, with an accuracy of 93.27%, outperformed some variations of models with attention. However, the highest accuracy was obtained by the model with 50 attention heads, achieving 94.09% accuracy. All models with attention had more efficient training times compared to the model without attention. **Conclusion.** Adding attention heads does not always provide a significant increase in accuracy. However, it can improve training time efficiency.

Keywords: text classification; deep learning; attention; multi-head attention; BiLSTM

# DAFTAR ISI

**Halaman**

[ABSTRAK i](#_Toc202001782)

[ABSTRACT ii](#_Toc202001783)

[DAFTAR ISI iii](#_Toc202001784)

[DAFTAR TABEL v](#_Toc202001785)

[DAFTAR GAMBAR vi](#_Toc202001786)

[DAFTAR LAMPIRAN vii](#_Toc202001787)

[BAB I PENDAHULUAN 1](#_Toc202001788)

[1.1 Latar belakang 1](#_Toc202001789)

[1.2 Landasan Teori 2](#_Toc202001790)

[1.2.1 *Natural* *Language* *Processing*/NLP 2](#_Toc202001791)

[1.2.2 Representasi Teks 5](#_Toc202001792)

[1.2.3 *Deep* *Learning* 6](#_Toc202001793)

[1.2.4 *Grid* *Search* 12](#_Toc202001794)

[1.2.5 Evaluasi Model 13](#_Toc202001795)

[1.3 Tujuan dan Manfaat 14](#_Toc202001796)

[1.3.1 Tujuan 14](#_Toc202001797)

[1.3.2 Manfaat 14](#_Toc202001798)

[BAB II METODE PENELITIAN 15](#_Toc202001799)

[2.1 Tempat dan Waktu 15](#_Toc202001800)

[2.2 Instrumen Penelitian 15](#_Toc202001801)

[2.3 Tahapan Penelitian 15](#_Toc202001802)

[2.4 Perancangan dan Implementasi Sistem 16](#_Toc202001803)

[2.4.1 *Data* *Collection* 17](#_Toc202001804)

[2.4.2 *Exploratory* *Data* *Analysis* 18](#_Toc202001805)

[2.4.3 *Data* *Pre*-*processing* 20](#_Toc202001806)

[2.4.4 Perancangan Model 21](#_Toc202001807)

[2.4.5 Skenario Penelitian 22](#_Toc202001808)

[2.4.6 Training dan Evaluasi Model 23](#_Toc202001809)

[2.4.7 Hasil 23](#_Toc202001810)

[BAB III HASIL DAN PEMBAHASAN 24](#_Toc202001811)

[3.1 Hasil Pembuatan Model 24](#_Toc202001812)

[3.2 Hasil Analisis 25](#_Toc202001813)

[3.2.1 Skenario Eksperimen *Grid* *Search*. 26](#_Toc202001814)

[3.2.2 Skenario Eksperimen Variasi Jumlah *Head* *Multi*-*Head* *Attention* 27](#_Toc202001815)

[3.2.3 Perbandingan Fokus pada *Head* *Attention* 30](#_Toc202001816)

[BAB IV KESIMPULAN DAN SARAN 35](#_Toc202001817)

[4.1 Kesimpulan 35](#_Toc202001818)

[4.2 Saran 35](#_Toc202001819)

[DAFTAR PUSTAKA 36](#_Toc202001820)

[LAMPIRAN 40](#_Toc202001821)

# DAFTAR TABEL

|  |  |
| --- | --- |
| Nomor urut | Halaman |

1. [Skenario penelitian 22](#_Toc200490593)
2. [Hasil eksperimen *grid search* dengan 10 kombinasi *hyperparameter* terbaik. 26](#_Toc200490594)
3. [Pengaruh intervensi data terhadap jumlah *vocabulary*. 26](#_Toc200490595)
4. [Hasil akurasi skenario jumlah *head*. 27](#_Toc200490596)
5. [Nilai TP, FP, FN, dan TN dari *confusion matrix* model berdasarkan jumlah *head*. 28](#_Toc200490597)
6. [Perbandingan prediksi pada model 1 *head*, 3 *head*, dan 50 *head*. 30](#_Toc200490598)

# DAFTAR GAMBAR

|  |  |
| --- | --- |
| Nomor urut | Halaman |

1. [Ilustrasi skip-gram (Johnson et al., 2024) 5](#_Toc200490657)
2. [Ilustrasi *Neural Network* (NN) (Herrmann & Kollmannsberger, 2024). 6](#_Toc200490658)
3. [Sel LSTM (Gao et al., 2023). 7](#_Toc200490659)
4. [(a) Grafik *output* GELU dan (b) grafik *output* ReLU (Raschka, 2024). 10](#_Toc200490660)
5. [(a) NN yang belum menggunakan dropout dan (b) NN setelah menggunakan dropout (Salehin & Kang, 2023). 12](#_Toc200490661)
6. [Tahapan penelitian. 16](#_Toc200490662)
7. [Perancangan alur sistem. 17](#_Toc200490663)
8. [Grafik histogram persebaran data berdasarkan kategori. 19](#_Toc200490664)
9. [Grafik histogram persebaran data berdasarkan kategori setelah menghapus kategori *ABBR*. 19](#_Toc200490665)
10. [Jumlah kata yang paling sering muncul pada data selain *stopwords* pada data *train* 20](#_Toc200490666)
11. [(a) Arsitektur AC-BiLSTM oleh Liu & Guo (2019) dan (b) arsitektur GPT-2 oleh Raschka (2024). 22](#_Toc200490667)
12. [Arsitektur model BiLSTM dan attention yang diusulkan. 24](#_Toc200490668)
13. [*Feed forward* NN pada model. 25](#_Toc200490669)
14. [*Output* *attention* model dengan 1 *head*. 31](#_Toc200490670)
15. [*Output attention* model dengan 3 *head* pada (a) *head* 1, (b) *head* 2, dan (c) *head* 3. 32](#_Toc200490671)
16. [*Output attention* model dengan 50 *head* pada (a) *head* 4, (b) *head* 15, (c) *head* 25, (d) *head* 26, (e) *head* 29, dan (f) *head* 47. 34](#_Toc200490672)

# DAFTAR LAMPIRAN

|  |  |
| --- | --- |
| Nomor urut | Halaman |

1. [*Confusion matrix* setiap model 40](#_Toc200490682)
2. [Program model 42](#_Toc200490683)
3. [Hasil eksperimen *grid search* 43](#_Toc200490684)
4. [Hasil eksperimen variasi jumlah *head* *multi*-*head attention* 44](#_Toc200490685)
5. [*Source code* 45](#_Toc200490686)

|  |  |
| --- | --- |
| Istilah | Arti dan Penjelasan |
| <PAD> | token yang ditambahkan ke input yang memiliki panjang kurang dari t untuk mencapai panjang yang diinginkan |
| <UNK> | token yang mewakili kata-kata yang tidak terdapat pada *vocabulary* model |
| *Backpropagation* | algoritma yang digunakan dalam pelatihan *neural network* untuk menghitung gradien dari *loss function* terhadap *weights* kemudian memperbarui *weights* model untuk meminimalkan *loss* |
| *Batch size* | Ukuran data yang diproses per iterasi training |
| *Exploratory Data Analysis* | tahap untuk memahami karakteristik data secara mendalam seperti panjang kata, ciri khas kata per kategori, dan distribusi data |
| *Hyperparameter* | parameter model yang diatur sebelum proses *training* dimulai, seperti learning rate, dropout, dan weight decay |
| *Optimizer* | Algoritma yang digunakan untuk memperbarui weights model |
| *Overfitting* | kondisi di mana model tidak dapat mempelajari pola pada data *train* |
| Sampel | Satu buah entri data individual dalam data, seperti satu kalimat pertanyaan beserta label |
| *Training* | proses mesin mempelajari pola pada data *train* dan divalidasi pada data *validation* |

|  |  |
| --- | --- |
| **Lambang/Singkatan** | **Arti dan Penjelasan** |
| CV | *Computer Vision* |
| NLP | *Natural* *Language Processing* |
| DL | *Deep Learning* |
| BERT | *Bidirectional Encoder Representations from Transformers* |
| GPT | *Generative Pre-trained Transformer* |
| NER | *Named Entity Recognition* |
| OCR | *Optical Character Recognition* |
| NLU | *Natural Language Understanding* |
| NLG | *Natural Language Generation* |
| NN | *Neural Network* |
| RNN | *Recurrent Neural Network* |
| LSTM | *Long Short-Term Memory* |
| DNN | *Deep Neural Network* |
| CNN | *Convolutional Neural Network* |
| GELU | *Gaussian Error Linear Unit* |
| ReLU | *Rectified Linear Unit* |
| EDA | *Exploratory Data Analysis* |
|  | *Output* dari *hidden layer* pada NN |
|  | *Weight* model |
|  | Nilai aktivasi neuron ke- pada *layer* ke- |
|  | *Loss function* |
|  | Parameter NN yang terdiri dari *weight* dan *bias*. |
|  | *Forget gate* pada sel LSTM |
|  | *Input gate* pada sel LSTM |
|  | Nilai *cell state* yang ingin ditambahkan pada sel LSTM |
|  | *Output gate* pada sel LSTM |
|  | *Cell state* sel LSTM |
|  | *Hidden state* sel LSTM |
|  | *Output* *forward* BiLSTM |
|  | *Output backward* BiLSTM |
|  | *Context vector attention* dari arah *forward* |
|  | *Context vector attention* dari arah *backward* |
|  | *Output* dari model |
| *ABBR* | Singkatan ( *eviation*) |
| *ENTY* | Entitas (*entity*) |
| *DESC* | Deskripsi (*description*) |
| *HUM* | Manusia (*human*) |
| *LOC* | Lokasi (*location*) |
| *NUM* | Nomor (*number*) |

# PENDAHULUAN

## Latar belakang

Industri pakan di Indonesia memiliki peran penting dalam mendukung upaya meningkatkan kualitas sektor peternakan. Sebagai negara dengan populasi ternak yang besar seperti yang tercatat pada hasil ST2023 (humahpkh, 2023), kebutuhan akan pakan yang berkualitas terus meningkat untuk memastikan pertumbuhan dan kesehatan ternak yang optimal. Penyediaan pakan yang sesuai dengan kebutuhan ternak menjadi faktor utama dalam keberhasilan peningkatan kualitas dan kuantitas ternak. Dalam formulasi pakan, bahan baku yang digunakan harus memenuhi standar kualitas agar dapat memberikan manfaat maksimal bagi ternak serta mencegah potensi risiko kontaminasi yang dapat merugikan.

