

KLASIFIKASI SARAN DAN KRITIK PADA SIMAK UNISMUH DENGAN MENGGUNAKAN ALGORITMA RECURRENT NEURAL NETWORK (RNN)

Ahmad Faisal¹, Titin Wahyuni², Fahrir Irhamna Rachman³

^{1,2,3} Informatika, Universitas Muhammadiyah Makassar, Makassar, 90221, Indonesia
105841112420@student.unismuh.ac.id

Abstract

SIMAK Unismuh Makassar is an important platform used by students to submit suggestions and criticisms related to various academic aspects. In this study, researchers implemented the Recurrent Neural Network (RNN) algorithm to classify suggestions and criticisms received through SIMAK Unismuh. The purpose of this study was to determine the implementation of the RNN Algorithm in classifying suggestions and criticisms on the SIMAK Unismuh page and how successful the RNN Algorithm was in classifying suggestions and criticisms on the SIMAK Unismuh page. RNN was chosen because of its ability to process sequential text data, such as input in the form of sentences, which allows the model to capture the context of the input more effectively. The dataset used in this study consists of a number of suggestion and criticism data that have been categorized manually. The RNN model that was built was then trained and tested using the data to assess its accuracy and performance. The results showed that the model achieved the highest accuracy of 91% and the lowest accuracy of 90%. Although there were variations in model performance, these results indicate that RNN has good potential in classifying suggestion and criticism texts. The RNN model can help institutions understand and respond to user input more effectively, although it still requires further optimization to improve the consistency and accuracy of the results. The conclusion of this study shows that the RNN model is able to classify suggestions and criticisms with an adequate level of accuracy. The application of this model is expected to help the Unismuh administration in managing student input more efficiently, as well as providing more appropriate and faster responses to academic needs.

Keywords: Text Classification, Recurrent Neural Network (RNN), SIMAK Unismuh, Suggestions and Criticisms, Academic Information System.

Abstrak

SIMAK Unismuh Makassar merupakan platform penting yang digunakan oleh mahasiswa untuk menyampaikan saran dan kritik terkait berbagai aspek akademik. Dalam penelitian ini, peneliti mengimplementasikan algoritma Recurrent Neural Network (RNN) untuk mengklasifikasikan saran dan kritik yang diterima melalui SIMAK Unismuh. Tujuan dari penelitian ini adalah untuk mengetahui implementasi Algoritma RNN dalam mengklasifikasi saran dan kritik di laman SIMAK Unismuh dan bagaimana keberhasilan Algoritma RNN dalam mengklasifikasi saran dan kritik di laman SIMAK Unismuh. RNN dipilih karena kemampuannya dalam mengolah data teks yang berurutan, seperti masukan dalam bentuk kalimat, yang memungkinkan model untuk menangkap konteks dari masukan tersebut secara lebih efektif. Dataset yang digunakan dalam penelitian ini terdiri dari sejumlah data saran dan kritik yang telah dikategorikan secara manual. Model RNN yang dibangun kemudian dilatih dan diuji menggunakan data tersebut untuk menilai akurasi dan performanya. Hasil penelitian menunjukkan bahwa model RNN mencapai akurasi tertinggi sebesar 91% dan akurasi terendah sebesar 90%. Meskipun terdapat variasi dalam performa model, hasil ini menunjukkan bahwa RNN memiliki potensi yang baik dalam mengklasifikasikan teks saran dan kritik. Model RNN dapat membantu institusi dalam memahami dan merespon masukan dari pengguna dengan lebih efektif, meskipun masih memerlukan optimasi lebih lanjut untuk meningkatkan konsistensi dan akurasi hasil. Kesimpulan dari penelitian ini menunjukkan bahwa model RNN mampu mengklasifikasikan saran dan kritik dengan tingkat akurasi yang memadai. Penerapan model ini diharapkan dapat membantu pihak administrasi Unismuh dalam mengelola masukan dari mahasiswa secara lebih efisien, serta memberikan respons yang lebih tepat dan cepat terhadap kebutuhan akademik.

Kata Kunci: Klasifikasi Teks, Recurrent Neural Network (RNN), SIMAK Unismuh, Saran dan Kritik, Sistem Informasi Akademik

1. Pendahuluan

Perkembangan teknologi informasi yang semakin pesat di era globalisasi saat ini tidak bisa dihindari lagi pengaruhnya terhadap dunia pendidikan. Dalam konteks pembelajaran dan pengembangan institusi pendidikan seperti universitas semakin mengedepankan teknologi guna meningkatkan pengalaman akademik hingga sistem administrative yang dapat ditawarkan kepada mahasiswa[1].

