

PENGEMBANGAN SISTEM PEMANTAUAN SENTIMEN BERITA BERBAHASA INDONESIA BERDASARKAN KONTEN DENGAN *LONG SHORT-TERM MEMORY*

Dewi Yanti Liliana^{*1}, Nadia Nurul Hikmah², Maykada Harjono³

^{1,2}Jurusan Teknik Informatika dan Komputer, Politeknik Negeri Jakarta

³Direktorat Jenderal Aplikasi Informatika, Kementerian Komunikasi dan Informatika

Email: ¹dewiyanti.liliana@tik.pnj.ac.id, ²nadia.hikmah.tik16@mhs.pnj.ac.id ³mayk001@kominfo.go.id

*Penulis Korespondensi

(Naskah masuk: 19 Januari 2021, diterima untuk diterbitkan: 18 Oktober 2021)

Abstrak

Kementerian Komunikasi dan Informatika (Kemkominfo) memiliki tugas salah satunya untuk mengawasi konten berita yang beredar di media digital. Dengan terus bertambahnya berita online di internet, Kemkominfo dihadapkan pada permasalahan pengklasifikasian sentimen berita yang masih dilakukan secara manual dengan membaca konten berita satu persatu lalu menangkap sentimen dari berita, yaitu sentimen positif, negatif, atau netral. Hal ini sangat melelahkan dan memakan waktu mengingat volume dan kecepatan pertumbuhan berita setiap harinya semakin masif. Untuk itu diperlukan pengembangan sistem pengklasifikasi sentimen berita daring secara otomatis untuk pemantauan berita berbahasa Indonesia. Sistem pengklasifikasi secara otomatis berbasis *machine learning* dilakukan dengan membangun model pembelajaran dari korpus berita yang berasal dari situs berita daring. Korpus data tersebut kemudian diproses menggunakan algoritma *Long Short-Term Memory* (LSTM). LSTM biasa digunakan untuk menangani kasus klasifikasi dalam berbagai bidang khususnya dengan input berupa teks sekuensial. Model LSTM diimplementasikan ke dalam aplikasi berbasis web untuk menentukan jenis dari sentimen berita. Berdasarkan hasil pengujian yang dilakukan, model LSTM yang dibuat memiliki tingkat akurasi sebesar 86%. Dengan demikian implementasi LSTM mampu menjadi suatu solusi untuk mengatasi masalah pengklasifikasian sentimen berita daring secara otomatis untuk pemantauan sentimen berita di Kemkominfo.

Kata kunci: *berita daring, LSTM, pengklasifikasi, sentimen berita, sistem pemantauan berita*

THE DEVELOPMENT OF CONTENT-BASED INDONESIAN NEWS SENTIMENT MONITORING SYSTEM USING LONG SHORT-TERM MEMORY

Abstract

The Ministry of Communication and Informatics (Kemkominfo) has one duty to monitor news content circulating in digital media. With the increasing number of online news in the internet, Kemkominfo is facing the problem of classifying news sentiment which is still done manually by reading the contents of the news one by one, and then capturing the sentiment of the news; either positive, negative, or neutral. This is very exhausting and time consuming considering the volume and speed of growth of news every day is getting massive. This requires the development of an automatic online news sentiment classification system for monitoring Indonesian news. Machine learning-based automatic classification systems are carried out by building a learning model from a news corpus originating from news sites. The data is then processed using the Long Short Term Memory (LSTM) algorithm. LSTM is commonly used to handle classification in various fields especially in a sequential input. The LSTM model is implemented into a web-based application to determine the types of news sentiment. Based on the results of the tests carried out, the LSTM model created has an accuracy rate of 86%. Thus, the implementation of LSTM is potentially become a solution to overcome the problem of automatic online news sentiment classification for the news content monitoring system at the Ministry of Communication and Information.

Keywords: *online news, LSTM, classifier, news sentiment, monitoring system*

1. PENDAHULUAN

Kementerian Komunikasi dan Informatika (Kemkominfo) merupakan lembaga kementerian

negara yang bergerak di bidang komunikasi dan informatika untuk membantu Presiden dalam menyelenggarakan pemerintahan negara

(www.kominfo.go.id). Kemkominfo memiliki beberapa fungsi yaitu untuk perumusan, penetapan, dan pelaksanaan kebijakan di bidang komunikasi dan informatika, serta pelaksanaan kegiatan teknis yang berskala nasional. Untuk melaksanakan fungsinya, Kemkominfo memiliki media yang dapat membantu masyarakat dalam memperoleh informasi yang cepat dan akurat, serta melakukan monitoring/pengawasan terhadap konten yang tersebar di media baik dalam bentuk video, tulisan, dan berita.