Pakan yang seimbang dan bermutu dipengaruhi oleh jenis bahan pakan yang digunakan dalam menyusun formula ransum. Jagung sebagai sumber energi merupakan komposisi terbesar dalam penyusunan formula ransum yang mencapai 50% - 60% dari total bahan pakan (Ali et al., 2023).Sebagai bahan utama dalam pakan ternak, menjaga kualitas jagung sangat penting agar tidak menimbulkan dampak negatif terhadap kesehatan hewan ternak maupun manusia yang akan mengkonsumsi produk hasil ternak tersebut. Salah satu permasalah utama dalam penyimpanan dan distribusi jagung adalah kontaminasi jamur. Jagung yang disimpan dalam kondisi yang tidak optimal, seperti kelembaban tinggi dan ventilasi buruk, menjadi media ideal bagi pertumbuhan jamur toksigenik. Aspergillus spp., Fusarium spp., dan Penicillium spp. merupakan spesies jamur cemaran yang paling sering menghasilkan mikotoksin berbahaya bagi konsumen.

Mikotoksin merupakan metabolit sekunder yang dihasilkan oleh spesies kapang tertentu selama pertumbuhannya pada bahan pangan maupun pakan. Sebuah studi yang dilakukan oleh DSM Mycotoxin Survey di tahun 2024 menemukan bahwa kontaminasi aflatoksin di komoditas sedang mengalami peningkatan (Kodape & Kodape, 2024). Keberadaan mikotoksin dalam pakan ternak berdampak terhadap kesehatan ternak berupa penurunan produksi, reproduksi dan berat badan, mempengaruhi sistem imunosupresi dan menimbulkan kerusakan organ serta dapat menimbulkan residu yang dapat membahayakan manusia (Widiastuti, 2024).

Salah satu jenis mikotoksin yang paling berpotensi menimbulkan permasalahan adalah aflatoksin. Aflatoksin merupakan metabolit yang diproduksi oleh galur jamur toksigenik, terutama Aspergillus flavus dan Aspergillus parasiticus, yang tumbuh di tanah, jerami, tumbuhan yang membusuk, dan biji-bijian (Dhakal et al., 2023). Diantara keempat jenis aflatoksin AFB1, AFB2, AFG1, dan AFG2, AFB1 paling mendapat banyak perhatian karena tingkat toksik nya yang paling tinggi dan bersifat karsinogenik, hepatotoksik, dan mutagenik. Keracunan akut yang dihasilkan oleh aflatoksin terjadi di organ hati yang menyebabkan kegagalan metabolisme karbohidrat dan lemak dan sintesis protein, sehingga fungsi hati mengalami penurunan karena adanya perombakan pembekuan darah (Widiastuti, 2024).

Untuk memastikan kualitas produk hasil ternak, industri pakan ternak harus menjaga standar kualitas pakan melalui pengawasan yang ketat. Salah satu hal penting dalam pengendalian kualitas adalah memastikan kandungan aflatoksin dalam jagung yang terkandung tidak melebihi batas yang telah ditetapkan oleh perusahaan pakan ternak. Oleh karena itu, tim Quality Control (QC) bertanggung jawab dalam melakukan pengujian kandungan aflatoksin pada jagung.

Proses pengujian kandungan aflatoksin dilakukan dengan mengambil sampel jagung yang sebelumnya telah melalui tahap penggilingan terlebih dahulu. Langkah ini bertujuan agar aflatoksin yang cenderung berkembang di bagian dalam biji jagung dapat terlihat. Setelah itu sampel jagung ditempatkan di ruangan khusus dengan kondisi gelap dan dilengkapi dengan sinar UV sebagai alat bantu untuk mendeteksi keberadaan aflatoksin secara kasat mata atau pemeriksaan visual.

Saat ini, penentuan kandungan aflatoksin masih dilakukan secara konvensional, yaitu dengan melakukan uji laboratorium hingga dengan mengidentifikasi dan menentukan nilai aflatoksin satu per satu keping jagung yang telah digerus yang menunjukkan intensitas yang berbeda di bawah sinar UV menggunakan alat penghitung manual hand tally counter oleh perusahaan pakan ternak. Namun, uji laboratorium ini memiliki kelemahan, seperti keterbatasan waktu analisis, kebutuhan biaya operasional yang tinggi, dan membutuhkan tenaga ahli yang terlatih (Maphaisa et al., 2025). Sedangkan untuk perhitungan manual oleh tim Quality Control (QC) perusahaan, dalam proses pemeriksaan visual terdapat perbedaan ukuran dan tingkat kecerahan dari aflatoksin, yang menyebabkan variasi dalam pembacaan hasil dapat memiliki pendapat yang berbeda atau subjektif. Perbedaan perkiraan antar anggota tim Quality Control (QC) dapat menyebabkan ketidakkonsistenan dalam perhitungan total kandungan aflatoksin, sehingga meningkatkan risiko kesalahan dalam pengukuran.

Metode yang masih sering digunakan dalam pengujian kandungan aflatoksin berupa pengujian laboratorium dengan metode seperti Enzyme-Linked Immunosorbent Assay (ELISA) (Widiyanti, 2020). Metode ini merupakan pengujian laboratorium yang terbukti dalam mendeteksi aflatoksin dengan hasil analisis masih di bawah batas regulasi SNI namun membutuhkan biaya mahal dan waktu pengujian yang memakan waktu yang harus dilakukan oleh tenaga ahli khusus. Pada penelitian lainnya dengan judul Desain Computer Vision Berbasis Citra Reflektansi dan Fluoresensi Menggunakan Machine Learning Sebagai Pendeteksi Biji Kakao (Sadimantara, Syukri, et al., 2024), analisis kadar aflatoksin pada penelitian tersebut dilakukan pada biji kakao yang juga menggunakan arsitektur CNN dalam mengklasifikasikan tingkat kontaminasi aflatoksin dengan mempelajari fitur citra. Sistem ini berhasil mengklasifikasikan kandungan aflatoksin dengan nilai akurasi sebesar 96%. Untuk pengembangannya, akan dilakukan penelitian lebih lanjut untuk menguji performa CNN dalam melakukan klasifikasi tingkat kontaminasi aflatoksin pada objek yang lebih kecil, yaitu jagung dengan menggunakan arsitektur turunan CNN guna meningkatkan akurasi dan efisiensi model.

Maka pada penelitian ini, akan dibuat sebuah sistem klasifikasi tingkat kontaminasi aflatoksin. Penelitian ini merupakan bagian dari penelitian disertasi yang bertujuan menghasilkan aplikasi prediksi kualitas jagung dengan menerapkan pendekataan deep learning, sistem ini menjadi solusi bagi industri pakan ternak dalam menentukan tingkat kontaminasi jagung yang masih dilakukan secara konvensional. Dalam prosesnya, akan dikumpulkan dataset citra hasil dari perhitungan manual, yang kemudian akan diklasifikasikan ke dalam 3 kelas, yaitu jagung dengan tingkat kontaminasi rendah (0-75 PPB), sedang (76-150 PPB), dan tinggi (151-200 PPB). Penelitian ini ditujukan sebagai kontribusi terhadap industri pakan ternak dalam menentukan kadar aflatoksin dengan waktu yang lebih cepat dan akurat, sehingga mendukung proses pengolahan yang lebih efisien.

## Landasan Teori

### Kontaminasi Aflatoksin Jagung

sdfhiwhfe.

### Visi Komputer

sgergsdv.

### *Deep Learning*

aididfio.

### *Convolutional Neural Network* (CNN)

acoief.

### Tensorflow

Sogiowrg

### Evaluasi Model

Aegwoig.

## Tujuan dan Manfaat

### Tujuan

Tujuan dari penelitian ini adalah:

1. Mengetahui dan memahami bagaimana penerapan *Convolutional Neural Network* (CNN) dalam pengembangan sistem klasifikasi tingkat kontaminasi aflatoksin pada jagung.
2. Menganalisis nilai akurasi klasifikasi tingkat kontaminasi aflatoksin menggunakan model *Convolutional Neural Network* (CNN).

### Manfaat

Manfaat dari penelitian ini adalah:

1. Membantu tim *quality control* dalam meningkatkan efisiensi waktu kerja penentuan tingkat kelas kontaminasi aflatoksin pada jagung.
2. Memberikan kontribusi pada bidang penelitian terkait klasifikasi tingkat kontaminasi aflatoksin pada jagung.

# METODE PENELITIAN

## Tempat dan Waktu

Kegiatan penelitian dilaksanakan sejak Juni 2025 hingga BULAN SELESAI di beberapa tempat, yaitu LOKASI JAGUNG sebagai tempai pemngambilan data primer dan Laboratorium *Cloud Computing and Information System* (CCIS), Departemen Teknik Informatika, Fakultas Teknik, Universitas Hasanuddin sebagai tempat pengolahan dan analisis data.

