

PENERAPAN ALGORITMA Bi-LSTM PADA APLIKASI KLASIFIKASI SUBJEK KARYA ILMIAH BERDASARKAN JUDUL

Febriana Angela¹, Tony Darmanto², Jimmy Tjen³

^{1,3}Program Studi Informatika, Fakultas Teknologi Informasi Universitas Widya Dharma Pontianak

²Program Studi Sistem Informasi, Fakultas Teknologi Informasi Universitas Widya Dharma Pontianak

e-mail: ¹ 20421332_febriana_a@widyadharma.ac.id, ² tony.darmanto@yahoo.com, ³ jimmy.tjen@mathmods.eu

Abstract

The classifying process for subject paper is still carried out manually, while the volume of paper continues to increase, which in turn increase the complexity of classification task. Considers the current advances in natural language processing technology, opportunities arise to automate classification tasks. In this research, data collection was carried out by conducting observations and documentation studies. The validation of algorithm was carried out using the Black Box and confusion matrix methods. Then an application was developed using Visual Studio Code with Python as the programming language. The application shows a promising results in classifies paper with an accuracy more than 67%. It can be concluded that the Bi-RNN algorithm can be implemented into applications.

Keywords—Text classification, Algorithm Bi-LSTM, NLP, paper subject

Abstrak

Proses klasifikasi subjek karya ilmiah masih dilakukan secara manual, sementara volume karya ilmiah terus meningkat menambah kompleksitas tugas pengklasifikasian. Dengan melihat kemajuan teknologi pemrosesan bahasa alami saat ini, muncul peluang untuk mengotomatiskan tugas pengklasifikasian. Dalam penelitian ini, pengumpulan data dilakukan dengan melakukan observasi dan studi dokumentasi. Untuk menganalisis hasil penerapan algoritma pada aplikasi, dilakukan pengujian dengan menggunakan metode Black Box dan confusion matrix. Kemudian aplikasi dirancang menggunakan Visual Studio Code dengan Python sebagai bahasa pemrogramannya. Aplikasi yang dirancang menunjukkan hasil yang menjanjikan dalam mengklasifikasikan karya ilmiah dengan akurasi lebih dari 67 persen (%). Dapat disimpulkan algoritma Bi-RNN dapat diimplementasikan ke dalam aplikasi.

Kata kunci—Klasifikasi teks, Algoritma Bi-LSTM, NLP, subjek karya ilmiah

1. PENDAHULUAN

Dalam dunia akademis dan riset, volume besar karya ilmiah diproduksi setiap harinya di berbagai bidang ilmu. Proses klasifikasi subjek dari karya-karya ilmiah ini menjadi semakin penting untuk memfasilitasi pencarian informasi yang efisien, guna membantu para peneliti untuk mencari acuan dari banyaknya literatur yang ada. Namun proses klasifikasi subjek karya ilmiah masih dilakukan secara manual, sementara volume karya ilmiah terus meningkat menambah kompleksitas tugas pengklasifikasian. Hal ini dikarenakan proses klasifikasi manual menggunakan sumber daya yang mahal, rentan akan *human error* dan membutuhkan waktu yang cenderung lebih lama. Seiring dengan kemajuan teknologi pemrosesan bahasa alami, muncul peluang untuk mengotomatiskan dan meningkatkan proses klasifikasi subjek karya ilmiah berdasarkan analisis judul.

Pemodelan *Natural Language Processing* (NLP) telah menunjukkan kemampuannya dalam memahami dan mengekstraksi informasi dari teks, termasuk judul karya ilmiah^[1]. Penelitian sebelumnya telah mengaplikasikan teknik-teknik NLP, seperti *word embeddings* dan model berbasis *Transformer* atau *Bidirectional Encoder Representations* (BERT)^[2], untuk tugas-tugas pemrosesan bahasa alami. Namun, penerapan NLP dalam konteks klasifikasi subjek karya ilmiah masih memerlukan eksplorasi dan pengembangan lebih lanjut.

2. METODE PENELITIAN

Pada aplikasi klasifikasi subjek karya ilmiah sistem dirancang sedemikian rupa agar pengguna dapat memasukkan judul yang ingin diklasifikasikan dan melihat hasil klasifikasi yang diprediksi oleh algoritma yang diterapkan. Aplikasi dirancang menggunakan Visual Studio Code dengan Python sebagai bahasa pemrogramannya. Visual Studio Code (juga disebut VS Code) adalah kode editor gratis bersifat *open source* dan

lintas platform yang dikembangkan oleh Microsoft^[3]. Sedangkan Python adalah bahasa pemrograman tingkat tinggi yang memiliki fungsi untuk melakukan eksekusi jumlah instruksi multi guna secara interpretatif^[4]. Setelah aplikasi selesai dirancang, pengujian pada aplikasi akan dilakukan menggunakan pengujian *Black Box*. Pengujian *black box* adalah pengujian yang berfokus pada pengujian persyaratan fungsional perangkat lunak untuk meyakinkan bahwa apakah fungsi-fungsi yang ada sudah berfungsi dengan baik dan tidak memperhatikan struktur logika internal perangkat lunak^[5].

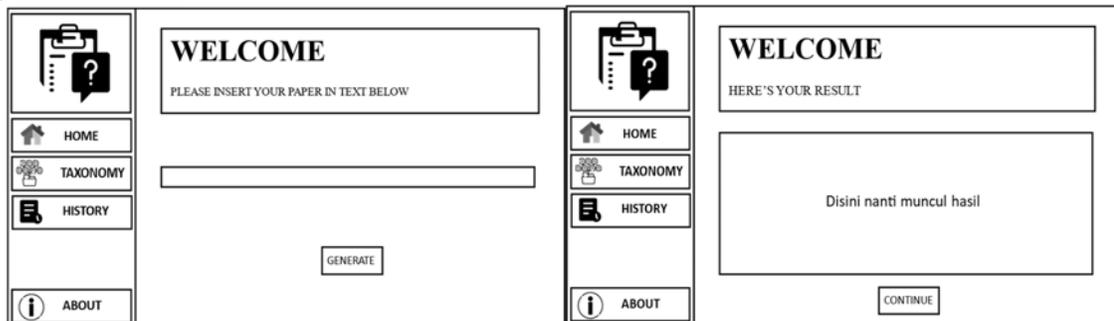
Selanjutnya untuk menerapkan algoritma pada aplikasi yang telah dirancang diperlukan himpunan data untuk melakukan pelatihan pada algoritma serta pengujian untuk melihat seberapa baik algoritma dapat berjalan. Himpunan data yang digunakan dalam pelatihan disebut juga sebagai *dataset*^[6]. Untuk pengujian algoritma yang diterapkan akan dilakukan dengan menganalisis grafik akurasi dan *loss* yang dihasilkan selama pelatihan serta *confusion matrix*. *Confusion matrix* adalah salah satu metode untuk mengevaluasi keakuratan dari pengklasifikasi. Setelah pengklasifikasi telah dicocokkan, *Confusion matrix* akan melaporkan bagaimana nilai pengujian individual untuk setiap kelas yang diprediksi relatif terhadap kelas sebenarnya^[7].