Pada tiap langkah waktu, proses dari RNN tidak hanya berupa data masukan asal, tapi juga nilai dari lapis tersembunyi yang terdapat pada langkah waktu sebelumnya sebuah metode untuk melakukan klasifikasi terhadap objek berdasarkan data pembelajaran yang jaraknya paling dekat dengan objek tersebut[2]. Berdasarkan penelitian sebelumnya dengan judul "Klasifikasi Kalimat Ilmiah Menggunakan "RNN" Penelitian ini telah berhasil membuat model komputasi klasifikasi kalimat menggunakan RNN, dengan fitur yang telah diekstraksi menggunakan fungsi Word2Vec untuk menghasilkan satu set vektor[3].

2. Metode Penelitian

2.1 Simak

Sistem manajemen akademik atau yang dikenal dengan SIMAK adalah aplikasi perangkat lunak yang berfungsi untuk menyajikan informasi dan mengelolah administrasi terkait kegiatan akademik. Penggunaan perangkat lunak semacam ini diharapkan dapat meningkatkan efisiensi pengolahan akademik dan memudahkan akses terhadap informasi yang dibutuhkan dengan cepat dan mudah[4].

Sistem Informasi Manajemen Akademik adalah segala macam hasil interaksi antara elemen di lingkungan akademik untuk menghasilkan informasi yang kemudian dijadikan landasan pengambilan sebuah keputusan melaksanakan tindakan, baik oleh pelaku proses itu sendiri maupun dari pihak luar kampus[5]. Merujuk PP Nomor 60 Tahun 1999 pasal 9 ayat 1 administrasi akademik pendidikan tinggi diselenggarakan dengan menerapkan sistem kredit semester, ayat 2 pelaksanaan ketentuan sebagaimana dimaksud pada ayat 1 diatur oleh Menteri[6].

2.2 Kritik dan Saran

Kritik berasal dari kata kritein yang artinya hakekat, esensi atau substansi. Yaitu inti daripada sebuah persoalan. Sesungguhnya semua kritik pasti melihat sisi negatif. Tidak ada kritik melihat sisi positif. Tujuan semua kritik sama, supaya sesuatu yang negatif bisa menjadi positif. Masalahnya adalah, banyak orang tidak memahami hakekat daripada kritik itu sendiri[7]. Saran adalah sebuah solusi yang ditujukan untuk menyelesaikan permasalahan yang dihadapi. Saran harus bersifat membangun, mendidik, dan secara objektif dan sesuai dengan topik yang dibahasn[8].

2.3 Recurrent Neural Network (RNN)

RNN merupakan salah satu jenis algoritma pembelajaran mesin yang berbasis pengolahan informasi dari sebuah data dan merupakan algoritma deep learning. RNN menyimpan pola-pola informasi masa lalu dengan melakukan perulangan di dalam arsitekturnya agar dapat membuat informasi tersebut tetap tersimpan. RNN merupakan pengembangan dari Jaringan Saraf Tiruan (JST) dan arsitekturnya mirip dengan Multilayer Perceptron (MLP)[9]. Hidden layer sebelumnya menyimpan informasi dari proses ekstraksi fitur di awal sehingga saat hidden layer dijadikan input untuk proses selanjutnya itu menyimpan memory (ingatan) dari input sebelumnya[10].

Hidden layer sebelumnya menyimpan informasi dari proses ekstraksi fitur di awal sehingga saat hidden layer dijadikan input untuk proses selanjutnya itu menyimpan memory (ingatan) dari input sebelumnya[11].

2.4 Supervised Learning

Dengan menggunakan algoritma-algoritma supervised learning, model dapat digunakan untuk mengklasifikasikan data ke dalam kategori yang tepat atau untuk melakukan prediksi numerik[12].

Dalam algoritma supervised learning, sistem akan diberikan data pelatihan yang berisi informasi input dan output yang diinginkan, sehingga memungkinkan sistem untuk mempelajari berdasarkan data yang telah ada. Sistem akan mencari pola dalam dataset tersebut, kemudian pola tersebut akan dijadikan sebagai acuan untuk kumpulan data yang ada[13].

2.5 NLP

NLP adalah cabang ilmu komputer yang terkait dengan kecerdasan buatan yang

mempelajari interaksi antara bahasa manusia dan komputer. Dalam ruang lingkup yang lebih luas, NLP mencakup segala bentuk manipulasi komputer terhadap bahasa alami[14].

Dalam tugas NLP juga melibatkan proses penguraian kata atau Bahasa menjadi komponen yang lebih sederhana. Ini melibatkan pemahaman hubungan antar komponen tersebut dan penelusuran cara komponen – komponen tersebut bekerja secara bersamaan untuk membentuk makna yang utuh[15].