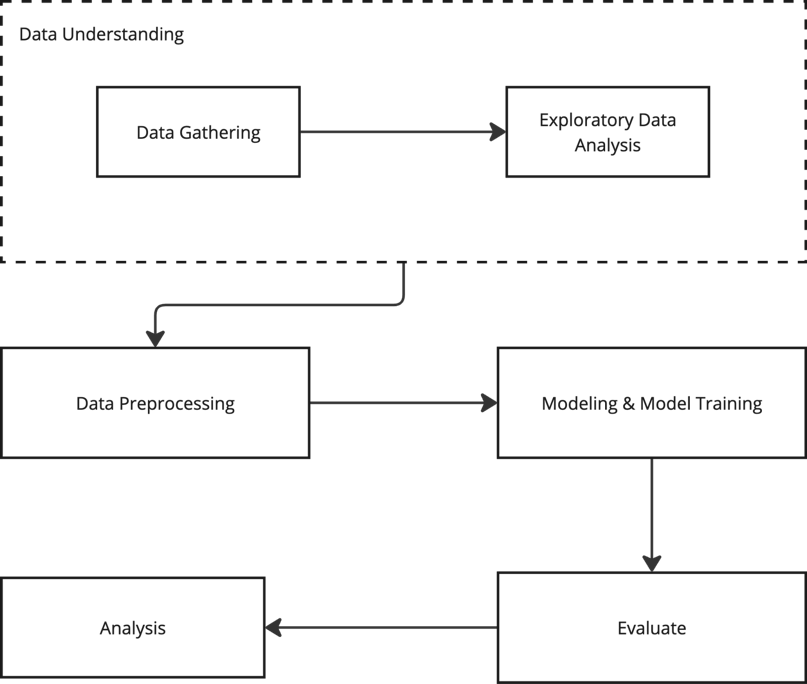
## Instrumen Penelitian

Berikut instrumen atau alat yang digunakan dalam penelitian ini:

1. Perangkat lunak:
   1. Visual Studio Code
   2. Microsoft Word
   3. Microsoft Excel
   4. Microsoft PowerPoint
   5. Windows 11
   6. Mendeley
2. Perangkat keras:
   1. 11Th Gen Intel(R) Core(TM) i9-11900F @ 2.50GHz, RAM 32GB,
   2. 11th Gen Intel(R) Core(TM) i7-1165G7 @ 2.80GHz, RAM 16GB
3. Bahasa pemrograman:
   1. Python v3.13.1
4. Algoritma:
   1. ResNet-50
5. *Library*:
   1. TensorFlow v2.20.0, digunakan untuk membangun, melatih, dan menyimpan model *deep learning* berbasis ResNet-50.
   2. Pathlib v3.13.1, digunakan untuk mengelola dan memanipulasi direktori serta jalur file dataset.
   3. Shutil v3.13.1, digunakan untuk menyalin file saat pembagian dataset ke folder *train*, *val*, dan *test*.
   4. Scikit-learn v1.6.1, digunakan untuk membagi data menjadi *train*, *validation*, dan *test set*, serta menghitung metrik evaluasi regresi (MAE, MSE, R²).
   5. Numpy v2.2.1, digunakan untuk pengolahan array numerik dan perhitungan matematis sederhana seperti RMSE.
   6. Matplotlib v3.10.1, digunakan untuk membuat visualisasi hasil pelatihan dan grafik perbandingan antara nilai aktual dan prediksi.

## Tahapan Penelitian

Penelitian ini dimulai dengan tahap *data understanding*, yang terdiri dari pengumpulan data dan memahami karakteristik data (*Exploratory Data Analysis*/EDA), seperti panjang kata pada setiap data, ciri khas kata yang muncul pada setiap kategori, dan distribusi jumlah data berdasarkan kategori. Selanjutnya dilakukan tahapan *data pre-processing* untuk membersihkan data dari karakter yang dapat mengganggu performa model. Pada tahap ketiga dilakukan *modeling & model training*, di mana model dibuat dan diuji pada berbagai skenario untuk mendapatkan kombinasi hyperparameter model dengan performa tertinggi. Model dengan performa tertinggi diuji coba dengan berbagai jumlah *head* pada *multi-head* *attention* sehingga seluruh data pada berbagai skenario dikumpulkan pada tahap *evaluate*. Hasil dari *evaluate* dianalisis untuk pada tahap *analysis* untuk mencari tahu pengaruh jumlah *head* pada *multi-head attention* dengan performa model.



Gambar 6. Tahapan penelitian.

## Perancangan dan Implementasi Sistem

Gambar 7 merupakan diagram untuk perancangan alur sistem.

A screenshot of a black screen

AI-generated content may be incorrect.

Gambar 7. Perancangan alur sistem.

### Data Collection

Data yang digunakan merupakan data TREC yang berisi daftar kalimat pertanyaan dalam Bahasa Inggris dengan label kategori pertanyaan. Terdapat 5,452 data *train* dan 500 data *test* yang masing-masing memiliki 3 kolom, yakni *text*, *coarse\_label*, dan *fine\_label*. Kolom *coarse\_label* merupakan label teks secara umum, sedangkan *fine\_label* merupakan label teks yang lebih spesifik dari *coarse\_label*. Untuk menyederhanakan proses klasifikasi, penelitian ini hanya menggunakan *coarse\_label*. Berikut penjelasan lebih detail untuk masing-masing kategori pada *coarse\_label*:

1. *ABBR*  
   *ABBR* berisi kalimat yang bertanya mengenai singkatan suatu kata. Ciri khas dari kategori ini adalah memiliki kata ‘*stand for*’ atau ‘*abbreviation*’ pada kalimat. Berikut adalah contoh data dari *ABBR*:
   1. *What does* LOL *mean* ?
   2. *What does* IOC *stand for* ?
   3. *What does the abbreviation* IOC *stand for* ?
2. *ENTY*  
   *ENTY* berisi kalimat yang bertanya mengenai entitas. Secara harfiah kategori ini mungkin bersinggungan dengan kategori *HUM* (*human*/manusia), tetapi tetap memiliki ciri khas yang berbeda. *ENTY* hanya bertanya mengenai benda, makhluk hidup, atau hal lain selain manusia. Berikut adalah contoh data dari *ENTY*:
   1. *What two body parts grow all your life* ?
   2. *What is the main ingredient of yogurt* ?
   3. Garry Kasparov *plays what game* ?
3. *DESC*  
   *DESC* merupakan kalimat yang bertanya mengenai deskripsi atau konsep abstrak. Kategori ini biasanya memiliki kata ‘*what is*’. Berikut adalah contoh data dari *DESC*:
   1. *What is Zionism* ?
   2. *What is meant by the term* `` *yield to maturity* '' *in reference to bonds* ?
   3. *How did the war of* 1812 *affect* Delaware?
4. *HUM*  
   *HUM* merupakan kalimat yang bertanya mengenai manusia. Kalimat pertanyaan pada kategori ini merujuk kepada manusia, biasanya terdapat kata ‘*who*’ sebagai kata tanya. Berikut adalah contoh data dari *HUM*:
   1. *What* Hungarian *cardinal was first a state prisoner and then a refugee in the* U.S. *embassy* 1956-1971 ?
   2. *Who won the* Superbowl *in* ?
   3. *What* Italian *leader had a lifelong fear of the evil eye* ?
5. *LOC*  
   *LOC* berisi kalimat pertanyaan mengenai lokasi. Ciri khas dari kategori ini adalah memiliki kata geografis atau kata ‘*where*’ sebagai kalimat tanya. Berikut adalah contoh data dari *LOC*:
   1. *Where does the tennis star* Stefan Edberg *come from* ?
   2. *What continent pushes up the Executive Committee mountain range* ?
   3. *In what county is* Eckley Colorado ?
6. *NUM*  
   *NUM* berisi kalimat pertanyaan mengenai nilai numerik. Kategori ini memiliki ciri khas angka sebagai jawaban dari pertanyaan atau memiliki kata ‘*many*’ dalam kalimat. Berikut adalah contoh data dari *NUM*:
   1. *How many web servers are there* ?
   2. *What date did man first land on the moon* ?
   3. *When did* Spain *and* Korea *start* *ambassadorial relations* ?

### Exploratory Data Analysis

Gambar 8 menunjukkan persebaran data berdasarkan kategori tidak seimbang baik pada data *train* maupun pada data *test*. Pada kedua data, kategori *ABBR* memiliki rasio jumlah yang sangat kecil dibanding kategori yang lain. Kategori ini tidak cukup urgen untuk dilakukan klasifikasi karena setiap negara dapat memiliki singkatan yang berbeda sehingga bisa dihilangkan dari data.

A graph of blue bars

AI-generated content may be incorrect.

Gambar 8. Grafik histogram persebaran data berdasarkan kategori.

Dapat dilihat pada Gambar 9 data menjadi lebih seimbang setelah kategori *ABBR* dihapus.

A graph of blue bars

AI-generated content may be incorrect.

Gambar 9. Grafik histogram persebaran data berdasarkan kategori setelah menghapus kategori *ABBR*.

Selanjutnya peneliti tertarik untuk melihat kata yang paling sering muncul pada data. Berdasarkan Gambar 10, kata ‘*many*’ sebagai kata terbanyak menunjukkan kategori *NUM* banyak menggunakan kata tersebut untuk bertanya mengenai jumlah. Kata ‘*name*’ merupakan ciri khas dari kategori *HUM* dan *ENTY*, terbukti dengan jumlah kedua kategori tersebut menjadi yang terbanyak berdasarkan Gambar 9.

A graph with blue and white bars

AI-generated content may be incorrect.