3. HASIL DAN PEMBAHASAN

3.1. Tampilan Aplikasi

Perancangan tampilan antarmuka (*user interface*) adalah sebuah perancangan yang berisi gambaran mengenai sistem yang akan dirancang. Perancangan ini dibuat untuk mempermudah, serta membantu pengguna dalam berinteraksi dengan aplikasi yang akan dibuat. Berikut adalah perancangan tampilan antarmuka pada sistem klasifikasi subjek karya ilmiah:

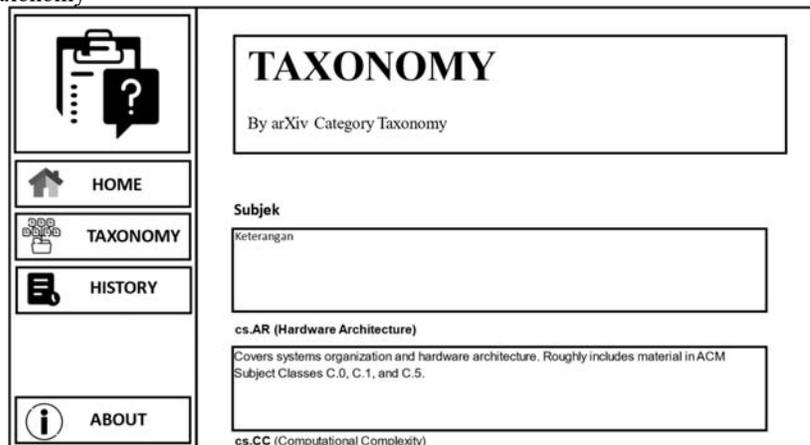
3.1.1. Menu Home



Gambar 1. Tampilan Antarmuka Halaman Menu Home dan Hasil

Menu Home adalah halaman yang didesain untuk menampilkan *textbox* serta tombol Generate untuk memproses hasil klasifikasi judul karya ilmiah yang akan dimasukkan oleh pengguna. Saat tombol diklik hasil akan ditampilkan di layar menu Home. Menu ini akan menjadi tampilan utama saat pengguna memasuki sistem.

3.1.2. Menu Taxonomy

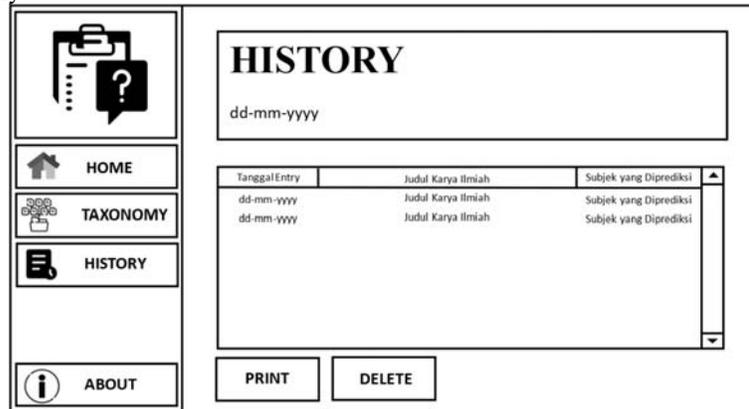


Gambar 2. Tampilan Antarmuka Halaman Taxonomy

Menu Taxonomy adalah halaman yang didesain untuk menampilkan keterangan mengenai sumber referensi *Taxonomy* pengklasifikasian, beserta keterangannya. Halaman ini didesain khusus untuk menambah pengetahuan bagi pembaca seputar subjek klasifikasi karya ilmiah yang digunakan dalam sistem ini. Beberapa

informasi yang akan disampaikan dalam menu Taxonomy adalah kode subjek, nama subjek, serta keterangan mengenai ruang lingkup subjek tersebut.

3.1.3. Menu History



Gambar 3. Tampilan Antarmuka Halaman History

Menu History adalah halaman yang didesain untuk menampilkan hasil klasifikasi judul karya ilmiah yang sudah dimasukkan pengguna ke dalam sistem untuk diproses. Pada menu ini, hasil akan ditampilkan berupa teks judul karya ilmiah beserta presentase hasil pengklasifikasian subjek karya ilmiah yang diprediksi oleh sistem. Menu History ini bersifat sementara dan akan hilang apabila pengguna menutup sistem.

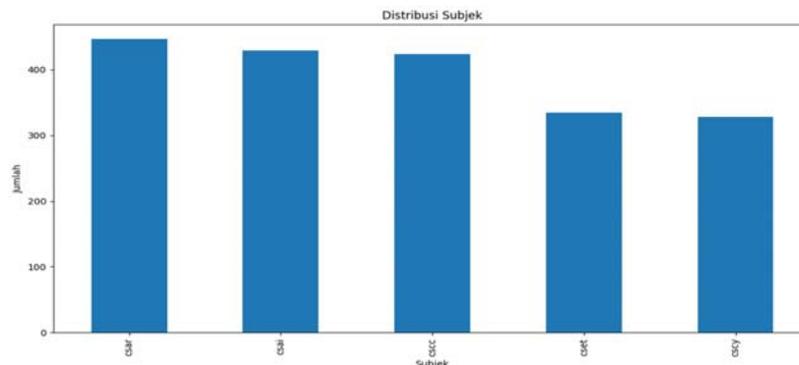
3.1.4. Menu About



Gambar 4. Tampilan Antarmuka Halaman About

Menu About adalah halaman yang didesain untuk menampilkan keterangan mengenai latar belakang pembuatan sistem, serta informasi mengenai pengembang sistem. Informasi yang dimaksud antara lain kontak atau sosial media yang dapat digunakan untuk menghubungi pengembang sistem. Secara khusus halaman ini juga memberikan panduan singkat mengenai cara kerja sistem.