3. Hasil Dan Pembahasan

3.1 Pengambilan Data

Tahap pengumpulan data yang digunakan dalam penelitian ini diperoleh dari Universitas SIMAK Muhammadiyah Makassar. SIMAK merupakan sumber data resmi yang mencatat informasi akademik mahasiswa, antara lain riwayat akademik, kualifikasi mata kuliah, dan status akademik lainnya. Berikut adalah hasil pengumpulan data ulasan yang telah disimpan dalam Excel :

Tabel 1. Dataset Ulasan

ULASAN	LABEL
Kedepannya kami bisa diberikan dosen yang lebih produktif dan mampu bekerja sama dengan mahasiswa menyelesaikan permasalahan dalam pertemuan covid ini. Tapi untuk dosen saat ini sudah luar biasa kok.	Kritik
Saya berharap pelayanan akademik mampu memberikan pelayanan yang baik kepada mahasiswa dan seluruh masyarakat Unismuh. Terutama untuk para bapak/ibu dosen sekiranya mampu memberikan metode yang baik bagi mahasiswa sehingga mahasiswa lebih aktif dan produktif dalam proses pembelajaran.	Kritik
Memperbaiki yang baik menjadi lebih baik lagi.	Saran

3.2 Pelabelan Data

Pelabelan sentimen dilakukan secara manual untuk mengidentifikasi pola dan karakteristik dalam teks yang menunjukkan sentimen Saran dan kritik. Ulasan yang diperoleh SIMAK UNISMUH disimpan dalam atribut ulasan, sedangkan nilai klasifikasi saran dan kritik disimpan dalam atribut label. Berikut tabel pelabelan data yang dapat dilihat di bawah ini:

Tabel 2. Tahap Pelabelan Data

ULASAN	LABEL
proses perkuliahan sangat baik dan disiplin.	Kritik
Semoga ruangan perkuliahan bisa memiliki pendingin ruangan (AC)	Saran

3.3 Preprocessing

Preprocessing data adalah tahap di mana data mentah disiapkan dan dibersihkan sebelum dianalisis. Proses ini melibatkan beberapa langkah untuk memastikan data siap digunakan. Berikut tahapan-tahapan dalam preprocessing data:

a. Cleaning

Tahap cleaning data melibatkan proses menghapus tanda baca seperti koma, titik, tanda tanya, tanda seru, bintang, dan pagar dari teks atau data.

Tabel 3. Tahap Cleaning

Sebelum	Sesudah
Sangat komunikatif dengan mahasiswa	Sangat komunikatif dengan mahasiswa
Sudah baik , terimakasih atas ilmu yang telah diberikan	Sudah baik terimakasih atas ilmu yang telah diberikan

Walaupun yang masuk diklas asdos tetapi cara mengajarnya sangat baik, terimakasih atas ilmu yang telah diberikan	Walaupun yang masuk diklas asdos tetapi cara mengajarnya sangat baik terimakasih atas ilmu yang telah diberikan
--	---

b. Tokenizing

Tokenizing adalah langkah dalam pemrosesan teks yang melibatkan pemecahan teks menjadi unit-unit lebih kecil yang disebut token. Unit-unit ini dapat berupa kata-kata, frasa, atau entitas lainnya.

Tabel 4. Tahap Tokenizing

Sebelum	Sesudah
maksimalkan pertemuan dalam perkuliahan	['maksimalkan', 'pertemuan', 'dalam', 'perkuliahan']
sangat komunikatif dengan mahasiswa	['sangat', 'komunikatif', 'dengan', 'mahasiswa']
kedepannya bisa lebih ditingkatkan lagi	['kedepannya', 'bisa', 'lebih', 'ditingkatkan', 'lagi']