Gambar 10. Jumlah kata yang paling sering muncul pada data selain *stopwords* pada data *train*

### Data Pre-processing

Tahap ini meliputi modifikasi data sebelum akhirnya digunakan oleh model pada proses *training*. Intervensi data juga diterapkan dengan menghapus seluruh data untuk kategori ABBR karena jumlah data pada kategori ini tidak seimbang dengan jumlah data pada kategori lainnya sehingga kategori pada label hanya berjumlah 5 (*ENTY*, *DESC*, *HUM*, *LOC*, dan *NUM*). Modifikasi yang diterapkan pada penelitian ini adalah *case folding*, *stopwords*, *punctuation removal*, *stemming*, *lemmatization*, dan *tokenization*. Beberapa modifikasi diterapkan di beberapa skenario yang berbeda (lihat subbab 2.4.5) untuk menguji data yang memberikan performa terbaik pada model. Istilah tanpa intervensi data digunakan untuk mencakup hanya beberapa dari teknik di bawah ini. Berikut penjelasan lebih lanjut dari modifikasi yang dilakukan:

1. *Case folding* mengubah huruf kapital pada teks menjadi huruf kecil. Modifikasi ini berguna untuk menghindari kata yang sama, tetapi penggunaan kapital yang berbeda pada *vocabulary*. *Case folding* mengubah teks “*Who won the Superbowl in* ?” menjadi “*who won the superbowl in* ?”.
2. *Stopwords* menghapus kata-kata yang dianggap tidak memberi makna pada data. Beberapa kata yang dimaksud di antaranya, *at*, *is*, *on*, *what*, dan *if*. Modifikasi ini membantu membuang data yang tidak berguna. *Stopwords* mengubah teks “*What is* HTML ?” menjadi “HTML”.
3. *Punctuation removal* menghilangkan tanda baca pada data karena dianggap tidak memberi makna yang berarti. *Punctiation removal* mengubah teks “*The film* " Jaws '' *was made in what year* ?” menjadi “*The film* Jaws *was made in what year*”.
4. *Stemming* mengubah sebuah kata menjadi bentuk dasarnya. Modifikasi yang dihasilkan oleh *stemming* biasanya bukan kata baku karena tidak memerhatikan struktur linguistik. Stemming mengubah teks “*How can I easily remove red wine stains from t-shirts* ?” menjadi “*How can I easili remove red wine stain from t-shirt* ?”.
5. *Lemmatization* memiliki penggunaan yang sama seperti *stemming*. Namun, lemmatization memerhatikan struktur linguistik sehingga modifikasi yang dihasilkan lebih baik dibandingkan *stemming*. Lemmatization mengubah teks “*Where is the human skin least sensitive ?*” menjadi “*Where be the human skin least sensitive ?*”.
6. *Tokenization* mengubah teks menjadi bentuk token-token numerik yang memudahkan model untuk mengolah data teks. Seluruh kata pada *vocabulary* diberikan nomor unik yang berbeda untuk setiap kata. Tokenization mengubah kata “*who won the superbowl in*” menjadi “5, 87, 3, 46, 89”.

Selanjutnya dilakukan *split train test* yakni data *train* dibagi menjadi data *train* dan data *validation* dengan rasio data *train* sebanyak 80% dan data *validation* sebanyak 20%. Pada proses ini dipastikan distribusi kategori data memiliki rasio yang sama sebelum dan sesudah *split train test*.

### Perancangan Model

Model dibangun mengambil referensi arsitektur BiLSTM dan *attention* oleh Liu & Guo (2019), di mana untuk setiap *output hidden states forward* dan *backward*  diproses oleh lapisan *multi*-*head attention* yang berbeda. *Output* *multi*-*head attention* dari keduanya digabung lalu diproses oleh lapisan NN berikutnya (lihat Gambar 11 (a)). Selain itu, referensi juga diambil dari arsitektur GPT-2 berdasarkan buku oleh Raschka (2024) untuk penempatan *dropout layer*, *normalization layer*, dan *feed forward* (lihat Gambar 11 (b)).

|  |  |
| --- | --- |
| A diagram of a computer network  AI-generated content may be incorrect. | A diagram of a diagram  AI-generated content may be incorrect. |
| (a) | (b) |

Gambar 11. (a) Arsitektur AC-BiLSTM oleh Liu & Guo (2019) dan (b) arsitektur GPT-2 oleh Raschka (2024).

### Skenario Penelitian

Penelitian ini menggunakan dua skenario. Skenario pertama menggunakan *grid search* dengan *hyperparameter* *dropout*, *learning rate*, *weight decay*, dan intervensi data. *Hyperparameter* dari hasil *grid search* dengan akurasi tertinggi digunakan pada skenario kedua yang menguji variasi jumlah *head multi*-*head attention* dan tanpa *attention*. Penjelasan tiap skenario dapat dilihat pada Tabel 1. Jabarkan hyperparameter skenario pertama

Tabel 1. Skenario penelitian

|  |  |
| --- | --- |
| No. | Skenario |
| 1 | Eksperimen *grid search* *hyperparameter*:   1. *dropout* (0.1, 0.3, dan 0.5). 2. *learning rate* (0.01, 0.001, dan 0.0001). 3. *weight decay* (0.01, 0.001, dan 0.0001). 4. Intervensi data (Tanpa intervensi, *stopwords*, *stemming*, dan *lemmatization*). |
| 2 | Eksperimen variasi jumlah *head* (1, 2, 3, 5, 6, 10, 15, 25, 30, 50, 75, dan 150) dan tanpa *attention*. |

### *Training* dan Evaluasi Model

*Pre-trained* *word vectors* dari Google News yang tersedia secara publik digunakan pada penelitian ini dan digunakan pada lapisan *embedding* pada model. Lapisan *embedding* di-*freeze* sehingga *weights* pada lapisan ini tidak berubah selama proses *training*. Dimensi *embedding* yang digunakan adalah 300 dengan *batch size* sebesar 50 selama proses *training*. *Optimizer* yang digunakan adalah AdamW dengan nilai *epoch* 100 dan menggunakan *early stopping* dan *model checkpointing* selama proses *training*. Untuk setiap *epoch*, model diuji pada data *validation*. Performa model dari seluruh skenario dievaluasi menggunakan metrik akurasi. F1-*score* juga digunakan untuk mengevaluasi performa model pada setiap kategori.

### Hasil

Hasil akhir dari penelitian ini adalah menganalisis hasil dari tahap *training* dan *testing* dari seluruh varian model pada skenario jumlah *head multi*-*head attention*. Dari tahap ini diketahui bagaimana pengaruh jumlah *heads* pada mekanisme *attention* memengaruhi akurasi dan durasi waktu *training* model. Hasil yang diberikan diharapkan dapat menjadi referensi bagi peneliti lain untuk mengembangkan model dengan mekanisme *attention*.

# HASIL DAN PEMBAHASAN

## Hasil Pembuatan Model

Berdasarkan penelitian yang dilakukan, hasil pembuatan model dapat dilihat pada Gambar 12.

A diagram of a computer network

AI-generated content may be incorrect.

Gambar 12. Arsitektur model BiLSTM dan attention yang diusulkan.

Model yang diusulkan menghilangkan *convolutional layer* pada model AC-LSTM milik Liu & Guo (2019) dan menambahkan *dropout layer*, *normalization*, dan *feed forward*. Model yang dibangun memiliki *input layer* dengan input sepanjang , di mana merupakan input yang telah diubah menjadi bentuk token. Indeks yang memiliki panjang lebih dari hanya diambil indeksnya hingga dan indeks yang memiliki panjang kurang dari ditambahkan token <PAD> hingga panjang menjadi . Selanjutnya diproses oleh *embedding layer* sehingga seluruh memiliki dimensi 300. Pada proses *training* diaktifkan *dropout* pada *embedding layer* sehingga beberapa indeks pada diubah nilainya menjadi 0. Selanjutnya diproses oleh BiLSTM secara *forward* dan *backward* sehingga diperoleh *output forward* dan *output backward* yang masing-masing memiliki panjang dan dimensi 150. Selanjutnya dan diproses oleh dua *multi*-*head attention layer* yang berbeda secara paralel sehingga diperoleh *forward context vectors*  dan *backward context vectors* yang masing-masing mengandung informasi hubungan antar kata pada . *Concatenation* (penggabungan) dari dan menghasilkan dengan panjang dan dimensi 300. Selanjutnya diproses oleh *normalization layer*, dilanjut dengan *feed forward NN*, dan terakhir diproses oleh *output layer* sehingga diperoleh *output model* dengan panjang , di mana merupakan jumlah label pada data sehingga sama dengan 5 (lihat subbab 2.4.3). Nilai yang terdapat pada adalah *logits*, yakni nilai skor mentah sebelum dikonversi menjadi probabilitas menggunakan *softmax*, yang digunakan oleh *cross entropy loss* untuk mengukur performa model. *Feed forward* NN pada model, yang mengambil referensi dari model GPT-2, mengembangkan dimensi input sebesar empat kali lalu diproses oleh fungsi aktivasi GELU (dapat dilihat pada persamaan (16)) sehingga model mendapatkan informasi yang lebih kaya pada data (lihat Gambar 13).

A screenshot of a computer

AI-generated content may be incorrect.

Gambar 13. *Feed forward* NN pada model.

## Hasil Analisis

### Skenario Eksperimen Grid Search.

Hasil eksperimen dari *grid search* dapat dilihat pada Tabel 2 di mana *head* yang digunakan berjumlah 1. Isi tabel hanya menampilkan 10 model teratas dengan akurasi tertinggi secara berurutan dari yang terbesar.

Tabel 2. Hasil eksperimen *grid search* dengan 10 kombinasi *hyperparameter* terbaik.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Nomor model | *Dropout* | *Learning rate* | *Weight decay* | Intervensi data | Akurasi (%) |
| 22 | **0.5** | **0.001** | **0.01** | **Tanpa intervensi\*** | **93.89** |
| 23 | 0.5 | 0.001 | 0.001 | Tanpa intervensi\* | 93.27 |
| 95 | 0.3 | 0.001 | 0.001 | *Lemmatization* | 92.87 |
| 103 | 0.5 | 0.001 | 0.01 | *Lemmatization* | 92.66 |
| 13 | 0.3 | 0.001 | 0.01 | Tanpa intervensi\* | 92.66 |
| 5 | 0.1 | 0.001 | 0.001 | Tanpa intervensi\* | 92.26 |
| 105 | 0.5 | 0.001 | 0.0001 | *Lemmatization* | 92.05 |
| 24 | 0.5 | 0.001 | 0.0001 | Tanpa intervensi\* | 91.85 |
| 99 | 0.3 | 0.0001 | 0.0001 | *Lemmatization* | 91.64 |
| 97 | 0.3 | 0.0001 | 0.01 | *Lemmatization* | 91.64 |

\*Tanpa intervensi: tidak menggunakan teknik *stopwords*, *stemming*, dan *lemmatization*.