3.2. Dataset



Gambar 4. Dataset Terproses

Dataset yang digunakan diambil dari situs Kaggle. Data tersebut memiliki 56.182 baris dan lebih dari 40 kelas. Namun melihat banyaknya data dan kelas yang ada ditambah dengan adanya ketidakseimbangan pada penyebaran data tiap kelas, peneliti mengambil keputusan untuk memproses kembali data tersebut. Pemrosesan data dilakukan dengan menggunakan aplikasi Ms. Excel, pemrosesan tersebut meliputi penghapusan kelas dan menyeimbangkan penyebaran data. Sehingga pada penelitian ini, hanya lima kelas yang digunakan apabila diperlukan kedepannya jumlah kelas yang ada dapat ditambah. Kemudian untuk menyeimbangkan data dari kelima kelas yang dipilih sebelumnya, peneliti mengambil 447 baris data dari kelas “csar” (*Hardware Architecture*), 429 baris data dari kelas “csai” (*Artificial Intelligence*), 424 baris data dari kelas “cscce” (*Computational Complexity*), 335 baris data dari kelas “cset” (*Emerging Technologies*) dan 328 baris data dari kelas “cscyc” (*Computers and Society*). Apabila dijumlahkan, total data yang kini tersisa adalah 1.963 baris.

3.3. Penerapan Algoritma pada Model Pelatihan

```
# Model RNN
model = Sequential()
model.add(Embedding(input_dim=5000, output_dim=32, input_length=max_word))
model.add(Bidirectional(LSTM(units=100, dropout=0.2, recurrent_dropout=0.2)))
model.add(Flatten())
model.add(Dense(512,activation='relu'))
model.add(Dropout(0.25))
model.add(Dense(units=y_train.shape[1], activation='sigmoid'))

optimizer = tf.keras.optimizers.Adam(learning_rate=0.01)

# Compile the model
model.compile(optimizer=optimizer, loss='categorical_crossentropy', metrics=['accuracy'])

# Display the model summary
model.summary()
```

Gambar 5. Source Code Model Bi-LSTM

Dalam penerapannya di dalam sistem, peneliti menggunakan algoritma Long Short-Term Memory (LSTM), tepatnya Bidirectional LSTM (Bi-LSTM). LSTM adalah salah satu jenis model RNN yang mampu untuk pembelajaran jangka panjang. Sedangkan Bi-LSTM adalah versi pengembangan lainnya dari LSTM. Bi-LSTM memperbaiki kemampuan yang lebih baik dari LSTM untuk mempelajari dan memproses data sekuensial. Bi-LSTM memiliki dua lapisan, lapisan pertama digunakan untuk memproses data sekuensial dari depan dan lapisan kedua digunakan untuk memproses data sekuensial dari belakang. Dengan demikian, dalam pemrosesannya model dapat mempelajari dan memahami pemrosesan data dari dua arah. Gambar 5 menunjukkan source code hasil implementasi Bi-LSTM yang telah disesuaikan dengan aplikasi.

Pengujian sistem akan membahas seputar seberapa baik sistem dapat berjalan, mulai dari sisi tampilan antar muka hingga model yang sudah dilatih. Oleh sebab itu dibagi lah pembahasan menjadi dua bagian, yakni pengujian tampilan antarmuka dan pengujian model. Setiap pembahasan akan menggunakan teknik pengujian yang berbeda. Pengujian tampilan antar muka menggunakan *Black Box Testing*, sedangkan pengujian model akan menggunakan visualisasi akurasi dari hasil pelatihan dan *Confusion Matrix*.

3.4. Pengujian Black Box

Black Box testing adalah metode pengujian yang fokus pada fungsionalitas sistem tanpa melihat struktur code yang ada di dalam sistem. Oleh sebab itu pengujian ini umum digunakan untuk menguji apakah tampilan antarmuka dapat menjalankan fungsi-fungsi yang ada dengan baik atau tidak. Hasil pengujian Black Box akan disampaikan dalam bentuk tabel dengan enam kolom yang bertuliskan “No.,” “Skenario Pengujian,” “Test Case,” “Hasil yang Diharapkan,” “Hasil Pengujian” dan “Status”.

Kolom “No.” berisi nomor urut dari setiap skenario pengujian. Kolom “Skenario Pengujian” berisi deskripsi singkat mengenai apa yang akan di uji. Kolom “Test Case” berisi jenis ujian yang dijalankan. Kolom “Hasil yang Diharapkan” berisi gambaran akan apa yang seharusnya terjadi apabila sistem bekerja dengan benar. Kemudian yang terakhir, kolom “Status” berisi penilaian dari hasil pengujian yang dilakukan dengan menyesuaikan hasil dari kolom “Hasil yang Diharapkan”. Berikut tabel hasil pengujian dari tiap tampilan antarmuka yang ada pada sistem:

3.4.1. Halaman Menu Home

Tabel 1. Hasil Pengujian Halaman Menu Home

No.	Skenario Pengujian	Test Case	Hasil yang Diharapkan	Hasil Pengujian	Status
1.	Memilih menu <i>Home</i>	Pengujian Fungsionalitas dasar	Tidak akan terjadi perubahan apapun.	Sesuai harapan	<i>Valid</i>
2.	Memilih menu <i>Taxonomy</i>	Pengujian Fungsionalitas dasar	Akan menampilkan halaman <i>Taxonomy</i>	Sesuai harapan	<i>Valid</i>
3.	Memilih menu <i>History</i>	Pengujian Fungsionalitas dasar	Akan menampilkan halaman <i>History</i>	Sesuai harapan	<i>Valid</i>
4.	Memilih menu <i>About</i>	Pengujian Fungsionalitas dasar	Akan menampilkan halaman <i>About</i>	Sesuai harapan	<i>Valid</i>
5.	Mengetikkan judul	Pengujian Fungsionalitas dasar	Akan menampilkan judul yang telah diketik	Sesuai harapan	<i>Valid</i>
6.	Mengeklik tombol <i>Generate</i>	Pengujian Fungsionalitas dasar	Akan menampilkan panel berisi judul dan hasil	Sesuai harapan	<i>Valid</i>
7.	Mengeklik tombol <i>Continue</i>	Pengujian Fungsionalitas dasar	Akan menghapus panel hasil dan menampilkan <i>text box</i> yang kosong.	Sesuai harapan	<i>Valid</i>