3.4 Model Recurrent Neural Network (RNN)

Recurrent Neural Networks (RNN) adalah jenis jaringan saraf yang sangat efektif untuk memproses data urutan atau data sekuensial, seperti teks, audio, video, dan data waktu lainnya. RNN memiliki neuron yang berulang, yang artinya keluaran dari satu langkah waktu diberikan kembali ke neuron sebagai masukan untuk langkah waktu berikutnya.

```
import pandas as pd
import numpy as np
from tqdm import tqdm
from keras.preprocessing.text import Tokenizer
tqdm.pandas(desc="progress-bar")
from gensim.models import Doc2Vec
from sklearn import utils
from sklearn.model_selection import train_test_split
from keras.preprocessing.sequence import pad_sequences
import gensim
from sklearn.linear_model import LogisticRegression
from gensim.models.doc2vec import TaggedDocument
import re
import seaborn as sns
import matplotlib.pyplot as plt

df = pd.read_excel('/content/drive/MyDrive/Skripsi/2020/Ahmad Faisal
RNN/saran.xlsx', sheet_name="Sheet1")
# Replace 'path_to_your_excel_file.xlsx' with your actual file path
df = df[['ULASAN',
'LABEL']]
# Selecting relevant columns
df = df[pd.notnull(df['ULASAN'])]
# Dropping rows with null 'ULASAN' values
df.rename(columns={'ULASAN': 'ULASAN'}, inplace=True)
# Rename 'ULASAN' to 'ULASAN' for consistency
# Mengatur ulang indeks baris
df.index = range(len(df))
# Menghitung total jumlah kata dalam kolom 'ULASAN'
total_words = df['ULASAN'].apply(lambda x: len(x.split(' '))).sum()
print("Total jumlah kata dalam semua ulasan:", total_words)
# Menghitung jumlah kemunculan setiap nilai dalam kolom 'LABEL'
cnt_pro = df['LABEL'].value_counts()
# Menggambar diagram batang menggunakan Seaborn
```

```

plt.figure(figsize=(12, 4))
sns.barplot(x=cnt_pro.index, y=cnt_pro.values, alpha=0.8)
plt.ylabel('Jumlah Kemunculan', fontsize=12)
plt.xlabel('LABEL', fontsize=12)
plt.xticks(rotation=90)
plt.show()

def print_message(index):
    example = df.iloc[index][['ULASAN', 'LABEL']].values
    if len(example) > 0:
        print('ULASAN:', example[0])
        print('LABEL:', example[1])
    # Menggunakan fungsi print_message() dengan indeks tertentu
    print_message(12)
import string
def remove_punctuation(text):
    return text.translate(str.maketrans("", "", string.punctuation))
# Menghapus tanda baca dari kolom ULASAN
df['ULASAN'] = df['ULASAN'].apply(remove_punctuation)
import nltk
# Download the 'punkt' resource
nltk.download('punkt')
# Tokenisasi teks menggunakan nltk
def tokenize_text(text):
    tokens = []
    for sent in nltk.sent_tokenize(text):
        for word in nltk.word_tokenize(sent):
            if len(word) <= 0:
                continue
            tokens.append(word.lower())
    return tokens
# Memisahkan data menjadi train dan test
train, test = train_test_split(df, test_size=0.01, random_state=42)
# TaggedDocument untuk train dan test set
train_tagged = train.apply(
    lambda r: TaggedDocument(words=tokenize_text(r['ULASAN']), tags=[r.LABEL]),
    axis=1)
test_tagged = test.apply(
    lambda r: TaggedDocument(words=tokenize_text(r['ULASAN']), tags=[r.LABEL]),
    axis=1)
# Pengaturan tokenizer
max_features = 500000 # Jumlah maksimum kata yang akan digunakan
max_sequence_length = 50 # Panjang maksimum setiap teks
tokenizer = Tokenizer(num_words=max_features, split=' ', filters='!"#$%&()*+,-./:;<=>?@[\\^_`{}~', lower=True)
tokenizer.fit_on_texts(df['ULASAN'].values)
# Konversi teks ke dalam urutan angka (sequences)
X_train = tokenizer.texts_to_sequences(train['ULASAN'].values)
X_train = pad_sequences(X_train, maxlen=max_sequence_length)
X_test = tokenizer.texts_to_sequences(test['ULASAN'].values)
X_test = pad_sequences(X_test, maxlen=max_sequence_length)
print('Found %s unique tokens.' % len(tokenizer.word_index))
# Konversi teks ke dalam urutan angka (sequences)
X = tokenizer.texts_to_sequences(df['ULASAN'].values)
X = pad_sequences(X, maxlen=max_sequence_length)
print('Shape dari data tensor:', X.shape)
#train_tagged.values
train_tagged.values

```

```

from gensim.models.doc2vec import Doc2Vec, TaggedDocument
# Ubah ukuran vektor (vector_size) sesuai kebutuhan Anda
vector_size = 20
# Inisialisasi model Doc2Vec
d2v_model = Doc2Vec(dm=1, dm_mean=1, vector_size=vector_size, window=8,
min_count=1, workers=1, alpha=0.065, min_alpha=0.065)
# Membangun kosakata dari tagged documents pada data pelatihan
train_tagged = [TaggedDocument(words=tokenize_text(row['ULASAN']),
tags=[row['LABEL']]) for index, row in train.iterrows()]
d2v_model.build_vocab(train_tagged)
# Bangun vocab dari train_tagged
d2v_model.build_vocab(train_tagged)
# Latih model
for epoch in range(30):
    d2v_model.train(utils.shuffle(train_tagged), total_examples=len(train_tagged),
epochs=1)
    d2v_model.alpha -= 0.002 # Reduksi alpha setiap epoch
    d2v_model.min_alpha = d2v_model.alpha # Tetapkan min_alpha sesuai alpha saat
ini
# Mendapatkan jumlah kata dalam kosakata
num_words = len(d2v_model.wv.key_to_index)
print("Jumlah kata dalam kosakata:", num_words)
# Mengakses kata-kata dalam kosakata
words_in_vocab = list(d2v_model.wv.key_to_index.keys())
print("Kata-kata dalam kosakata:", words_in_vocab)
# Inisialisasi matriks embedding kosong
embedding_matrix = np.zeros((len(d2v_model.dv.vectors), d2v_model.vector_size))
# Mengisi matriks embedding dengan vektor-vektor dokumen dari model Doc2Vec
for i in range(len(d2v_model.dv.vectors)):
    embedding_matrix[i] = d2v_model.dv.vectors[i]
# Contoh penggunaan matriks embedding
print("Ukuran matriks embedding:", embedding_matrix.shape)
print("Contoh vektor untuk dokumen pertama:", embedding_matrix[0])

```

Cara kerja RNN dalam penelitian ini adalah dengan memproses teks saran dan kritik secara berurutan, mengingat konteks kata-kata sebelumnya untuk memahami makna kalimat secara keseluruhan, dan kemudian mengklasifikasikan teks tersebut ke dalam kategori yang sesuai berdasarkan pelatihan yang telah dilakukan. Model ini dapat membantu mengotomatisasi dan meningkatkan efisiensi dalam pengelolaan masukan pengguna di sistem SIMAK UNISMUH.