Model dengan akurasi tertinggi diperoleh oleh model 22 dengan kombinasi *hyperparameter* *dropout* 0.5, *learning rate* 0.001, *weight decay* 0.01, dan tanpa intervensi data. Terdapat selisih akurasi sebesar 0.62% antara model 22 dengan model 23 yang dipengaruhi oleh perbedaan *weight decay*. Model 95 memiliki selisih akurasi sebesar 0.5% dengan model 23, di mana perbedaan dari keduanya adalah model 23 memiliki *dropout* 0.3 dan menggunakan *lemmatization*. Pengaruh intervensi data terhadap jumlah *vocabulary* dapat dilihat pada Tabel 3.

Tabel 3. Pengaruh intervensi data terhadap jumlah *vocabulary*.

|  |  |
| --- | --- |
| Intervensi data | Jumlah *vocabulary* |
| Tanpa intervensi | 8898 |
| *Stopwords* | 8766 |
| *Stemming* | 7375 |
| *Lemmatization* | 8175 |

Penggunaan *stemming* dan *stopwords* menghasilkan jumlah *vocabulary* 7.375 kata dan 8.766 kata secara berurutan. Kedua intervensi data tersebut tidak terdapat pada 10 model dengan akurasi tertinggi karena kata-kata yang penting pada data dihapus sehingga tidak terdapat ciri khas antar kategori yang bisa dibedakan oleh model. Kombinasi *hyperparameter* dari model 22 digunakan pada skenario selanjutnya.

Hasil pada *grid search* juga menunjukkan 20 model dengan performa terendah dipenuhi oleh model dengan *learning rate* 0.01. Hal ini disebabkan oleh nilai *learning rate* tersebut terhitung besar sehingga model mengubah nilai *weights* dengan perubahan yang besar.

### Skenario Eksperimen Variasi Jumlah Head Multi-Head Attention

Hasil eksperimen dari jumlah *head* dapat dilihat pada Tabel 4 di mana 0 *head* adalah model tanpa lapisan *attention* dan *hyperparameter* yang digunakan untuk seluruh model adalah *epoch* 100, *dropout* 0.5, *learning rate* 0.001, *weight decay* 0.01, tanpa intervensi data, dan *patience early stopping* 5.

Tabel 4. Hasil akurasi skenario jumlah *head*.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Jumlah *head* | Akurasi (%) | *Epoch early stopping* | Waktu *training* (menit:detik) | Dimensi setiap *head* |
| 0 | 93.27 | 42 | 02:15 | - |
| 1 | 93.89 | 23 | 01:24 | 150 |
| 2 | 91.85 | 25 | 01:37 | 75 |
| 3 | 91.03 | 18 | 01:11 | 50 |
| 5 | 92.05 | 20 | 01:17 | 30 |
| 6 | 91.64 | 24 | 01:31 | 25 |
| 10 | 93.27 | 22 | 01:22 | 15 |
| 15 | 93.27 | 24 | 01:29 | 10 |
| 25 | 93.07 | 28 | 01:45 | 6 |
| 30 | 93.07 | 24 | 01:29 | 5 |
| 50 | **94.09** | **25** | **01:33** | **3** |
| 75 | 93.48 | 23 | 01:25 | 2 |
| 150 | 93.27 | 21 | 01:19 | 1 |

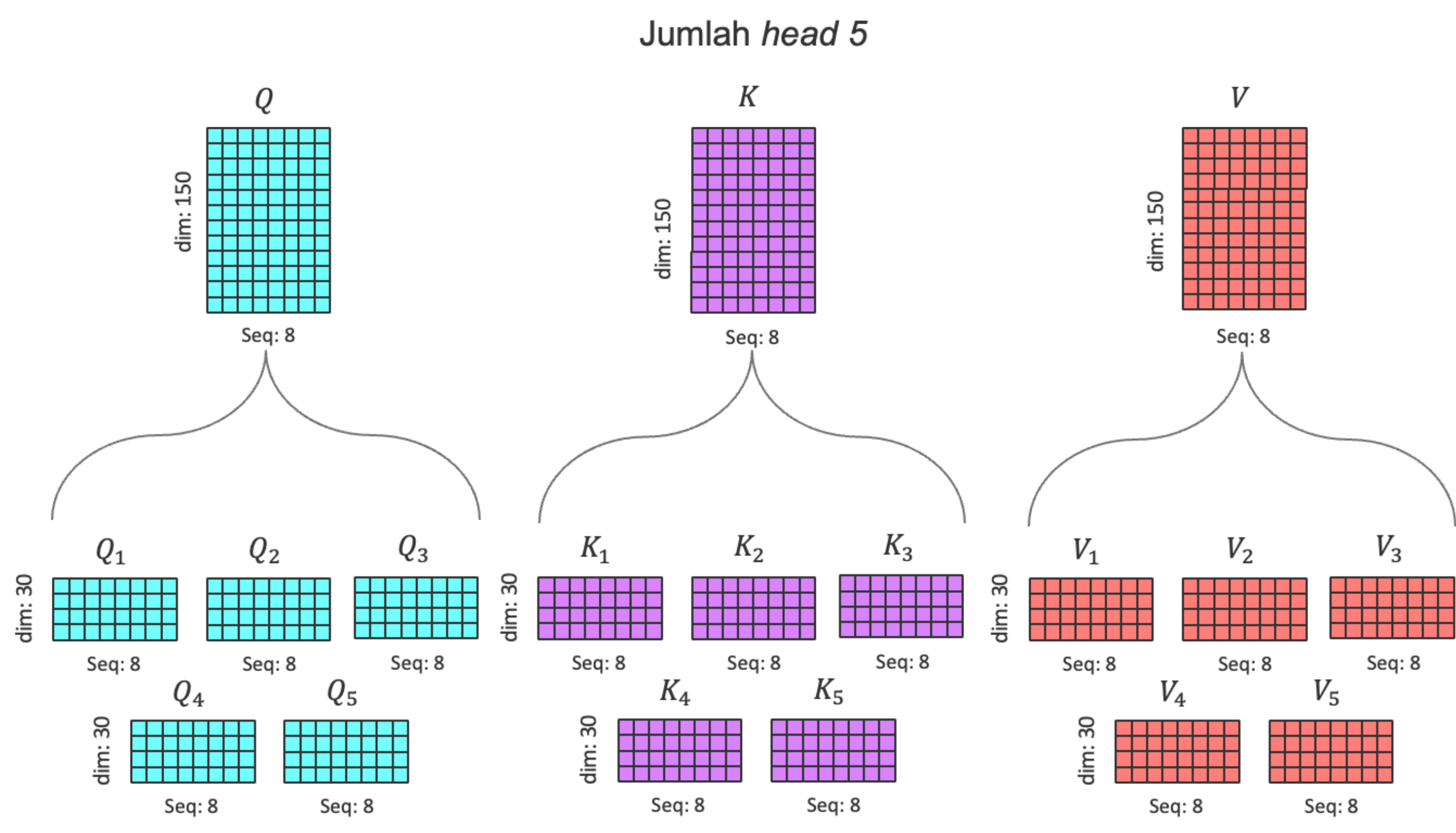
Berdasarkan Tabel 4, model dengan *attention* tidak selalu memiliki performa yang lebih baik dibandingkan dengan model tanpa *attention,* khusunya pada penelitian ini. Ini dapat dilihat berdasarkan akurasi dari beberapa model dengan *attention* yang lebih rendah dibandingkan dengan model tanpa *attention*, menunjukkan bahwa pengaruh *head attention* tidak berkontribusi besar pada riset ini.

Nilai akurasi yang tidak selalu meningkat seiring dengan penambahan jumlah *head* dapat disebabkan oleh perbedaan dimensi pada masing-masing *head*. Dalam mekanisme *multi*-*head* *attention*, setiap *head* membagi dimensi total menjadi beberapa sub-ruang yang lebih kecil (lihat Gambar 14 dan Gambar 15). Perbedaan dimensi ini menyebabkan hasil dari perkalian antara vektor *Query* (), *Key* (), dan *Value* () pada tiap *head* membentuk *attention score* dan *context vector* yang berbeda-beda.

A screenshot of a diagram

AI-generated content may be incorrect.

Gambar 14. Pembagian dimensi vektor untuk 3 *head*.



Gambar 15. Pembagian dimensi vektor untuk 5 *head*.

Model tanpa *attention* membutuhkan waktu 2 menit 15 detik sebelum dihentikan oleh *early stopping* dengan akurasi yang hanya mencapai 93.27%. Proses ini lebih lambat dibandingkan beberapa model yang menggunakan *attention*, seperti model dengan 1 *head* dan 50 *head*, yang mampu mencapai akurasi lebih tinggi dalam waktu kurang dari 2 menit. Ini menunjukkan model dengan *attention* lebih unggul dalam waktu *training* dibandingkan model tanpa *attention*.

Model dengan 50 *head* hanya memproses 3 dimensi *embedding* pada setiap *head*. Meskipun dimensi *embedding* yang diproses kecil, model tersebut memiliki performa yang lebih tinggi dibandingkan model dengan 3 *head* yang memproses 50 dimensi *embedding* pada setiap *head*, di mana model ini merupakan model dengan akurasi terendah. Detail nilai TP, FP, FN, dan TN masing-masing model disajikan pada Tabel 5.