3.4.2. Halaman Menu Taxonomy

Tabel 2. Hasil Pengujian Halaman Menu Taxonomy

No.	Skenario Pengujian	Test Case	Hasil yang Diharapkan	Hasil Pengujian	Status
1.	Memilih menu <i>Home</i>	Pengujian Fungsionalitas dasar	Akan menampilkan halaman <i>Home</i>	Sesuai harapan	<i>Valid</i>
2.	Memilih menu <i>Taxonomy</i>	Pengujian Fungsionalitas dasar	Tidak akan terjadi perubahan apapun.	Sesuai harapan	<i>Valid</i>
3.	Memilih menu <i>History</i>	Pengujian Fungsionalitas dasar	Akan menampilkan halaman <i>History</i>	Sesuai harapan	<i>Valid</i>
4.	Memilih menu <i>About</i>	Pengujian Fungsionalitas dasar	Akan menampilkan halaman <i>About</i>	Sesuai harapan	<i>Valid</i>
5.	Menggunakan <i>scroll bar</i>	Pengujian Fungsionalitas dasar	Akan menampilkan teks yang terpotong lebar <i>window</i> .	Sesuai harapan	<i>Valid</i>

Halaman Menu History

Tabel 3. Hasil Pengujian Halaman Menu History

No.	Skenario Pengujian	Test Case	Hasil yang Diharapkan	Hasil Pengujian	Status
1.	Memilih menu <i>Home</i>	Pengujian Fungsionalitas dasar	Akan menampilkan halaman <i>Home</i>	Sesuai harapan	<i>Valid</i>
2.	Memilih menu <i>Taxonomy</i>	Pengujian Fungsionalitas dasar	Akan menampilkan halaman <i>Taxonomy</i>	Sesuai harapan	<i>Valid</i>
3.	Memilih menu <i>History</i>	Pengujian Fungsionalitas dasar	Tidak akan terjadi perubahan apapun.	Sesuai harapan	<i>Valid</i>

4.	Memilih menu <i>About</i>	Pengujian Fungsionalitas dasar	Akan menampilkan halaman About	Sesuai harapan	<i>Valid</i>
5.	Menampilkan riwayat klasifikasi	Pengujian Fungsionalitas dasar	Akan menampilkan hasil prediksi yang tersimpan dalam aplikasi Excel.	Sesuai harapan	<i>Valid</i>
6.	Mencetak riwayat klasifikasi	Pengujian Fungsionalitas dasar	Akan menjalankan proses mencetak riwayat ke dalam bentuk fisik.	Sesuai harapan	<i>Valid</i>
7.	Menghapus baris data yang terpilih.	Pengujian Fungsionalitas dasar	Akan menampilkan riwayat yang telah diperbarui.	Sesuai harapan	<i>Valid</i>
8.	Menggunakan <i>scroll bar</i> .	Pengujian Fungsionalitas dasar	Akan menampilkan teks yang terpotong oleh panjang panel.	Sesuai harapan	<i>Valid</i>

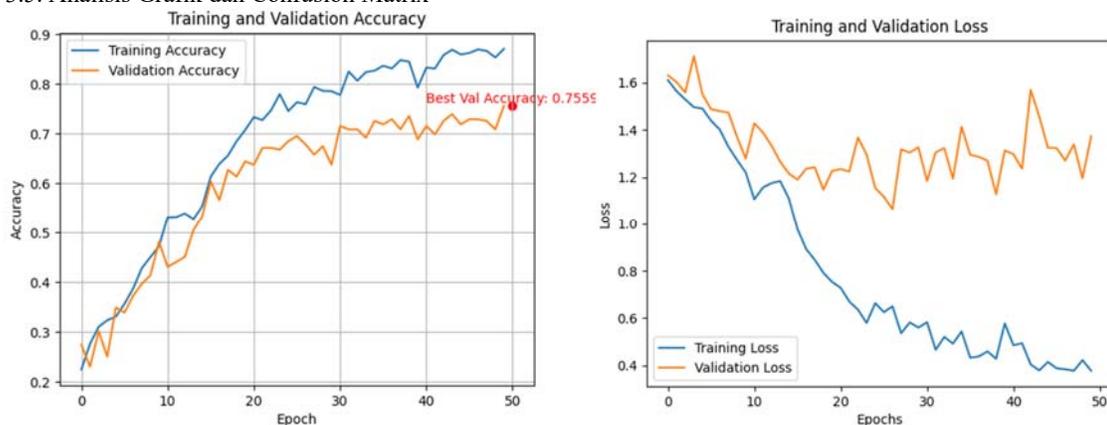
3.4.4. Halaman Menu About

Tabel 4. Hasil Pengujian Halaman Menu About

No.	Skenario Pengujian	<i>Test Case</i>	Hasil yang Diharapkan	Hasil Pengujian	<i>Status</i>
1.	Memilih menu <i>Home</i>	Pengujian Fungsionalitas dasar	Akan menampilkan halaman Home	Sesuai harapan	<i>Valid</i>
2.	Memilih menu <i>Taxonomy</i>	Pengujian Fungsionalitas dasar	Akan menampilkan halaman Taxonomy	Sesuai harapan	<i>Valid</i>
3.	Memilih menu <i>History</i>	Pengujian Fungsionalitas dasar	Akan menampilkan halaman History	Sesuai harapan	<i>Valid</i>
4.	Memilih menu <i>About</i>	Pengujian Fungsionalitas dasar	Tidak akan terjadi perubahan apapun.	Sesuai harapan	<i>Valid</i>
5.	Menampilkan teks isi menu <i>About</i>	Pengujian Fungsionalitas dasar	Dapat melihat teks panduan, latar belakang dan perihal <i>developer</i> .	Sesuai harapan	<i>Valid</i>