3.5 Hasil Pengujian Metode RNN

Dengan langkah-langkah ini, hasil prediksi pada data validasi telah ditampilkan dan disimpan ke dalam file Excel untuk dianalisis lebih lanjut atau dipresentasikan kepada pihak terkait.

- a. Pembagian Data
 - 1) 90 : 10
 - a) Epoch

```

Epoch 1/15
420/420 - 6s - loss: 0.5924 - acc: 0.7060 - val_loss: 0.4212 - val_acc: 0.8271 - 6s/epoch - 13ms/step
Epoch 2/15
420/420 - 4s - loss: 0.3486 - acc: 0.8714 - val_loss: 0.2550 - val_acc: 0.9142 - 4s/epoch - 10ms/step
Epoch 3/15
420/420 - 4s - loss: 0.2451 - acc: 0.9200 - val_loss: 0.2555 - val_acc: 0.9196 - 4s/epoch - 11ms/step
Epoch 4/15
420/420 - 4s - loss: 0.1971 - acc: 0.9380 - val_loss: 0.2453 - val_acc: 0.9169 - 4s/epoch - 10ms/step
Epoch 5/15
420/420 - 4s - loss: 0.1905 - acc: 0.9387 - val_loss: 0.2530 - val_acc: 0.9196 - 4s/epoch - 11ms/step
Epoch 6/15
420/420 - 4s - loss: 0.1547 - acc: 0.9537 - val_loss: 0.2493 - val_acc: 0.9236 - 4s/epoch - 10ms/step
Epoch 7/15
420/420 - 4s - loss: 0.1322 - acc: 0.9590 - val_loss: 0.2496 - val_acc: 0.9196 - 4s/epoch - 10ms/step
Epoch 8/15
420/420 - 4s - loss: 0.1161 - acc: 0.9620 - val_loss: 0.2599 - val_acc: 0.9129 - 4s/epoch - 10ms/step
Epoch 9/15
420/420 - 4s - loss: 0.1153 - acc: 0.9642 - val_loss: 0.2723 - val_acc: 0.9008 - 4s/epoch - 10ms/step
Epoch 10/15
420/420 - 4s - loss: 0.1125 - acc: 0.9621 - val_loss: 0.2689 - val_acc: 0.9209 - 4s/epoch - 10ms/step
Epoch 11/15
420/420 - 4s - loss: 0.0837 - acc: 0.9753 - val_loss: 0.2871 - val_acc: 0.9196 - 4s/epoch - 10ms/step
Epoch 12/15
420/420 - 4s - loss: 0.0779 - acc: 0.9735 - val_loss: 0.3067 - val_acc: 0.9209 - 4s/epoch - 10ms/step
Epoch 13/15
420/420 - 4s - loss: 0.0745 - acc: 0.9770 - val_loss: 0.3570 - val_acc: 0.9102 - 4s/epoch - 10ms/step
Epoch 14/15
420/420 - 4s - loss: 0.0590 - acc: 0.9830 - val_loss: 0.3184 - val_acc: 0.9182 - 4s/epoch - 10ms/step
Epoch 15/15
420/420 - 4s - loss: 0.0591 - acc: 0.9812 - val_loss: 0.3531 - val_acc: 0.9102 - 4s/epoch - 10ms/step
dict_keys(['loss', 'acc', 'val_loss', 'val_acc'])
validation loss: [0.421220064163208, 0.25503015518188477, 0.2554517388343811, 0.2452619969844818, 0.2526

```

Gambar 1. Epoch 1 – 15

Dari gambar di atas dapat dilihat bahwa pada pembagian data 90:10 dengan batch size 16, Epoch pertama (1), untuk Training Accuracy mencapai 70% dan Validasi Accuracy 82%.

b) Metrik

```

{x} 47/47 [=====] - 0s 4ms/step
F1 Score: 0.9042476634055933
Precision: 0.90438188389531
Recall: 0.9041554959785523

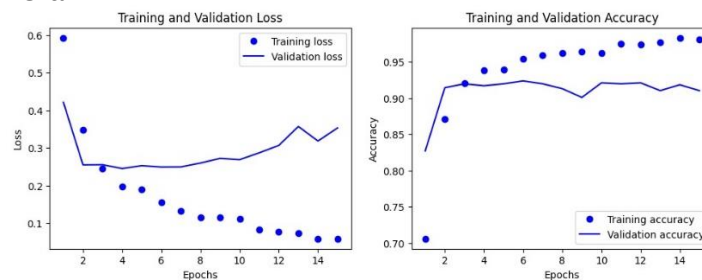
```

	precision	recall	f1-score	support
0	0.85	0.83	0.84	446
1	0.93	0.93	0.93	1046
micro avg	0.90	0.90	0.90	1492
macro avg	0.89	0.88	0.89	1492
weighted avg	0.90	0.90	0.90	1492
samples avg	0.90	0.90	0.90	1492