Berita merupakan hal yang sedang terjadi dalam beberapa waktu terakhir yang dapat menarik minat banyak orang dan menghasilkan rasa keingintahuan yang tinggi di antara para pendengar. Pada awalnya, berita disampaikan melalui koran, majalah, televisi, maupun radio. Seiring berjalannya waktu, berita dapat diakses melalui website maupun aplikasi, atau yang dikenal sebagai portal berita online.

Ketika sebuah media menyebarluaskan berita akan ada dampak yang ditimbulkan bagi pembaca, baik itu positif, negatif, maupun tidak memengaruhi sama sekali (netral). Agar pembaca dapat bijaksana dalam memilih berita, maka diperlukan sistem yang bisa mengategorikan sentimen berita menjadi positif, negatif, atau netral agar suatu berita dapat diklasifikasikan jenis sentimennya oleh

Kemkominfo. Selama ini Kemkominfo kesulitan dalam menyeleksi berita yang beredar di media online. Pada saat menyeleksi berita masih dilakukan dengan cara manual dan isi beritanya harus dianalisis terlebih dahulu baru kemudian dikategorikan jenis sentimennya. Selain itu, artikel yang dimuat di berbagai situs semakin banyak dengan kategori yang beragam dan isi yang serupa.

Untuk pemilihan kategori berita yang relevan, maka diperlukan sistem klasifikasi yang dapat membantu Kemkominfo dalam menentukan sentimen berita berdasarkan konten berita. Penelitian ini menerapkan algoritma Long short-term memory (LSTM). Long Short Term Memory (LSTM) adalah algoritma pemrosesan berbasis Deep Learning yang merupakan varian dari Recurrent Neural Network (RNN). Pada LSTM dilakukan modifikasi pada RNN dengan menambahkan sel memori yang dapat menyimpan informasi untuk jangka waktu yang lama (Souma, Vodenska and Aoyama, 2019). LSTM memiliki kelebihan daripada RNN di mana terdapat arsitektur untuk mengingat dan melupakan output yang akan diproses kembali menjadi input. Selain itu, LSTM juga dapat mengelola error yang terjadi (Zhang, 2016).

Tabel 1. Penelitian Sejenis

Judul	Problem	Metode	Performa	Saran
<i>Fake News Detection Method Based on Text-Features</i> (Drif, 2019)	Media sosial atau portal berita yang memuat berita maupun informasi yang belum tentu valid	CNN dan RNN-LSTM	Model CNN-LSTM memiliki performa yang baik dalam mendeteksi berita palsu dengan akurasi 62,34% (dataset LIAR) dan 72,5% (dataset artikel)	Diusulkan penggunaan CNN-LSTM <i>hybrid</i> untuk peningkatan kinerja dan akurasi
<i>Online News Classification Using Deep Learning Technique</i> (Kaur, 2016)	Kendala dalam mengklasifikasikan ke dalam suatu kategori mengingat banyaknya berita yang dimuat	<i>Neural Network</i> (NN)	<i>Neural Network</i> (NN) memiliki akurasi yang tinggi sekitar 99,3%	Metode yang diusulkan memiliki akurasi yang lebih tinggi dibandingkan dengan metode <i>machine learning</i> tradisional
Klasifikasi Berita Online Menggunakan Metode <i>Support Vector Machine</i> dan <i>K-Nearest Neighbor</i> (Asiyah, 2016)	a. Proses pengolompokan suatu berita dilakukan manual b. Beberapa artikel yang dikategorikan memiliki kemiripan isi	<i>Support Vector Machine</i> (SVM) dan <i>K-Nearest Neighbor</i> (KNN)	SVM memiliki performa yang lebih baik daripada KNN dengan akurasi sebesar 93,2% dengan menggunakan kernel linier dan <i>polynomial</i> .	Perlu peningkatan performa pada kasus <i>multiclass</i> SVM karena hanya menggunakan metode <i>one against one</i> .
<i>Improving Nepali News Recommendation Using Classification Based on LSTM Recurrent Neural Networks</i> (Basnet, 2018)	a. Pengguna tidak dapat membaca artikel serta melewati beberapa kategori yang diminati b. Beberapa portal berita di Nepal menyajikan rekomendasi konten secara manual	RNN <i>Long Short-Term Memory</i> (LSTM)	LSTM memiliki kinerja yang lebih baik daripada SVM dengan tingkat akurasi mencapai 84,63% saat evaluasi dengan <i>Word2Vector</i>	Dalam studi lebih lanjut, perlu peningkatan situs web dan kategori untuk klasifikasi berita yang lebih banyak serta penggunaan <i>library stemming</i> untuk bahasa Nepal
<i>News Article Text Classification in Indonesian Language</i> (Wongso, 2017)	Banyak dikembangkannya klasifikasi dokumen namun mengalami keterbatasan dokumen untuk bahasa Indonesia	TF-IDF, <i>Multinomial Naive Bayes</i> , SVD	Kombinasi dari TF-IDF dan <i>Multinomial Naive Bayes</i> (MNB) memiliki nilai presisi dan <i>recall</i> yang tinggi mencapai 98,4%	Dalam studi lebih lanjut akan digunakan <i>training data set</i> ke dalam bahasa Indonesia serta kategori yang beragam