Tabel 5. Nilai TP, FP, FN, dan TN dari *confusion matrix* model berdasarkan jumlah *head*.

| Jumlah *head* | Kategori | TP | FP | FN | TN | F1-Score (%) | *Mean* F1-Score (%) |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 0 | *ENTY* | 81 | 11 | 13 | 386 | **87.09** | **93.57** |
| *DESC* | 136 | 5 | 2 | 348 | **97.49** |
| *HUM* | 61 | 6 | 4 | 420 | 92.42 |
| *LOC* | 77 | 5 | 4 | 405 | 94.47 |
| *NUM* | 107 | 2 | 6 | 376 | 96.39 |
| 1 | *ENTY* | 77 | 7 | 17 | 390 | 86.51 | 93.56 |
| *DESC* | 134 | 9 | 4 | 344 | 95.37 |
| *HUM* | 63 | 7 | 2 | 419 | 93.33 |
| *LOC* | 79 | 5 | 2 | 405 | **95.75** |
| *NUM* | 108 | 2 | 5 | 376 | 96.86 |
| 2 | *ENTY* | 73 | 12 | 21 | 385 | 81.56 | 91.29 |
| *DESC* | 132 | 6 | 6 | 347 | 95.65 |
| *HUM* | 61 | 5 | 4 | 421 | 93.12 |
| *LOC* | 78 | 14 | 3 | 396 | 90.17 |
| *NUM* | 107 | 3 | 6 | 375 | 95.96 |
| 3 | *ENTY* | 63 | 5 | 31 | 392 | 77.77 | 90.12 |
| *DESC* | 137 | 16 | 1 | 337 | 94.15 |
| *HUM* | 60 | 7 | 5 | 419 | 90.90 |
| *LOC* | 79 | 13 | 2 | 397 | 91.32 |
| *NUM* | 108 | 3 | 5 | 375 | 96.42 |
| 5 | *ENTY* | 70 | 7 | 24 | 390 | 81.87 | 91.49 |
| *DESC* | 135 | 15 | 3 | 338 | 93.75 |
| *HUM* | 60 | 6 | 5 | 420 | 91.60 |
| *LOC* | 77 | 7 | 4 | 403 | 93.33 |
| *NUM* | 110 | 4 | 3 | 374 | **96.91** |
| 6 | *ENTY* | 70 | 8 | 24 | 389 | 81.39 | 90.83 |
| *DESC* | 135 | 11 | 3 | 342 | 95.07 |
| *HUM* | 59 | 7 | 6 | 419 | 90.07 |
| *LOC* | 79 | 14 | 2 | 396 | 90.80 |
| *NUM* | 107 | 1 | 6 | 377 | 96.83 |
| 10 | *ENTY* | 76 | 7 | 18 | 390 | 85.87 | 92.96 |
| *DESC* | 134 | 12 | 4 | 341 | 94.36 |
| *HUM* | 61 | 5 | 4 | 421 | 93.12 |
| *LOC* | 79 | 7 | 2 | 403 | 94.61 |
| *NUM* | 108 | 2 | 5 | 376 | 96.86 |
| 15 | *ENTY* | 79 | 10 | 15 | 387 | 86.33 | 93.10 |
| *DESC* | 133 | 10 | 5 | 343 | 94.66 |
| *HUM* | 59 | 1 | 6 | 425 | **94.4** |
| *LOC* | 81 | 10 | 0 | 400 | 94.18 |
| *NUM* | 106 | 2 | 7 | 376 | 95.92 |
| 25 | *ENTY* | 73 | 5 | 21 | 392 | 84.88 | 92.87 |
| *DESC* | 136 | 16 | 2 | 337 | 93.79 |
| *HUM* | 63 | 6 | 2 | 420 | 94.02 |
| *LOC* | 79 | 5 | 2 | 405 | **95.75** |
| *NUM* | 106 | 2 | 7 | 376 | 95.92 |
| 30 | *ENTY* | 73 | 8 | 21 | 389 | 83.42 | 92.40 |
| *DESC* | 134 | 5 | 4 | 348 | 96.75 |
| *HUM* | 59 | 5 | 6 | 421 | 91.47 |
| *LOC* | 80 | 8 | 1 | 402 | 94.67 |
| *NUM* | 111 | 8 | 2 | 370 | 95.68 |
| 50 | *ENTY* | 81 | 11 | 13 | 386 | **87.09** | **93.57** |
| *DESC* | 136 | 5 | 2 | 348 | **97.49** |
| *HUM* | 61 | 6 | 4 | 420 | 92.42 |
| *LOC* | 77 | 5 | 4 | 405 | 94.47 |
| *NUM* | 107 | 2 | 6 | 376 | 96.39 |
| 75 | *ENTY* | 78 | 10 | 16 | 387 | 85.71 | 92.95 |
| *DESC* | 137 | 7 | 1 | 346 | 97.16 |
| *HUM* | 61 | 7 | 4 | 419 | 91.72 |
| *LOC* | 79 | 6 | 2 | 404 | 95.18 |
| *NUM* | 104 | 2 | 9 | 376 | 94.97 |
| 150 | *ENTY* | 76 | 9 | 18 | 388 | 84.91 | 92.77 |
| *DESC* | 134 | 7 | 4 | 346 | 96.05 |
| *HUM* | 63 | 8 | 2 | 418 | 92.64 |
| *LOC* | 75 | 4 | 6 | 406 | 93.75 |
| *NUM* | 110 | 5 | 3 | 373 | 96.49 |

\**ENTY*: entitas, *DESC*: deskripsi, *HUM*: manusia, *LOC*: lokasi, *NUM*: nomor.

Berdasarkan Tabel 5, nilai F1-score tertinggi untuk masing-masing kategori diperoleh oleh model yang berbeda. Model tanpa *attention* dan model dengan jumlah *head* 50 memiliki nilai F1-score tertinggi untuk kategori *ENTY* dan *DESC* dengan nilai 87.09% dan 97.49%. Model dengan jumlah *head* 15 memiliki nilai F1-score tertinggi untuk kategori *HUM* dengan nilai 94.4%. Model dengan jumlah *head* 25 memiliki nilai F1-score tertinggi untuk kategori *LOC* dengan nilai 95.75%. Model dengan jumlah *head* 15 memiliki nilai F1-score tertinggi untuk kategori *NUM* dengan nilai 96.91%. Ini menunjukkan setiap model memiliki kelebihannya masing-masing dalam menggeneralisasi setiap kategori. Terdapat 2 model yang memiliki nilai *mean* F1-score tertinggi dengan nilai yang sama, yakni model tanpa *attention* dan model dengan jumlah *head* 50 dengan nilai 93.57%. Meskipun model tanpa *attention* memiliki akurasi yang lebih rendah, nilai *mean* F1-score-nya setara dengan model dengan jumlah *head* 50. Hal ini kembali membuktikan bahwa pengaruh *head attention* tidak berkontribusi besar pada riset ini. Pada Lampiran 1 diberikan ilustrasi *confusion matrix* untuk semua skenario dan menunjukkan bahwa label *ENTY* sering diinterpretasikan sebagai kelas lain oleh model.

### Perbandingan Fokus pada Head Attention

Pada bagian ini, *output* dari *attention layer* divisualisasikan untuk membandingkan performa antar model serta mengamati bagaimana model menghasilkan prediksi. Model yang dibandingkan mencakup varian dengan 0 *head*, 1 *head*, 3 *head*, dan 50 *head*. Teks input yang digunakan adalah “*What is the depth of the Nile river ?*” (berapa kedalaman sungai Nil?), yang merupakan salah satu sampel dari data *test* berlabel *NUM*. Pada sampel ini, model dengan 1 *head* dan 3 *head* memberikan prediksi yang salah dengan label *LOC* (lihat Tabel 6), sedangkan model 0 *head* dan 50 *head* memberikan prediksi yang benar dengan label *NUM*. Kata *depth* menjadi kata kunci pada pertanyaan tersebut karena identik dengan angka sehingga model yang dapat fokus pada kata tersebut memiliki kemungkinan yang lebih besar untuk dapat memberikan prediksi yang tepat.

Tabel 6. Perbandingan prediksi pada model 0 *head,* 1 *head*, 3 *head*, dan 50 *head*.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Pertanyaan | Label |  | Prediksi model | | |
| 0 *head* | 1 *head* | 3 *head* | 50 *head* |
| *What is the depth of the Nile river ?* | *NUM* | *NUM* | *LOC* | *LOC* | *NUM* |

Model dengan 0 *head* bisa memberikan prediksi yang benar meskipun tidak menggunakan *attention*. Gambar 14 menampilkan *output attention* dari model dengan 1 *head*. Kata *of* tidak terdapat pada *vocabulary* sehingga dinyatakan sebagai token <UNK>. Pada *output* *forward*, model lebih fokus pada kata *river*. Ini dapat dilihat dengan nilai *attention score* dari kata *is*, *the*,dan *depth* yang tinggi terhadap kata *river*. Pada *output backward*, model lebih fokus pada kata *what*, *the*, dan *nile*. Ini dapat dilihat dengan nilai *attention score* dari kata *what*, *is*, dan <UNK> yang tinggi terhadap kata *the* dan *nile* serta kata *nile* yang memiliki nilai *attention score* yang tinggi terhadap kata *what*. Kombinasi *attention score* pada model ini berfokus pada kata *nile* dan *river*, di mana kedua kata tersebut memiliki kaitan erat dengan lokasi sehingga model memprediksi pertanyaan dengan label *LOC*.

A screenshot of a graph

AI-generated content may be incorrect.

Gambar 16. *Output* *attention* model dengan 1 *head*.

Gambar 15 menampilkan *output attention* dari model dengan 3 *head*. *Output* *head* pertama (lihat gambar Gambar 15 (a)) bagian *forward* lebih fokus pada kata *what*. Ini dapat dilihat dengan nilai *attention score* dari kata *is*, *the*, *depth*, <UNK>, *nile*, dan *river* yang tinggi terhadap kata *what*. *Output* *head* pertama bagian *backward* lebih fokus pada kata *the* dan *nile*. Ini dapat dilihat dengan nilai *attention score* dari kata *what*, is, *the*, *depth*, dan *river* yang tinggi terhadap kata *the* dan *nile*. *Output* *head* kedua (lihat gambar Gambar 15 (b)) bagian *forward* lebih fokus pada kata *what*. Ini dapat dilihat dengan nilai *attention score* dari kata *is*, *the*, *depth*, <UNK>, *nile*, dan *river* yang tinggi terhadap kata *what*. *Output* *head* kedua bagian *backward* lebih fokus pada kata *the* dan *nile*. Ini dapat dilihat dengan nilai *attention score* dari kata *what*, is, *the*, *depth*, dan *river* yang tinggi terhadap kata *the* dan *nile*. *Output* *head* ketiga (lihat Gambar 15 (c)) memiliki nilai *attention score* yang mirip dengan *output head kedua*. Perbedaan nilai *attention score* pada *head* pertama model 1 *head* dan model *3 head* disebabkan oleh jumlah dimensi *embedding* yang diproses berbeda. Kombinasi *attention score* pada model ini berfokus pada kata *nile* dan *river*, di mana kedua kata tersebut memiliki kaitan erat dengan lokasi sehingga model memprediksi pertanyaan dengan label *LOC*.