Berdasarkan tabel-tabel tersebut, dapat disimpulkan bahwa semua fungsionalitas dasar dari tiap tampilan antarmuka yang ada pada sistem klasifikasi subjek karya ilmiah melalui judul karya ilmiah bekerja dengan baik. Hal ini dikarenakan setiap tabel memiliki nilai status *valid* pada setiap pengujian yang dilakukan pada fungsionalitas dari masing-masing tampilan antarmuka, sehingga dapat dikatakan bahwa tidak ada masalah apapun pada tampilan antarmuka sistem aplikasi pengklasifikasian subjek judul karya ilmiah. Tampilan antarmuka sistem berjalan sesuai yang diharapkan.

3.5. Analisis Grafik dan Confusion Matrix

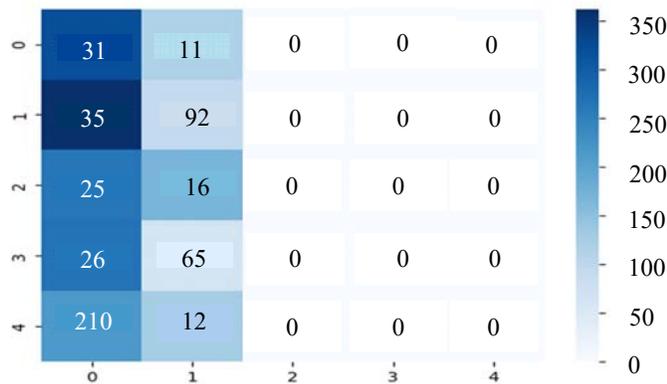


Gambar 6. Grafik Akurasi dan Loss

Analisis pengujian pada model pertama dilakukan dengan melihat grafik akurasi dan loss yang dilakukan selama pelatihan. Melalui Gambar 5 diketahui bahwa data latih (direpresentasikan oleh garis biru) memiliki akurasi

tertinggi sebesar 80 persen (%), sedangkan data validasi (direpresentasikan oleh garis oranye) memiliki akurasi tertinggi sebesar 75.55 persen (%) selama pelatihan dijalankan. Kemudian pada grafik *Loss* dapat dilihat pula penurunan tingkat kegagalan pelatihan model. Gambar tersebut menunjukkan bahwa kerugian pelatihan berkurang seiring bertambahnya jumlah epoch. Namun dengan membandingkan kedua garis tersebut didapati bahwa garis oranye mengalami ketidakstabilan yang menuntun pada adanya kemungkinan terjadi *overfitting*. *Overfitting* adalah kondisi dimana model memiliki kinerja yang lebih baik saat melakukan pelatihan menggunakan data latih dari pada data validasi^[8].

Selanjutnya pengujian dilakukan dengan menggunakan *confusion matrix*. *Confusion matrix* adalah tabel yang memvisualisasikan performa dari algoritma pada model. Tiap baris pada *confusion matrix* akan merepresentasikan nilai kelas yang diprediksi, sedangkan tiap kolom pada *confusion matrix* akan merepresentasikan nilai kelas yang sebenarnya. Gambar 6. menunjukkan tabel matrix dengan ukuran 5x5 yang berarti ada lima kelas yang diprediksi. Warna-warna pada tabel tersebut menandakan bahwa semakin gelap warna yang ada semakin tinggi pula persentase keakuratan prediksi yang dilakukan. Kemudian untuk nilai yang pada tiap cell matrix merepresentasikan persentase dari prediksi yang dilakukan oleh model. Dengan menganalisis *confusion matrix* didapati model hanya dapat mengklasifikasikan dua dari lima subjek yang ada. Melalui analisis pada pengujian ini, dapat disimpulkan model perlu ditingkatkan kembali.



Gambar 7. Confusion Matrix

Dengan menggunakan data dan kelas yang telah disampaikan sebelumnya, model dilatih kembali demi mencari hasil dengan akurasi yang lebih baik. Ringkasan dari hasil pelatihan akan ditampilkan melalui Tabel 5. Tabel 5. menampilkan empat kolom utama, yakni “No.,” “Tunning,” “Hasil dan Keterangan”. Kolom “No.” merepresentasikan nomor urut dari percobaan yang telah dilakukan. Kolom “Tunning” merepresentasikan parameter yang digunakan dalam pelatihan model.

Parameter yang dimaksud antara lain “LR”, “NW”, “TS”, “BS” dan “Epoch”. “LR” adalah singkatan dari Learning Rate yang menggambarkan kecepatan belajar dari model. “NW” atau Num Word adalah jumlah kata yang digunakan dalam pelatihan model. “TS” atau Test Size adalah nilai pembagian data yang digunakan untuk validasi pelatihan model. Kemudian terakhir “BS” dan “Epoch”. “BS” adalah singkatan dari Batch Size yang menggambarkan jumlah sampel data yang diproses sebelum model diperbarui. Epoch adalah jumlah siklus lengkap pelatihan pada seluruh dataset. Selanjutnya kolom Hasil menampilkan hasil akurasi yang didapatkan dan Keterangan menampilkan lokasi gambar visual dari hasil pelatihan yang telah dilakukan.