2. KAJIAN LITERATUR

2.1 Penelitian Sejenis

Bagian ini memuat perbandingan terhadap karya-karya ilmiah terkait sebelumnya dan dijadikan sebagai acuan untuk penelitian selanjutnya yang memiliki permasalahan sama atau serupa. Tabel 1 adalah penjelasan dari setiap penelitian yang serupa terkait analisis sentimen berita.

Berdasarkan matriks tinjauan pustaka dapat disimpulkan bahwa pemanfaatan algoritma *Deep Learning* untuk pengklasifikasi berita berbahasa Indonesia merupakan topik terkini yang sedang banyak dikerjakan oleh berbagai peneliti karena menunjukkan performa yang baik.

2.2. Klasifikasi

Klasifikasi merupakan pendekatan *supervised learning* pada *machine learning* di mana komputer mempelajari input data yang diberikan dan menggunakan hasil pembelajaran ini untuk mengklasifikasi observasi data uji. Klasifikasi adalah proses kategorisasi dataset yang diberikan. Hal ini dapat dilakukan pada data yang terstruktur maupun tidak. Proses diawali dengan melatih data latih dan memprediksi kelas dari data yang diberikan.. Tujuan utama klasifikasi adalah untuk mengidentifikasi atau mengkategorisasi kelas dari data uji yang diberikan (Waseem, 2019).

2.3. Long Short Term Memory (LSTM)

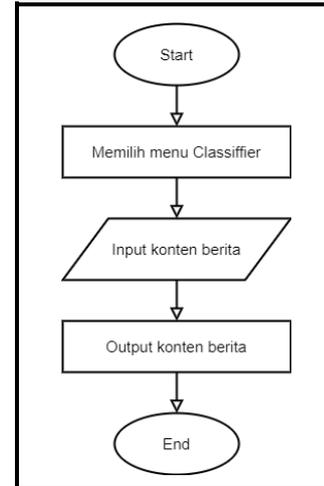
Jaringan Long Short-Term Memory (LSTM) adalah jenis jaringan syaraf berulang (RNN) yang mampu mempelajari dependensi dalam masalah urutan prediksi. Hal ini merupakan perilaku yang diperlukan dalam menyelesaikan masalah domain seperti translasi mesin, speech recognition, dan masih banyak lagi (Brownlee, 2017). LSTM memiliki tiga jenis gate antara lain *input gate*, *output gate*, dan *forget gate*.

3. PERANCANGAN SISTEM

Perancangan sistem monitoring sentimen berita ini digambarkan dalam bentuk flowchart yang terdiri dari dua bagian yaitu *flowchart process* dan *flowchart system*.

3.1. Flowchart Aplikasi

Gambar 1 merupakan alur penggunaan sistem klasifikasi dalam menghasilkan suatu output prediksi. Proses dimulai ketika pengguna mengakses menu *classifier*. Pengguna mengisi data melalui formulir prediksi yang tersedia di antarmuka. Lalu data masukan tersebut akan diproses sampai menghasilkan prediksi yang mungkin. Setelah itu, sistem akan menampilkan hasil prediksi dari setiap konten berita yang terklasifikasi.



Gambar 1. Flowchart Process pada penggunaan aplikasi

3.2. Flowchart Sistem

Gambar 2 merupakan perancangan *flowchart* sistem monitoring sentimen berita. Proses dimulai dengan melakukan *scraping* (*pengambilan*) data dari situs berita. Kemudian dilanjutkan ke tahap *pre-processing*, yang merupakan tahap pembersihan dan pemilihan data yang mewakili setiap kelas. Setelah itu, data dilatih menggunakan LSTM dan diakhiri dengan pengujian model. Nantinya, sistem akan menampilkan output berupa prediksi setiap konten dari hasil klasifikasi.

Selanjutnya adalah menetapkan analisis kebutuhan sistem. Analisis kebutuhan perangkat lunak menentukan apa yang harus dilakukan sistem dan mendefinisikan batasan batasan operasi dan implementasinya agar dapat mengomunikasikan secara tepat semua fungsi yang diberikan (Sommerville, 2011). Analisis kebutuhan dibagi menjadi dua bagian yaitu kebutuhan fungsional dan kebutuhan non-fungsional.