A screenshot of a graph

AI-generated content may be incorrect.

(a)

A screenshot of a graph

AI-generated content may be incorrect.

(b)

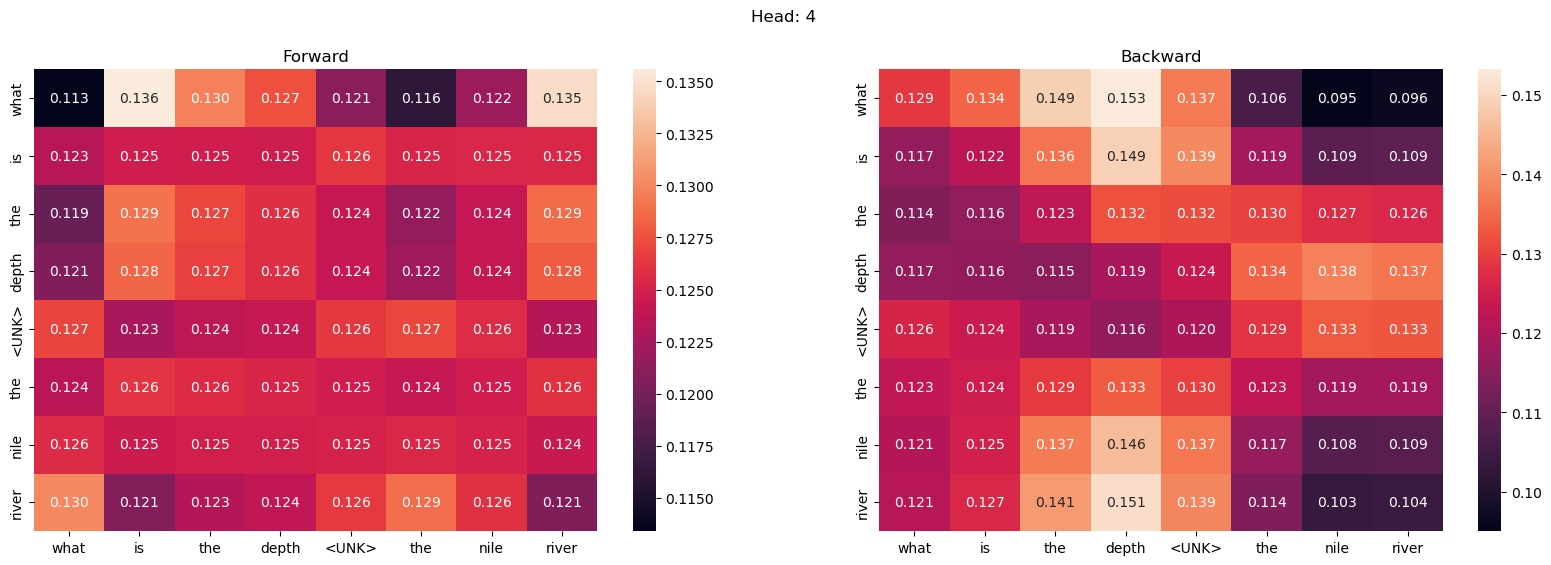
A screenshot of a graph

AI-generated content may be incorrect.

(c)

Gambar 17. *Output attention* model dengan 3 *head* pada (a) *head* 1, (b) *head* 2, dan (c) *head* 3.

Gambar 16 menampilkan *output attention* dari beberapa *head* dari model dengan dengan 50 *head*. *Output* dari model ini hampir mirip dengan model-model sebelumnya, di mana terdapat *head* yang fokus pada kata *nile* dan *river*, seperti pada Gambar 16 (b) dan (c). Namun, terdapat beberapa *head* yang fokus pada kata *depth*, seperti pada Gambar 16 (a), (d), (e), dan (f). Dengan beberapa *head* yang fokus pada kata *depth*, model ini dengan benar memprediksi pertanyaan dengan label *NUM*.



(a)

A screenshot of a graph

AI-generated content may be incorrect.

(b)

A screenshot of a graph

AI-generated content may be incorrect.

(c)

A screenshot of a graph

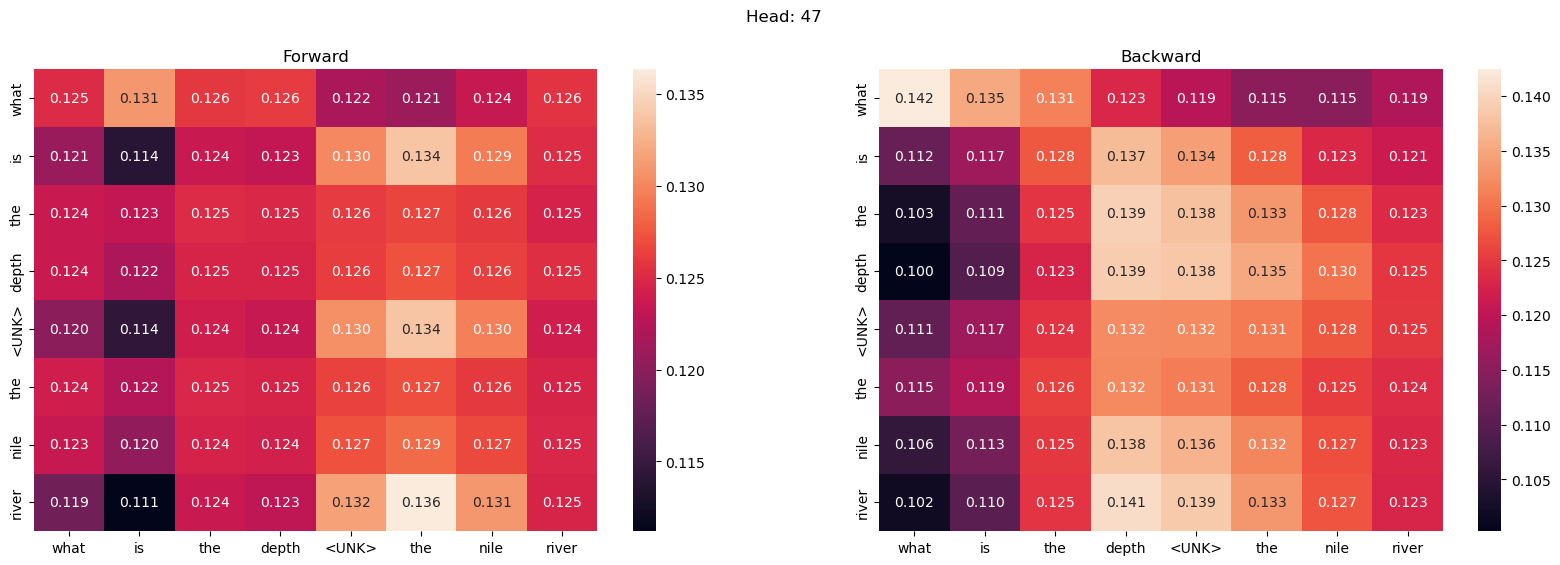
AI-generated content may be incorrect.

(d)

A screenshot of a graph

AI-generated content may be incorrect.

(e)



(f)

Gambar 18. *Output attention* model dengan 50 *head* pada (a) *head* 4, (b) *head* 15, (c) *head* 25, (d) *head* 26, (e) *head* 29, dan (f) *head* 47.

Perbandingan yang dilakukan pada bagian ini kembali membuktikan bahwa pengaruh *head attention* tidak berkontribusi besar pada riset ini.

# KESIMPULAN DAN SARAN

## Kesimpulan

Penelitian ini bertujuan untuk menganalisis pengaruh variasi jumlah *head* pada mekanisme *multi-head attention* dalam model BiLSTM terhadap performa klasifikasi teks, serta membandingkannya dengan model tanpa *attention*. Hasil penelitian menunjukkan bahwa:

* Model *deep learning* berbasis *attention* berhasil diimplementasikan dengan arsitektur BiLSTM yang digabung dengan mekanisme *multi*-*head attention*. Jumlah *head* divariasikan dari 1 hingga 150 dengan ketentuan bahwa setiap nilai merupakan pembagi dari dimensi *embedding attention* sebesar 150.
* Penambahan *head* *attention* tidak selalu memberikan peningkatan akurasi yang signifikan. Model dengan *attention* tidak selalu memiliki performa yang lebih baik dibandingkan dengan model tanpa *attention*. Selain itu, meskipun model dengan 50 *head* menghasilkan akurasi tertinggi (94.09%), perbedaannya hanya 0.82% dibandingkan model tanpa *attention* (93.27%). Ini menunjukkan bahwa dalam konteks data TREC yang relatif pendek per sampel, pengaruh *attention* terbatas. Selain itu, efisiensi waktu pelatihan meningkat dengan adanya *attention*. Model dengan *attention* umumnya mencapai titik konvergensi lebih cepat melalui *early stopping* dibandingkan model tanpa *attention*, menandakan manfaat dalam efisiensi pelatihan meskipun tanpa peningkatan akurasi yang signifikan.

## Saran

Berdasarkan hasil dari penelitian ini, berikut beberapa saran untuk penelitian selanjutnya:

1. Menggunakan data dengan jumlah teks yang panjang untuk menguji sejauh mana *attention* berperan dalam konteks data yang membutuhkan pemahaman jangka panjang.
2. Menguji penggunaan *attention* menggunakan data yang bervariasi dibandingkan menggunakan jumlah *head attention* yang bervariasi.

# DAFTAR PUSTAKA

Albarrak, K., Gulzar, Y., Hamid, Y., Mehmood, A., & Soomro, A. B. (2022). A Deep Learning-Based Model for Date Fruit Classification. *Sustainability (Switzerland)*, *14*(10). https://doi.org/10.3390/su14106339

An, Q., Chen, W., & Shao, W. (2024). A Deep Convolutional Neural Network for Pneumonia Detection in X-ray Images with Attention Ensemble. *Diagnostics*, *14*(4). https://doi.org/10.3390/diagnostics14040390

Devlin, J., Chang, M. W., Lee, K., & Toutanova, K. (2019). BERT: Pre-training of deep bidirectional transformers for language understanding. *NAACL HLT 2019 - 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies - Proceedings of the Conference*, *1*.