Tabel 5. Hasil Pengujian Berulang

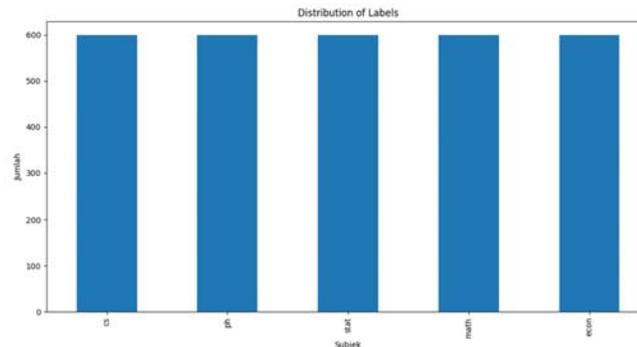
No	Tunning					Hasil
	LR	NW	TS	BS	Epoch	
1.	0.01	5000	0.15	32	50	Akurasi validasi tertinggi mencapai 75,59 persen.
2.	0.01	5000	0.15	32	50	Akurasi validasi tertinggi mencapai 71,86 persen.
3.	0.01	5000	0.15	32	50	Akurasi validasi tertinggi mencapai 73,90 persen.
4.	0.01	5000	0.15	32	50	Akurasi validasi tertinggi mencapai 78,31 persen.
5.	0.01	5000	0.15	32	50	Akurasi validasi tertinggi mencapai 74,92 persen.
6.	0.01	5000	0.15	32	50	Akurasi validasi tertinggi mencapai 73,90 persen.
7.	0.01	5000	0.15	32	50	Akurasi validasi tertinggi mencapai 75,25 persen.
8.	0.01	5000	0.15	32	50	Akurasi validasi tertinggi mencapai 76,95 persen.
9.	0.01	5000	0.15	32	50	Akurasi validasi tertinggi mencapai 73,90 persen.
10.	0.01	5000	0.15	32	50	Akurasi validasi tertinggi mencapai 76,27 persen.

Melalui Tabel 5 diketahui bahwa akurasi tertinggi yang dapat dicapai dengan parameter tersebut terdapat pada pelatihan nomor empat dengan akurasi mencapai 78,31 persen dari sepuluh percobaan yang dilakukan tanpa melakukan perubahan pada parameter yang ada. Selanjutnya pelatihan dilakukan dengan mengubah beberapa parameter. Perubahan dan hasil pelatihan tersebut disebut juga tuning dan dapat dilihat pada Tabel 6 dengan arsitektur yang serupa dengan tabel sebelumnya. Nomor pelatihan satu sampai lima pada Tabel 6 menjalankan pelatihan dengan mengubah parameter Num Word menjadi 10.000, melalui perubahan tersebut didapati akurasi tertinggi yang dicapai adalah 79.66 persen. Kemudian pelatihan nomor enam sampai sepuluh pada Tabel 6 menjalankan pelatihan dengan mengubah Learning Rate menjadi 0,001 dan Num Word menjadi 20.000, sehingga melalui perubahan tersebut didapati akurasi tertinggi yang telah dicapai adalah 74,24 persen.

Tabel 6. Hasil Pengujian Tunning

No	Tunning					Hasil
	LR	NW	TS	BS	Epoch	
1.	0,01	10.000	0,15	32	50	Akurasi validasi tertinggi mencapai 79,66 persen.
2.	0,01	10.000	0,15	32	50	Akurasi validasi tertinggi mencapai 78,46 persen.
3.	0,01	10.000	0,15	32	50	Akurasi validasi tertinggi mencapai 72,88 persen.
4.	0,01	10.000	0,15	32	50	Akurasi validasi tertinggi mencapai 77,29 persen.
5.	0,01	10.000	0,15	32	50	Akurasi validasi tertinggi mencapai 73,90 persen.
6.	0,001	20.000	0,15	32	50	Akurasi validasi tertinggi mencapai 72,88 persen.
7.	0,001	20.000	0,15	32	50	Akurasi validasi tertinggi mencapai 74,24 persen.
8.	0,001	20.000	0,15	32	50	Akurasi validasi tertinggi mencapai 68,81 persen.
9.	0,001	20.000	0,15	32	50	Akurasi validasi tertinggi mencapai 71,86 persen.
10.	0,001	20.000	0,15	32	50	Akurasi validasi tertinggi mencapai 72,54 persen.

Demi mendapatkan hasil yang lebih baik, dilakukan perubahan pada dataset dengan hipotesis bahwa judul karya ilmiah yang terdapat pada dataset sulit diklasifikasikan karna kelas yang sejenis (yakni bidang komputer). Dataset yang baru, diambil melalui situs arXiv dan diolah secara pribadi. Jumlah kolom dan kelas tidak berubah yang membedakan adalah jumlah data, jenis kelas dan judul. Kelas pada dataset baru terdiri dari “math” (Matematika), “ph” (Fisika), “stat” (Statistika), “econ” (Ekonomi), dan “cs” (Komputer Sains). Jumlah data baru mencapai 600 data tiap kelas dan total 4.000 data secara keseluruhan.



Gambar 8. Dataset Baru

Pada Tabel 7. dapat dilihat hasil dari pelatihan menggunakan dataset baru beserta tiap parameternya. Dari empat pelatihan didapati akurasi tertinggi mencapai 69,78 persen dengan rata-rata akurasi lainnya tidak mencapai 50 persen. Hasil yang ditampilkan lebih buruk dibanding sebelumnya, sehingga dapat disimpulkan bahwa hipotesis yang diajukan bernilai salah atau tidak benar. Didasari oleh data tersebut maka dapat disampaikan bahwa model tidak sesuai dengan data yang dimiliki. Oleh sebab itu model perlu direkonstruksi kembali. Ringkasan pelatihan model rekonstruksi dapat dilihat pada Tabel 8.