1. Kebutuhan Fungsional

Kebutuhan fungsional meliputi kebutuhan atau fitur apa saja yang akan disediakan sistem. Serta mencakup bagaimana sistem berinteraksi pada masukan tertentu serta perilaku sistem pada suatu kondisi. Adapun kebutuhan fungsional sistem yang dibutuhkan antara lain.

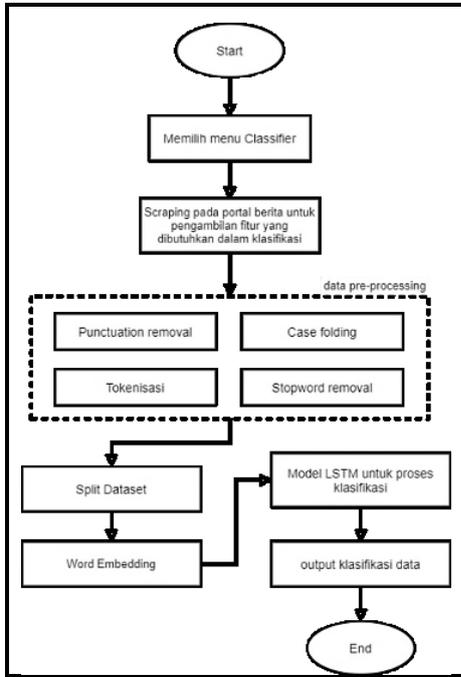
- Pihak Kemkominfo dapat melakukan input konten berita
- Pihak Kemkominfo dapat melihat hasil prediksi
- Sistem dapat menampilkan hasil prediksi kategori sentimen berita

2. Kebutuhan Non-Fungsional

Kebutuhan non-fungsional dilakukan untuk mengetahui spesifikasi kebutuhan sistem. Spesifikasi ini melibatkan spesifikasi *hardware/software* yang digunakan, kelayakan sistem yang dibuat, serta batasan layanan yang ditawarkan sistem. Berikut ini adalah kebutuhan non-fungsional pada sistem.

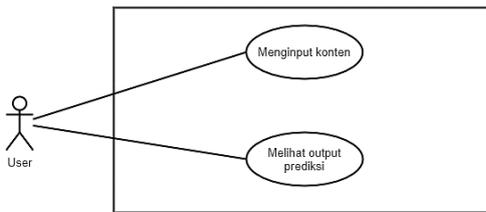
- Sistem bersifat responsif
- Sistem dapat digunakan dan bersifat user-friendly

Setelah menentukan analisis kebutuhan, selanjutnya adalah membuat rancangan arsitektur sistem dalam bentuk use case diagram yang bertujuan untuk menggambarkan alur sistem keseluruhan.



Gambar 2. Flowchart System pada aplikasi

Gambar 3 menunjukkan diagram use case sistem klasifikasi berita.



Gambar 3. Diagram use case sistem klasifikasi berita

Pada diagram ini (Gambar 3), terdapat satu aktor yang terlibat dalam melakukan interaksi dengan sistem yaitu pihak Kemkominfo. Aktivitas yang dilakukan antara lain mulai dari menginput konten berita yang akan diprediksi serta melihat hasil prediksi yang akan ditampilkan oleh sistem.

Berdasarkan perancangan yang telah ditentukan, kemudian dilanjutkan ke tahap implementasi. Implementasi ini dibagi menjadi dua tahap yaitu implementasi model dan aplikasi berbasis web.

3.3. Implementasi Model

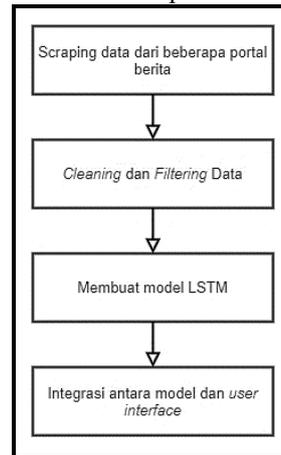
Pembuatan model dibagi menjadi beberapa tahap mulai dari tahap pengumpulan dataset dengan

scraping web, pembersihan dan pemilihan dataset ke dalam kelas tertentu untuk dijadikan sebagai data training, dan terakhir adalah pembuatan model LSTM untuk proses klasifikasi yang digambarkan pada Gambar 4.

Langkah-langkah implementasi dijelaskan sebagai berikut:

1. Pengumpulan dan Ekstraksi Data

Metode scraping adalah teknik pengambilan data berdasarkan informasi tertentu melalui website. Data yang digunakan berasal dari beberapa situs berita seperti Kompas, Detik, dan Tempo dengan kisaran waktu dari 2019 sampai bulan Mei 2020.