Gambar 2 . Metrik 0.1

Gambar tersebut menunjukkan hasil evaluasi model klasifikasi dalam bentuk metrik kinerja. Tabel memperlihatkan nilai precision, recall, dan f1-score untuk dua kelas: 0 dan 1. Kelas 0 memiliki precision 0.85, recall 0.83, dan f1-score 0.84 dengan support 446, sedangkan kelas 1 memiliki precision 0.93, recall 0.93, dan f1-score 0.93 dengan support 1046.

c) Grafik



Gambar 3. grafik 0.1

Gambar di atas menunjukkan grafik yang menggambarkan perubahan nilai loss dan akurasi selama proses pelatihan dan validasi model machine learning.

d) Hasil Prediksi

Tabel 5. Hasil Prediksi 0.1

Ulasan	Label Sebenarnya	Prediksi	
sebaiknya materi yang akan di pelajari terlebih dahulu dijelaskan agar mahasiswa dapat lebih mudah memahaminya	Saran	Saran	1
Mahasiswa sering tidak tahu bagaimana cara mengajukan keluhan tentang penilaian	Kritik	Kritik	1

- 2) 80 : 20
a) Epoch

```
Epoch 1/15
373/373 - 6s - loss: 0.6218 - acc: 0.6950 - val_loss: 0.5278 - val_acc: 0.7218 - 6s/epoch - 15ms/step
Epoch 2/15
373/373 - 4s - loss: 0.3904 - acc: 0.8397 - val_loss: 0.3043 - val_acc: 0.8975 - 4s/epoch - 10ms/step
Epoch 3/15
373/373 - 4s - loss: 0.2627 - acc: 0.9130 - val_loss: 0.2641 - val_acc: 0.9048 - 4s/epoch - 11ms/step
Epoch 4/15
373/373 - 4s - loss: 0.2191 - acc: 0.9274 - val_loss: 0.2472 - val_acc: 0.9202 - 4s/epoch - 10ms/step
Epoch 5/15
373/373 - 4s - loss: 0.1803 - acc: 0.9418 - val_loss: 0.2515 - val_acc: 0.9202 - 4s/epoch - 10ms/step
Epoch 6/15
373/373 - 4s - loss: 0.1542 - acc: 0.9524 - val_loss: 0.2585 - val_acc: 0.9155 - 4s/epoch - 11ms/step
Epoch 7/15
373/373 - 4s - loss: 0.1274 - acc: 0.9584 - val_loss: 0.2790 - val_acc: 0.9102 - 4s/epoch - 11ms/step
Epoch 8/15
373/373 - 4s - loss: 0.1172 - acc: 0.9656 - val_loss: 0.2805 - val_acc: 0.9209 - 4s/epoch - 11ms/step
Epoch 9/15
373/373 - 4s - loss: 0.0973 - acc: 0.9703 - val_loss: 0.3050 - val_acc: 0.9169 - 4s/epoch - 11ms/step
Epoch 10/15
373/373 - 4s - loss: 0.0788 - acc: 0.9750 - val_loss: 0.3285 - val_acc: 0.8968 - 4s/epoch - 11ms/step
Epoch 11/15
373/373 - 4s - loss: 0.0880 - acc: 0.9728 - val_loss: 0.3038 - val_acc: 0.9068 - 4s/epoch - 11ms/step
Epoch 12/15
373/373 - 4s - loss: 0.0580 - acc: 0.9832 - val_loss: 0.3260 - val_acc: 0.9048 - 4s/epoch - 11ms/step
Epoch 13/15
373/373 - 4s - loss: 0.0516 - acc: 0.9844 - val_loss: 0.4559 - val_acc: 0.8861 - 4s/epoch - 11ms/step
Epoch 14/15
373/373 - 4s - loss: 0.0533 - acc: 0.9847 - val_loss: 0.3307 - val_acc: 0.9142 - 4s/epoch - 11ms/step
Epoch 15/15
373/373 - 4s - loss: 0.0461 - acc: 0.9851 - val_loss: 0.3723 - val_acc: 0.9062 - 4s/epoch - 11ms/step
dict keys(['loss', 'acc', 'val_loss', 'val_acc'])
validation loss: [0.5278254151344299, 0.304330050945282, 0.26413100957870483, 0.24721147119998932, 0.2514713443111111, 0.2585111111111111, 0.2790111111111111, 0.2805111111111111, 0.3050111111111111, 0.3285111111111111, 0.3038111111111111, 0.3260111111111111, 0.4559111111111111, 0.3307111111111111, 0.3723111111111111]
```

Gambar 4. Epoch 1 – 15 0.2

Dari gambar di atas dapat dilihat bahwa pada pembagian data 80:20 dengan batch size 16, Epoch pertama (1), untuk Training Accuracy mencapai 69% dan Validasi Accuracy 72%.