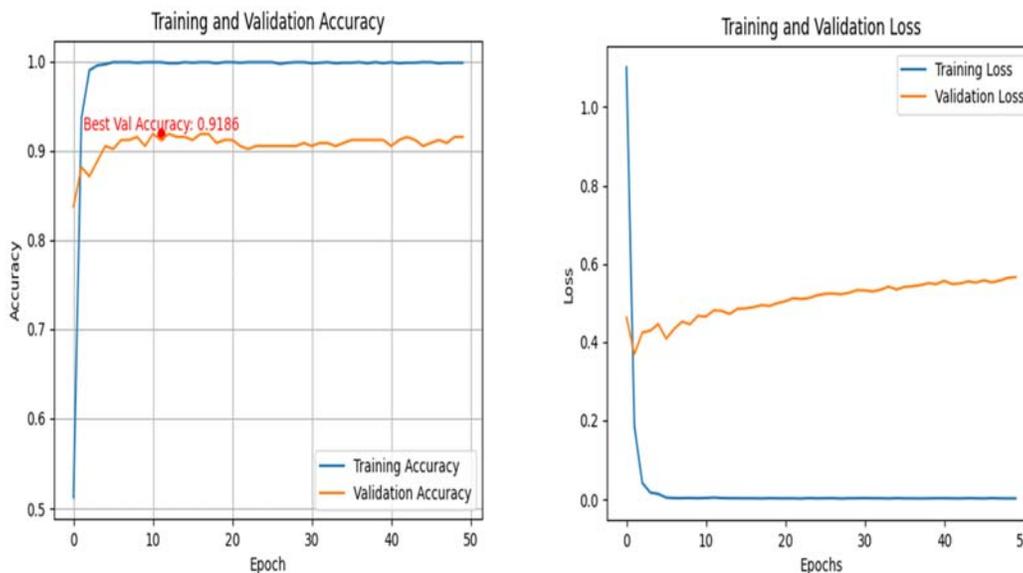
Tabel 7. Hasil Pengujian Dataset Baru

No	Tunning					Hasil
	LR	NW	TS	BS	Epoch	
1.	0.01	5000	0.15	32	50	Akurasi validasi tertinggi mencapai 44,67 persen.
2.	0.01	5000	0.15	32	50	Akurasi validasi tertinggi mencapai 69,78 persen.
3.	0.01	5000	0.15	32	50	Akurasi validasi tertinggi mencapai 43,33 persen.
4.	0.01	5000	0.15	32	50	Akurasi validasi tertinggi mencapai 35,78 persen.

Tabel 8. Hasil Pengujian Model Rekonstruksi

No	Tunning					Hasil
	LR	NW	TS	BS	Epoch	
1.	0.01	5000	0.15	32	50	Akurasi validasi tertinggi mencapai 90,17 persen.
2.	0.01	5000	0.15	32	50	Akurasi validasi tertinggi mencapai 91,19 persen.
3.	0.01	5000	0.15	32	50	Akurasi validasi tertinggi mencapai 90,17 persen.
4.	0.01	5000	0.15	32	50	Akurasi validasi tertinggi mencapai 87,46 persen.
5.	0.01	5000	0.15	32	50	Akurasi validasi tertinggi mencapai 91,53 persen.
6.	0.01	5000	0.15	32	50	Akurasi validasi tertinggi mencapai 88,81 persen.
7.	0.01	5000	0.15	32	50	Akurasi validasi tertinggi mencapai 90,51 persen.
8.	0.01	5000	0.15	32	50	Akurasi validasi tertinggi mencapai 90,85 persen.
9.	0.01	5000	0.15	32	50	Akurasi validasi tertinggi mencapai 90,51 persen.
10.	0.01	5000	0.15	32	50	Akurasi validasi tertinggi mencapai 91,86 persen.

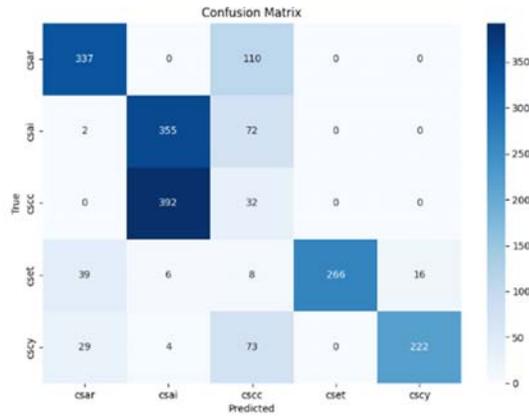
Dapat dilihat pada Tabel 8, pelatihan pada model rekonstruksi dilakukan sebanyak sepuluh kali menggunakan *dataset* awal (kelas : csar, csai, csc, cset, dan cscy). Dari sepuluh kali pelatihan akurasi tertinggi yang didapat berasal dari pelatihan nomor sepuluh dengan akurasi 91,86 persen. Namun melihat grafik akurasi dan hasil akurasi saja tidak cukup untuk menentukan apakah model itu cukup baik untuk dapat digunakan atau tidak. Dengan mempertimbangkan kestabilan grafik *Loss* penurunan kesalahan pada validasi dan keakuratan prediksi pada *confusion matrix*, didapati model terlatih yang terbaik terdapat pada pelatihan model rekonstruksi nomor sepuluh yang merupakan model dengan akurasi tertinggi. Berikut alasan serta pembahasannya, apabila gambar pelatihan model rekonstruksi nomor sepuluh dianalisa kembali, dapat dilihat bahwa grafik *Loss* penurunan kesalahan pada validasi model tersebut adalah yang paling stabil diantara pelatihan lainnya. Hal ini dikarenakan pada grafik tersebut dapat dilihat garis validasi *Loss* tidak memiliki kenaikan atau penurunan yang drastis.



Gambar 8. Grafik Akurasi dan Loss Model Rekonstruksi Nomor 10

Alasan berikutnya didapati dengan menganalisis *confusion matrix*. *Confusion matrix* pada pelatihan model rekonstruksi nomor sepuluh menunjukkan kemampuan model untuk dapat mengklasifikasikan empat subjek dengan hampir sempurna selain subjek “csc”. Subjek “csc” ditunjukkan banyak disalah prediksi sebagai “csai”. Hal ini mungkin terjadi dikarenakan judul karya ilmiah subjek “csc” memiliki banyak kesamaan atau hampir serupa dengan “csai”, hingga menyulitkan model untuk mengklasifikasikannya.

Melihat kembali semua percobaan sebelumnya, di antara semua percobaan pelatihan yang telah dilakukan, dapat disampaikan bahwa pelatihan dari model hasil rekonstruksi nomor sepuluh memberikan hasil yang jauh lebih baik diantara yang lainnya. Sehingga diputuskan model yang akan digunakan dalam sistem aplikasi pengklasifikasian subjek judul karya ilmiah adalah model Bi-LSTM hasil rekonstruksi nomor sepuluh.