Gambar 4. Gambaran umum perancangan aplikasi

Hasil scraping ini selanjutnya akan disimpan dalam format .csv dengan mengambil fitur title, link, dan desc. Setelah itu, data akan dibersihkan untuk menghindari nilai null maupun duplikasi.

2. Penentuan Label Sentimen Berita

Pelabelan manual melibatkan manusia memberikan label data yang diperoleh. Pengklasifikasian ini mampu menghasilkan data yang akurat karena manusia dapat membedakan dengan tepat apakah kata atau kalimat tersebut bermakna. Dalam hal ini Kemkominfo merupakan agen yang dapat memvalidasi data konten yang telah dikumpulkan. Dari hasil observasi ini, nantinya diperoleh 3 kelas yang digunakan sesuai dengan kata yang terkait di setiap konten berita seperti pada Tabel 2 yaitu positif, negatif, dan netral.

Tabel 2. Daftar kelas di setiap klasifikasi

No	Nama Kelas	Deskripsi
1.	Positif	Berisi konten yang memberikan kesan positif bagi pembaca, misalnya apresiasi, prestasi, motivasi, maupun inspirasi
2.	Negatif	Berisi konten yang mengandung kesan negatif bagi pembaca, artinya bukan hanya berisi konten pornografi tetapi mengarah pada tindakan/aksi yang dilakukan pelaku
3.	Netral	Berisi konten yang tidak mempengaruhi orientasi pembaca

Tabel 3. Kata kunci yang digunakan pada setiap kelas

No	Nama Kelas	Kata Kunci
1.	Positif	['bantuan','damai','pelatihan','pening katan', 'inspirasi','kerjasama','puji']
2.	Negatif	['bugil','ganja','paksa','aniaya','tega', hadang', 'gelantungan', 'mesum','teror']
3.	Netral	['bacaan','dicurigai','puasa','mengkri tik', , 'sholat','ibadah','duga','saran']

Setelah menentukan kelas beserta kata kuncinya, maka akan dilakukan beberapa tahap seperti:

- Pemilihan data berdasarkan kata kunci (*terms*) (Tabel 3)
- Menyimpan data sesuai dengan nama dalam format .csv

3. Cleaning dan Filtering Data

Tahap cleaning merupakan proses pembersihan data yang tidak lengkap, tidak akurat, serta memperbaiki kualitas data ke dalam format yang sesuai dengan format yang digunakan. Proses cleaning dibagi menjadi dua tahap, antara lain:

- Case Folding*, untuk menyamakan karakter dengan mengubah semua teks menjadi huruf kecil (*lowercase*).
- Punctuation Removal*, untuk menghapus karakter selain alphanumeric maupun spasi berlebih

Setelah dibersihkan, lalu dipilih beberapa data yang mewakili setiap kelas. Hasil pemilihan data tersebut kemudian disimpan ke dalam file .csv sebagai data training yang terdiri dari konten data (*descriptions*) dan kelas. Di mana jumlah data training yang digunakan sebanyak 985 data.

4. Pembuatan Model LSTM

Pembuatan model LSTM dilakukan untuk melatih data latih dan mengklasifikasikannya ke dalam suatu kelas dari data latih. Pada tahap ini, data akan melalui beberapa tahap pra-pemrosesan di antaranya:

- Stopword Removal*, untuk menghapus kata-kata yang memiliki makna yang rendah, pada tahap ini telah didefinisikan kata-kata pada *stopword list* antara lain “namun”, “menurut”, “antara”, ”dia”, “dua”, “ia”, seperti”, “jika”, “jika”, “sehingga”.
- Tokenizer*, merupakan proses pemisahan dari satu kalimat ke dalam suatu kata atau makna tertentu ”dia”, “dua”, “ia”, seperti”, “jika”, “jika”, “sehingga”.
- Word Embedding*, merupakan representasi dari kata maupun teks ke dalam urutan angka (*sequences of integer*). Pada tahap ini dibuatlah

kamus data yang terdiri dari hasil tokenisasi dan indeks di setiap tokennya.

- Split Dataset*, membagi dataset menjadi data latih dan data validasi. Dari 985 data akan dibagi menjadi data latih sebanyak 886 data dan data validasi sebanyak 99 data.
- Pembuatan Model, pengembangan model yang melibatkan 3 lapisan di antaranya lapisan *embedding (hidden layer)*, lapisan LSTM (*input layer*), dan lapisan *dense (output layer)*.