- b) Metrik

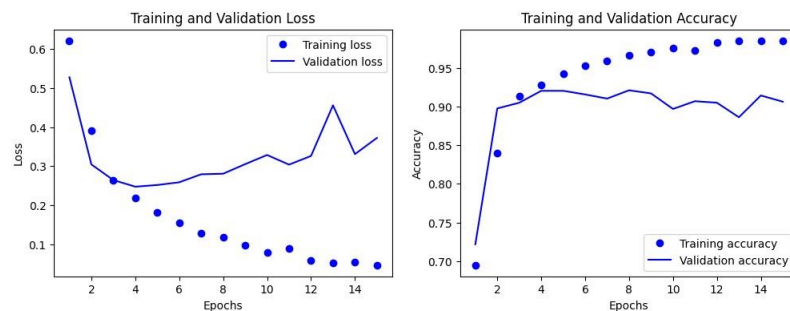
```
47/47 [=====] - 0s 4ms/step
F1 Score: 0.9042476634055933
Precision: 0.90438188389531
Recall: 0.9041554959785523
```

	precision	recall	f1-score	support
0	0.85	0.83	0.84	446
1	0.93	0.93	0.93	1046
micro avg	0.90	0.90	0.90	1492
macro avg	0.89	0.88	0.89	1492
weighted avg	0.90	0.90	0.90	1492
samples avg	0.90	0.90	0.90	1492

Gambar 5. Metrik 0.2

Gambar tersebut menunjukkan hasil evaluasi model klasifikasi dalam bentuk metrik kinerja. Tabel memperlihatkan nilai precision, recall, dan f1-score untuk dua kelas: 0 dan 1. Kelas 0 memiliki precision 0.85, recall 0.83, dan f1-score 0.84 dengan support 446, sedangkan kelas 1 memiliki precision 0.93, recall 0.93, dan f1-score 0.93 dengan support 1046.

- c) Grafik



Gambar 6. Grafik 0.2

Gambar tersebut menunjukkan dua grafik yang membandingkan metrik pelatihan dan validasi model pembelajaran mesin selama beberapa epoch.

d) Hasil Prediksi

Tabel 6. Hasil Prediksi 0.2

Ulasan	Label Sebenarnya	Prediksi	
Sudah bagus pembawaan materinya tetap di pertahankan saja dan kalau bisa dalam proses pembelajaran suasananya dosennya bisa membuat suasananya tidak terlalu tegang sehingga mahasiswa juga senang dalam menerima materi	Saran	Kritik	0
lebih baik di adakan perkuliahan tatap muka	Saran	Saran	1
Jam kuliah yang singkat sering sekali terasa kurang	Kritik	Kritik	1

Gambar di atas menunjukkan daftar komentar atau umpan balik dari mahasiswa mengenai berbagai aspek kampus mereka, seperti proses pembelajaran, fasilitas, dan pelayanan. Kolom A berisi teks komentar, kolom B dan C berisi label 'kritik' atau 'saran', dan kolom D berisi angka biner (0 atau 1) yang mungkin menunjukkan klasifikasi atau kategori dari komentar tersebut.

4. Kesimpulan

Penelitian ini berhasil mengimplementasikan Metode Recurrent Neural Network (RNN) untuk klasifikasi saran dan kritik pada SIMAK UNISMUH dengan menggunakan 7.457 data. Hasil evaluasi menunjukkan bahwa model mencapai akurasi tertinggi sebesar 91% dan akurasi terendah sebesar 90%. Meskipun terdapat variasi dalam performa model, hasil ini menunjukkan bahwa RNN memiliki potensi yang baik dalam mengklasifikasikan teks saran dan kritik. Model RNN dapat membantu institusi dalam memahami dan merespon masukan dari pengguna dengan lebih efektif, meskipun masih memerlukan optimasi lebih lanjut untuk meningkatkan konsistensi dan akurasi hasil. Untuk meningkatkan performa model, disarankan untuk melakukan pengembangan lebih lanjut dengan mencoba berbagai arsitektur RNN lainnya seperti LSTM atau GRU, serta melakukan optimasi hyperparameter dan penambahan data pelatihan.