Gambar 9. Confusion Matrix Model Rekonstruksi Nomor 10

Hasil rekonstruksi model, dapat dilihat pada Gambar 10, Gambar 10 menampilkan model dengan lapisan yang lebih sederhana. Rekonstruksi yang dilakukan antara lain menghapus *Flatten layer*, *Dropout layer*, dan penghapusan *dropout* yang ada di lapisan LSTM. Selain itu jumlah dari unit yang ada di lapisan LSTM juga diubah. Kemudian jumlah dari unit pada lapisan terakhir yang sebelumnya diatur secara fleksibel untuk mengikuti jumlah kelas, kini ditetapkan untuk lima kelas.

```
# Model RNN
model = Sequential()
model.add(Embedding(input_dim=5000, output_dim=32, input_length=max_word))
model.add(Dense(512, activation='relu'))
model.add(Bidirectional(LSTM(32)))
model.add(Dense(5, activation='sigmoid'))

optimizer = tf.keras.optimizers.Adam(learning_rate=0.01)

# Compile the model
model.compile(optimizer=optimizer, loss='categorical_crossentropy', metrics=['accuracy'])

# Display the model summary
model.summary()
```

Gambar 10. Confusion Matrix Model Rekonstruksi Nomor 10

4. KESIMPULAN

Berdasarkan pembahasan sebelumnya, mulai dari perancangan antarmuka sistem hingga pengujian serta analisis hasil pengujian model dan tampilan antarmuka pada sistem. Kesimpulan yang dapat diambil dapat dijabarkan sebagai berikut:

- Berdasarkan perancangan dan pengujian pada sistem yang telah dilakukan sejauh ini, diketahui bahwa dalam penerapan algoritma Bi-LSTM pada aplikasi klasifikasi subjek karya ilmiah berdasarkan judul memerlukan berbagai persiapan dan penyesuaian.
- Melalui pengujian *Black Box* didapati tampilan antarmuka sistem dapat berfungsi dengan baik. Pengguna dapat memasukkan judul untuk diprediksi sistem dan sistem dapat menampilkan hasilnya.
- Dari hasil pengujian dan analisis pengujian sistem, diketahui algoritma yang dipilih dapat diimplementasikan, ke dalam aplikasi yang telah dirancang.
- Melalui beberapa percobaan pelatihan yang telah dilakukan algoritma yang dipilih masih menunjukkan banyak ketidak stabilan yang mengacu pada kemungkinan *overfitting* (kondisi di mana model memiliki kinerja yang lebih baik saat melakukan pelatihan menggunakan data latih dari pada data validasi).

5. SARAN

Berdasarkan pembahasan sebelumnya, mulai dari perancangan antarmuka sistem hingga pengujian serta analisis hasil pengujian model dan tampilan antarmuka pada sistem. Saran yang dapat disampaikan untuk penelitian selanjutnya diharapkan:

- Menggunakan algoritma yang berbeda untuk dapat melihat apakah letak kekurangan ada pada data atau model dari algoritma yang dipilih.
- Menambahkan subjek pengklasifikasian, agar pengguna dapat mengklasifikasi judul karya ilmiah dengan subjek yang lebih bervariasi.

- c. Memperluas cakupan jenis bahasa pada sistem, sehingga sistem tidak hanya dapat mengklasifikasikan judul dalam satu bahasa saja.
- d. Menambah fitur-fitur baru seperti fungsi pencarian, fungsi sortir dan mengganti tema. Selain itu, peneliti juga dapat menambahkan fungsi yang dapat digunakan agar beberapa orang dapat mengakses data secara *realtime* secara bersamaan.

UCAPAN TERIMA KASIH

Dalam penyusunan penelitian ini, penulis banyak mendapat bantuan dari berbagai pihak, berupa petunjuk, bimbingan, informasi, nasihat dan saran. Maka dari itu, pada kesempatan ini penulis mengucapkan terima kasih kepada civitas academica Fakultas Teknologi Informasi Universitas Widya Dharma Pontianak atas bimbingan dan saran yang telah diberikan selama penelitian berlangsung. Terima kasih kepada ayah,ibu, saudara dan sahabat tercinta yang telah banyak memberikan bantuan dan dorongan moral selama penulisan jurnal. Serta terima kasih kepada pihak-pihak yang tidak dapat penulis sebutkan satu persatu yang telah turut membantu sehingga penelitian ini dapat diselesaikan.

Penulis menyadari bahwa penelitian ini masih jauh dari kata sempurna, meski demikian penelitian ini telah memberikan manfaat dan pengalaman bagi penulis. Akhir kata penulis berharap penelitian ini dapat bermanfaat bagi semua pihak dan dapat dipergunakan sebagaimana mestinya. Segala bentuk kritik dan saran yang bersifat membangun akan penulis terima dengan senang hati

DAFTAR PUSTAKA

- [1] Peng Zhou, Z., Qil, S., Zheng, J., Xu, H., Bao, B., Xu, (2016), Text Classification Improved by Integrating Bidirectional LSTM with Two-dimensional Max Pooling, <https://arxiv.org/pdf/1611.06639>, diakses tgl 17 Juni 2024.
- [2] Zhen Li, X., Wang, W., Yang, J., Wu, Z., Zhang, Z., Liu, M., Sun, H., Zhang, S., Liu (2022), A Unified Understanding of Deep NLP Models for Text Classification, <https://arxiv.org/pdf/2206.09355>, diakses tgl 17 Juni 2024.
- [3] Speight, April. (2021). *Visual Studio Code untuk Pemrogram Python* (judul asli: Visual Studio Code for Python Programmers). Wiley.
- [4] Pane, Syafrial Fachri dan Esi Vidia Rahmadani. (2020). *Big Data: Forecasting Menggunakan Python*. Kreatif Bandung.
- [5] Dawis, M., Aisyah, Y.W.S., Putra, Fitri, D. Hamidin, S.N., Yutia, Manniah, N.R., Feta, D.W., Rahma, F., Natsir. (2023). *Rekayasa Perangkat Lunak Panduan Praktis Untuk Pengembangan Aplikasi Berkualitas*. Widina Media Utama. Bandung.
- [6] Enterprise, Jubilee. (2023). *Python untuk Analisis dan Visualisasi Data*. Elex Media Komputindo. Jakarta.
- [7] Bongfort, Benjamin, Rebecca Bilbro dan Tony Ojeda. (2018). *Applied Text Analysis With Python*. O'Reilly Media. Sebastopol.
- [8] Dean, Jaret, (2014). *Big Data, Data Mining, and Machine Learning*. Wiley.