5. Integrasi model LSTM dengan Aplikasi

Setelah model dibuat, kemudian diintegrasikan ke dalam aplikasi berbasis *web*. Proses ini dilakukan karena adanya interaksi di antara keduanya serta penggunaan kamus data. Proses integrasi dipicu pada saat pengguna mengirimkan data melalui formulir prediksi yang terdapat pada aplikasi *website*. Gambar 5 menunjukkan ringkasan model yang digunakan.

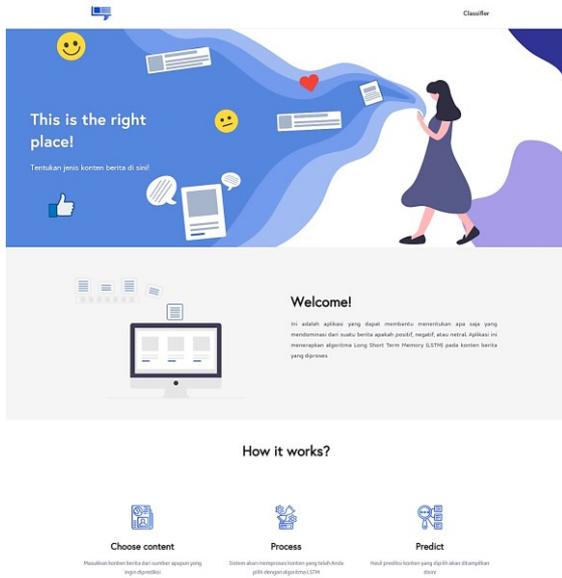
Layer (type)	Output Shape	Param #
embedding_1 (Embedding)	(None, 100, 300)	1050000
lstm_1 (LSTM)	(None, 100)	160400
dense_1 (Dense)	(None, 3)	303

Gambar 5. Ringkasan model yang digunakan

3.4. Implementasi Aplikasi

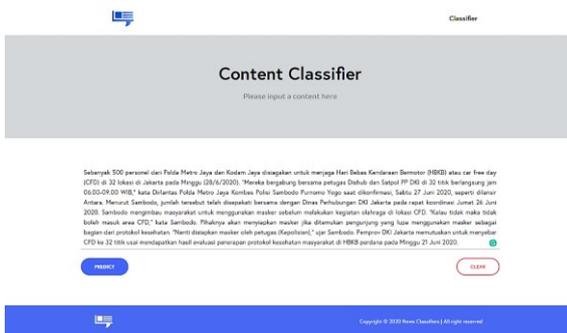
Sistem pengklasifikasi sentimen berita diimplementasikan dalam bentuk web menggunakan *Framework Flask* untuk konfigurasi *frontend* dan *API backend* menggunakan bahasa Python dan pustaka Keras. Flask adalah *web framework* yang menawarkan berbagai *tools*, pustaka, dan teknologi lainnya yang sesuai untuk membangun aplikasi *web*. Flask adalah salah satu *framework* terbaik karena memiliki *dependency* sedikit atau tidak sama sekali (Amigos-Maker, 2019). Python juga dikenal dengan bahasa pemrograman yang mudah dipelajari, karena struktur sintaksisnya rapi dan mudah dipahami. Keras adalah pustaka *open source* jaringan syaraf (neural network) dengan menggunakan bahasa Python yang dirancang secara modular, cepat, dan mudah digunakan.

Aplikasi ini dibagi menjadi beberapa modul utama antara lain *Home*, *Classifier*, dan *Result*. Tampilan halaman utama ditunjukkan pada Gambar 6.



Gambar 6. Halaman utama aplikasi

Gambar 7 menunjukkan tampilan halaman *content classifier*.



Gambar 7. Halaman *content classifier*

Sedangkan Gambar 8 menampilkan antarmuka halaman *result*.



Gambar 8. Halaman *Result*

4. HASIL DAN PEMBAHASAN

Pengujian yang dilakukan pada penelitian ini dibagi menjadi dua skenario yaitu pengujian model dan pengujian aplikasi website.

Pengujian model dilakukan dengan menggunakan metode *white-box testing* dan digunakan sebanyak 100 sampel data yang diambil secara acak. Hal ini bertujuan untuk menguji akurasi prediksi oleh model. Setelah dilakukan pengujian, terdapat 14 data yang hasilnya tidak sesuai dengan pelabelan manual. Hal ini disebabkan karena adanya kalimat yang masuk mengandung lebih dari satu kelas, pengaruh *word embedding*, dan nilai antar kelas yang berdekatan.

Contohnya pada kalimat ‘Lelaki warga Manado yang teridentifikasi berinisial HLK (52) ditemukan tewas tergantung dengan seutas tali nilon... Informasi yang dihimpun di lokasi kejadian menyebutkan HLK memilih mengakhiri hidupnya dengan gantung diri karena sakit hati istrinya selingkuh’ memiliki kemungkinan kelas netral. Namun, hasil prediksi model adalah kelas negatif karena model tidak dapat dilatih untuk memprediksi kombinasi kalimat di atas dengan tepat.