Referensi

- [1] Yusny, R., & Yasa, G. I. (2019). MENGEMBANGKAN (PEMBELAJARAN) BLENDED LEARNING DENGAN SISTEM LINGKUNGAN PEMBELAJARAN VIRTUAL (VLE) DI PTKIN. *Jurnal Ilmiah Islam Futura*, 19(1). <https://doi.org/10.22373/jiif.v19i1.3707>
- [2] Rozi, I. F., Wijyaningrum, V. N., & Khozin, N. (2020). KLASIFIKASI TEKS LAPORAN MASYARAKAT PADA SITUS LAPOR! MENGGUNAKAN RECURRENT NEURAL NETWORK. *SISTEMASI*, 9(3). <https://doi.org/10.32520/stmsi.v9i3.977>
- [3] Firmansyah, M. R., Ilyas, R., & Kasyidi, F. (2020). Klasifikasi Kalimat Ilmiah Menggunakan Recurrent Neural Network. *Prosiding The 11th Industrial Research Workshop and National Seminar*, 11(1).
- [4] Ardiansyah, D., Saepudin, A., Aryanti, R., & Fitriani, E. (2020). Perancangan Sistem Informasi Akademik Madrasah Aliyah Negeri (Man) 4 Karawang Berbasis Web. *Jurnal Teknologi Dan Open Source*, 3(2), 187–201. <https://doi.org/10.36378/jtos.v3i2.783>
- [5] Agustiandra, V., & Sabandi, A. (2019). Persepsi Guru Terhadap Penerapan Sistem Informasi Manajemen Di Sekolah Menengah Kejuruan (Smk) Negeri 3 Padang. *Jurnal Bahana Manajemen Pendidikan*, 8(1), 1. <https://doi.org/10.24036/bmp.v8i1.103704>

- [6] Map, J., Ilyasi, A., Susanto, H., & Astuti, S. J. W. (2020). PELAKSANAAN SISTEM ADMINISTRASI AKADEMIK DI UNIVERSITAS ISLAM JEMBER. MAP (Jurnal Manajemen Dan Administrasi Publik), 3(2). <https://doi.org/10.37504/map.v3i2.245>
- [7] Ardiansyah, D., Saepudin, A., Aryanti, R., & Fitriani, E. (2020). Perancangan Sistem Informasi Akademik Madrasah Aliyah Negeri (Man) 4 Karawang Berbasis Web. Jurnal Teknologi Dan Open Source, 3(2), 187–201. <https://doi.org/10.36378/jtos.v3i2.783>
- [8] Adanson, Y. N., & Fitriana, G. F. (2022). Rancang Bangun Aplikasi Midesk Berbasis Website Menggunakan Metode User Centered Design. Jurnal Sistem Komputer Dan Informatika (JSON), 4(1). <https://doi.org/10.30865/json.v4i1.4376>
- [9] Nurashila, S. S., Hamami, F., & Kusumasari, T. F. (2023). Perbandingan Kinerja Algoritma Recurrent Neural Network (Rnn) Dan Long Short-Term Memory (Lstm): Studi Kasus Prediksi Kemacetan Lalu Lintas Jaringan Pt Xyz. JIPI (Jurnal Ilmiah Penelitian Dan Pembelajaran Informatika), 8(3), 864–877. <https://doi.org/10.29100/jipi.v8i3.3961>
- [10] Bai, Y., Xie, J., Liu, C., Tao, Y., Zeng, B., & Li, C. (2021). Regression modeling for enterprise electricity consumption: A comparison of recurrent neural network and its variants. International Journal of Electrical Power and Energy Systems, 126. <https://doi.org/10.1016/j.ijepes.2020.106612>
- [11] Bai, Y., Xie, J., Liu, C., Tao, Y., Zeng, B., & Li, C. (2021). Regression modeling for enterprise electricity consumption: A comparison of recurrent neural network and its variants. International Journal of Electrical Power and Energy Systems, 126. <https://doi.org/10.1016/j.ijepes.2020.106612>
- [12] Mestika, J. C., Selan, M. O., & Qadafi, M. I. (2022). Menjelajahi Teknik-Teknik Supervised Learning untuk Pemodelan Prediktif Menggunakan Python. BIIKMA : Buletin Ilmiah Ilmu Komputer Dan Multimedia, 99(99).
- [13] Abijono, H., Santoso, P., & Anggreini, N. L. (2021). Algoritma Supervised Learning Dan Unsupervised Learning Dalam Pengolahan Data. Jurnal Teknologi Terapan: G-Tech, 4(2), 315–318. <https://doi.org/10.33379/gtech.v4i2.635>
- [14] Fajar Ramadhan, D., Noertjahjono, S., & Dedy Irawan, J. (2020). PENERAPAN CHATBOT AUTO REPLY PADA WHATSAPP SEBAGAI PUSAT INFORMASI PRAKTIKUM MENGGUNAKAN ARTIFICIAL INTELLIGENCE MARKUP LANGUAGE. JATI (Jurnal Mahasiswa Teknik Informatika), 4(1). <https://doi.org/10.36040/jati.v4i1.2375>
- [15] Husamuddin, H., Prasetyo, D. B., & Rustamadji, H. C. (2020). Otomatisasi Layanan Frequently Ask Questions Berbasis Natural Language Processing Pada Telegram Bot. Telematika, 17(2), 145. <https://doi.org/10.31315/telematika.v1i1.3383>