Dewi, C., Indriawan, F. A., & Christanto, H. J. (2023). Spam classification problems using support vector machine and grid search. *International Journal of Applied Science and Engineering*, *20*(4). https://doi.org/10.6703/IJASE.202312\_20(4).006

Gagliardi, G. (2024). Natural language processing techniques for studying language in pathological ageing: A scoping review. In *International Journal of Language and Communication Disorders* (Vol. 59, Issue 1). https://doi.org/10.1111/1460-6984.12870

Gao, F., Zhang, J., Zhang, C., Xu, S., & Ma, C. (2023). Long Short-Term Memory Networks with Multiple Variables for Stock Market Prediction. *Neural Processing Letters*, *55*, 4211–4229. https://doi.org/10.1007/s11063-022-11037-8

Gülmez, B. (2023). Stock price prediction with optimized deep LSTM network with artificial rabbits optimization algorithm. *Expert Systems with Applications*, *227*. https://doi.org/10.1016/j.eswa.2023.120346

Gygi, J. P., Kleinstein, S. H., & Guan, L. (2023). Predictive overfitting in immunological applications: Pitfalls and solutions. In *Human Vaccines and Immunotherapeutics* (Vol. 19, Issue 2). https://doi.org/10.1080/21645515.2023.2251830

Hassanin, M., Anwar, S., Radwan, I., Khan, F. S., & Mian, A. (2024). Visual attention methods in deep learning: An in-depth survey. *Information Fusion*, *108*. https://doi.org/10.1016/j.inffus.2024.102417

Herrmann, L., & Kollmannsberger, S. (2024). Deep learning in computational mechanics: a review. In *Computational Mechanics* (Vol. 74, Issue 2). https://doi.org/10.1007/s00466-023-02434-4

Hussein, B. M., & Shareef, S. M. (2024). An Empirical Study on the Correlation between Early Stopping Patience and Epochs in Deep Learning. *ITM Web of Conferences*, *64*, 01003. https://doi.org/10.1051/itmconf/20246401003

Jamshidi, S., Mohammadi, M., Bagheri, S., Najafabadi, H. E., Rezvanian, A., Gheisari, M., Ghaderzadeh, M., Shahabi, A. S., & Wu, Z. (2024). Effective text classification using BERT, MTM LSTM, and DT. *Data and Knowledge Engineering*, *151*. https://doi.org/10.1016/j.datak.2024.102306

Johnson, S. J., Murty, M. R., & Navakanth, I. (2024). A detailed review on word embedding techniques with emphasis on word2vec. *Multimedia Tools and Applications*, *83*(13). https://doi.org/10.1007/s11042-023-17007-z

Khurana, D., Koli, A., Khatter, K., & Singh, S. (2023). Natural language processing: state of the art, current trends and challenges. *Multimedia Tools and Applications*, *82*(3). https://doi.org/10.1007/s11042-022-13428-4

Kowsari, K., Meimandi, K. J., Heidarysafa, M., Mendu, S., Barnes, L., & Brown, D. (2019). Text classification algorithms: A survey. In *Information (Switzerland)* (Vol. 10, Issue 4). https://doi.org/10.3390/info10040150

Li, X., Zhou, S., Wang, F., & Fu, L. (2024). An improved sparrow search algorithm and CNN-BiLSTM neural network for predicting sea level height. *Scientific Reports*, *14*(1). https://doi.org/10.1038/s41598-024-55266-4

Liga, D., & Robaldo, L. (2023). Fine-tuning GPT-3 for legal rule classification. *Computer Law and Security Review*, *51*. https://doi.org/10.1016/j.clsr.2023.105864

Liu, G., & Guo, J. (2019). Bidirectional LSTM with attention mechanism and convolutional layer for text classification. *Neurocomputing*, *337*. https://doi.org/10.1016/j.neucom.2019.01.078

Mikolov, T., Sutskever, I., Chen, K., Corrado, G., & Dean, J. (2013). Distributed representations ofwords and phrases and their compositionality. *Advances in Neural Information Processing Systems*. https://doi.org/https://doi.org/10.48550/arXiv.1310.4546

Minaee, S., Kalchbrenner, N., Cambria, E., Nikzad, N., Chenaghlu, M., & Gao, J. (2021). Deep Learning-Based Text Classification. In *ACM Computing Surveys* (Vol. 54, Issue 3). https://doi.org/10.1145/3439726

Nailah, F., Larasati, D. I., Siswanto, S., & Kalondeng, A. (2024). OPTIMASI METODE JARINGAN SARAF TIRUAN BACKPROPAGATION UNTUK PERAMALAN CURAH HUJAN BULANAN DI KOTA DENPASAR. *MATHunesa: Jurnal Ilmiah Matematika*, *12*(1). https://doi.org/10.26740/mathunesa.v12n1.p134-140

Ogunsanya, M., Isichei, J., & Desai, S. (2023). Grid search hyperparameter tuning in additive manufacturing processes. *Manufacturing Letters*, *35*. https://doi.org/10.1016/j.mfglet.2023.08.056

Papanastasiou, G., Dikaios, N., Huang, J., Wang, C., & Yang, G. (2024). Is Attention all You Need in Medical Image Analysis? A Review. *IEEE Journal of Biomedical and Health Informatics*, *28*(3). https://doi.org/10.1109/JBHI.2023.3348436

Patil, R., Boit, S., Gudivada, V., & Nandigam, J. (2023). A Survey of Text Representation and Embedding Techniques in NLP. *IEEE Access*, *11*. https://doi.org/10.1109/ACCESS.2023.3266377

Raffel, C., Shazeer, N., Roberts, A., Lee, K., Narang, S., Matena, M., Zhou, Y., Li, W., & Liu, P. J. (2020). Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of Machine Learning Research*, *21*.

Rainio, O., Teuho, J., & Klén, R. (2024). Evaluation metrics and statistical tests for machine learning. *Scientific Reports*, *14*(1). https://doi.org/10.1038/s41598-024-56706-x

Raschka, S. (2024). *Build a Large Language Model (From Scratch)* (D. Archibald, D. Caswell, K. Rit, A. Dragosavljevic ́, K. Lucke, & A. Larson, Eds.). Manning.

Reiter, E. (2025). *Natural Language Generation*. https://doi.org/10.1007/978-3-031-68582-8

Salehin, I., & Kang, D. K. (2023). A Review on Dropout Regularization Approaches for Deep Neural Networks within the Scholarly Domain. In *Electronics (Switzerland)* (Vol. 12, Issue 14). https://doi.org/10.3390/electronics12143106

Sari, E., Courville, V., & Nia, V. P. (2022). iRNN: Integer-only Recurrent Neural Network. *International Conference on Pattern Recognition Applications and Methods*, *1*. https://doi.org/10.5220/0010975700003122

Ueslei da Fonseca, A., Lopes Parreira, P., da Silva Vieira, G., Paula Felix, J., Barreto Conte, M., Fouad Rabahi, M., & Soares, F. (2024). A novel tuberculosis diagnosis approach using feed-forward neural networks and binary pattern of phase congruency. *Intelligent Systems with Applications*, *21*. https://doi.org/10.1016/j.iswa.2023.200317

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

Wu, J. M. T., Li, Z., Herencsar, N., Vo, B., & Lin, J. C. W. (2023). A graph-based CNN-LSTM stock price prediction algorithm with leading indicators. *Multimedia Systems*, *29*(3). https://doi.org/10.1007/s00530-021-00758-w

Xia, Z. (2024). Overfitting of CNN model in cifar-10: Problem and solutions. *Applied and Computational Engineering*, *37*(1). https://doi.org/10.54254/2755-2721/37/20230511

Zhang, Y., Liu, C., Liu, M., Liu, T., Lin, H., Huang, C. B., & Ning, L. (2024). Attention is all you need: utilizing attention in AI-enabled drug discovery. *Briefings in Bioinformatics*, *25*(1). https://doi.org/10.1093/bib/bbad467

Zhou, Z., Zhu, J., & Li, W. (2024). Towards Understanding Neural Machine Translation with Attention Heads’ Importance. *Applied Sciences (Switzerland)*, *14*(7). https://doi.org/10.3390/app14072798

# LAMPIRAN

Lampiran 1. *Confusion matrix* setiap model

|  |  |  |
| --- | --- | --- |
| A close-up of a graph  AI-generated content may be incorrect. |  |  |
| (a) | (b) | (c) |
|  |  |  |
| (d) | (e) | (f) |
|  |  |  |
| (g) | (h) | (i) |
|  |  |  |
| (j) | (k) | (l) |
|  |  |  |
|  | (m) |  |

Confusion matrix model dengan jumlah *head* (a) 0, (b) 1, (c) 2, (d) 3, (e) 5, (f) 6, (g) 10, (h) 15, (i) 25, (j) 30, (k) 50, (l) 75, dan (m) 150.

Lampiran 2. Program model

A qr code on a white background

AI-generated content may be incorrect.

Tautan: <https://github.com/mfshobur/bilstm-attention-trec-classification/blob/main/model/alstm.py>

Lampiran 3. Hasil eksperimen *grid search*

A qr code on a white background

AI-generated content may be incorrect.

Tautan: <https://github.com/mfshobur/bilstm-attention-trec-classification/blob/main/model/gridsearch_scenario/gridsearch.csv>

Lampiran 4. Hasil eksperimen variasi jumlah *head* *multi*-*head attention*

A qr code on a white background

AI-generated content may be incorrect.

Tautan: <https://github.com/mfshobur/bilstm-attention-trec-classification/blob/main/model/gridsearch_scenario/n_head.csv>

Lampiran 5. *Source code*

A qr code on a white background

AI-generated content may be incorrect.

Tautan: <https://github.com/mfshobur/bilstm-attention-trec-classification/tree/main/model/gridsearch_scenario>