Proses *word embedding* dapat menghilangkan kata-kata tertentu termasuk kata yang memiliki makna. Hal ini menyebabkan data yang tidak ada pada proses pelatihan tidak dapat diprediksi dengan benar. Karena itu diperlukan solusi dengan menambah variasi data pada proses pelatihan.

Selain itu, nilai antar kelas yang berdekatan juga berpengaruh pada hasil prediksi. Artinya, nilai probabilitas yang ada pada suatu kelas memiliki nilai yang tidak jauh berbeda dengan nilai prediksi tertinggi. Misalnya pada kalimat “Vidya Rafika merasa beruntung Olimpiade 2020 ditunda setahun karena pandemi virus corona. Atlet menembak spesialis nomor 10 meter air rifle itu jadi mempunyai lebih banyak waktu untuk mempersiapkan diri bersaing dengan atlet asal Eropa dan Amerika...” di mana kemungkinan yang diharapkan adalah positif (0,439). Akan tetapi, model memprediksi data tersebut termasuk ke dalam kelas netral (0,504).

Kemudian lanjut ke tahap pembagian dataset menjadi data latih dan validasi. Jumlah data yang digunakan sebanyak 985 data yang terbagi menjadi 886 data latih dan 99 data untuk validasi. Data latih tersebut selanjutnya akan diuji dengan model LSTM. Untuk data validasi selanjutnya akan diuji dengan model yang telah dibuat, sehingga nanti diperoleh nilai *error/loss* dan akurasi pada model. Berikut adalah detail pembagian data yang terdapat pada Tabel 4.

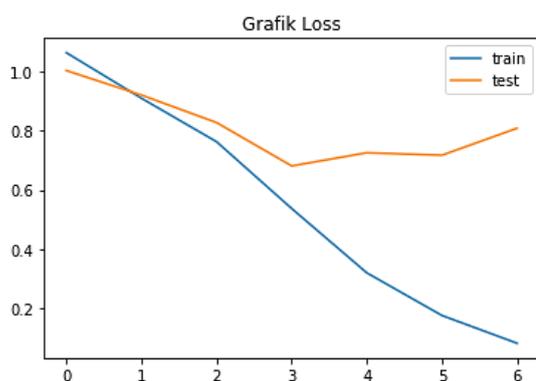
Tabel 4. Pembagian Data Pengujian

Jumlah Data	Data Training	Data Validasi
985 data	886 data	99 data

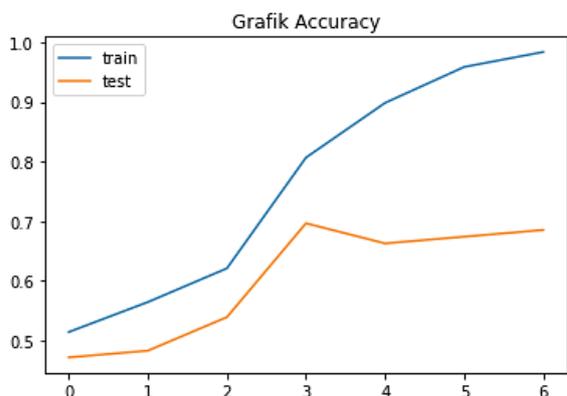
Pembuatan model LSTM mengimplementasikan model sekuen yang terdiri dari 3 lapisan (*layer*) antara

lain lapisan *embedding*, lapisan LSTM, dan lapisan *dense*. Lapisan *embedding* merupakan lapisan yang dapat menerima input berupa vektor 2D. Lapisan ini diinisialisasikan dengan bobot acak dan mempelajari semua kata dalam data. Parameter yang digunakan pada lapisan ini adalah *input_dim*, *output_dim*, dan *input_length*. Pada penelitian ini, jumlah *input_dim* disesuaikan dengan batas kata yang telah ditentukan ($MAX_NB_WORDS = 3500$ kata). Untuk *output_dim* sebanyak 300 dan *input_length* digunakan nilai 100.

Model tersebut kemudian dilatih dengan data training dan selanjutnya dievaluasi dengan data validasi. Adapun ukuran batch (*batch_size*) yang digunakan sebanyak 128 dengan *epoch* sebanyak 150. *Epoch* merupakan jumlah iterasi yang dilakukan saat melakukan training. Sedangkan *batch_size* merupakan jumlah data sampel yang dilatih pada suatu *batch*. Selain itu, terdapat fungsi *early stopping* guna menghentikan proses pelatihan data ketika terjadi overfit. Setelah dikompilasi, diperoleh nilai *loss* mencapai 0,528 dan nilai akurasi 0,8081. Hasil dari pelatihan ini juga ditampilkan dalam bentuk grafik seperti pada Gambar 9 dan Gambar 10.



Gambar 9. Grafik Loss



Gambar 10. Grafik Akurasi

Tahap analisis data bertujuan untuk mengevaluasi hasil pengujian yang telah dilakukan oleh model LSTM. Pengujian dilakukan terhadap 100 data yang diambil secara acak dan telah diberikan label. Kemudian dibuat perbandingan antara hasil

pelabelan manual dengan hasil pelabelan yang dilakukan oleh model. Dari perbandingan tersebut, sebanyak 14 dari 100 data testing yang memiliki klasifikasi yang tidak sesuai dengan kelasnya. Selanjutnya adalah menghitung nilai akurasi model. Perhitungan ini dilakukan dengan cara menghitung prosentase dengan membagi jumlah data yang diklasifikasikan benar dengan total data testing sehingga diperoleh nilai akurasi sebesar 86%. Dengan demikian model LSTM memiliki tingkat akurasi yang baik dan mampu mengklasifikasi sentimen berita berdasarkan konten dengan baik.

5. KESIMPULAN

Berdasarkan proses implementasi dan pembahasan sistem yang telah dilakukan di atas, dapat disimpulkan bahwa pengambilan dan pemrosesan data dari situs berita online berhasil dilakukan. Selain itu, penelitian ini menghasilkan sistem klasifikasi konten berita untuk memprediksi berita dengan menerapkan *Long Short-Term Memory (LSTM)*, *framework Flask*, dan pustaka Keras berbasis bahasa Python. Sementara itu, berdasarkan hasil dari pengujian menunjukkan bahwa model LSTM memiliki tingkat akurasi mencapai 86%. Di samping analisis sentimen, pemantauan juga perlu dilakukan terhadap kategori berita. Penelitian ke depan masih perlu dilakukan untuk mengategorisasikan jenis berita secara otomatis menggunakan LSTM atau metode lain yang sesuai. Selain itu penambahan dataset sangat diperlukan untuk meningkatkan akurasi sistem pengklasifikasi.

DAFTAR PUSTAKA

- AMIGOS-MAKER, 2019. DEV To: What is Flask. [Online] Tersedia: <<https://dev.to/amigosmaker/what-is-flask-1d63>> [Diakses 16 Juli 2020].
- ASIYAH, S. N. & FITHRIASARI, K., (2016). Klasifikasi Berita Online Menggunakan Metode Support Vector Machine dan K-Nearest Neighbor. *Jurnal Sains Dan Seni Its*, 5(2), pp. 317-322.
- BASNET, A. & TIMALSINA, A. K., (2018). Improving Nepali News Recommendation Using Classification Based on LSTM Recurrent Neural Networks. *IEEE 3rd International Conference on Computing, Communication, and Security (ICCCS)*, pp. 1-6.
- BROWNLEE, J., (2017). A Gentle Introduction to Long Short-Term Memory Networks by the Experts. [Online] Tersedia: <<https://machinelearningmastery.com/gentle-introduction-long-short-term-memory-networks-experts/>> [Diakses 16 Juni 2020].
- DRIF, A., HAMIDA, Z. F. & GIORDANO, S., (2019). Fake News Detection Method Based

on Text-Features. France, International Academy, Research, and Industry Association (IARIA), pp. 27-32.

Kementrian Komunikasi dan Informatika

Tersedia: <<https://www.kominfo.go.id>>
[Diakses 12 Juli 2020].

SOMMERVILLE, I., (2011). Software Engineering (Rekayasa Perangkat Lunak). 6 penyunting. Jakarta: Erlangga.

SOUMA, W., VODENSKA, I., AOYAMA, H. (2019) . Enhanced news sentiment analysis using deep learning methods, Journal of Computational Social Science. Springer Singapore, 2(1), pp. 33–46. doi: 10.1007/s42001-019-00035-x.

WASEEM, M., (2019). How To Implement Classification In Machine Learning?. Tersedia: <<https://www.edureka.co/blog/classification-in-machine-learning/>> [Diakses 18 Februari 2020].

WONGSO, R. ET AL., (2017). News Article Text Classification in Indonesian Language. 2nd International Conference on Computer Science and Computational Intelligence, ICCSCI, pp. 137-143.

ZHANG, L. & SONG, X., (2016). Theory, Methodology, Tools, and Applications for Modeling and Simulation of Complex Systems. 4 penyunt. Singapore: Springer + Business Media.