

KLASIFIKASI TEKS LAPORAN MASYARAKAT PADA SITUS LAPOR! MENGUNAKAN *RECURRENT NEURAL NETWORK*

Imam Fahrur Rozi, Vivi Nur Wijayaningrum, Nur Khozin

Jurusan Teknologi Informasi, Politeknik Negeri Malang

Jalan Soekarno Hatta 9, Malang

Email: imam.rozi@polinema.ac.id, vivinurw@polinema.ac.id, annurkhozin@gmail.com

(Diterima: 6 Agustus 2020, direvisi: 4 September 2020, disetujui: 9 September 2020)

ABSTRACT

The existence of public dissatisfaction with public services causes the public to be provided with facilities to make complaints. One of the sites that can be used to make complaints is the Public Service Complaint Management System (SP4N LAPOR!). With this site, complaints made by the public can be handled quickly, transparently and accountably in accordance with the authority of each organizer. However, the large number of complaints that had to be processed caused the process of data verification and sorting of reports by respective departments to take quite a long time, so the report classification process was needed to speed up the handling and follow-up of a report. The purpose of this research is to classify each complaint report from the public in preparation for the verification process of each public report document, which is expected to have an impact on the accelerated process of handling and follow-up of each related institution or agency. In this study, Long Short-Term Memory Recurrent Neural Network was used to perform the classification process for each public report document. The learning model is evaluated using k-fold cross-validation of 10 parts of data. The evaluation results show that the average f-measure percentage is 85.69% for the balanced dataset and 79.44% for the unbalanced dataset, while the highest evaluation value of all evaluations results in an f-measure of 88.82%. The high accuracy of the modeling indicates that the proposed method can be used to classify public report documents.

Keywords: *classification, complaint, long short-term memory, recurrent neural network, report*

ABSTRAK

Adanya ketidakpuasan masyarakat terhadap layanan publik menyebabkan masyarakat perlu disediakan fasilitas untuk melakukan pengaduan. Salah satu situs yang dapat digunakan untuk melakukan pengaduan adalah Sistem Pengelolaan Pengaduan Pelayanan Publik (SP4N LAPOR!). Dengan adanya situs ini, aduan yang dilakukan oleh masyarakat dapat ditangani dengan cepat, transparan, dan akuntabel sesuai dengan kewenangan masing-masing penyelenggara. Namun, banyaknya aduan yang harus diproses menyebabkan proses verifikasi data dan pemilahan laporan berdasarkan instansi masing-masing membutuhkan waktu yang cukup lama, sehingga proses klasifikasi laporan sangat dibutuhkan untuk mempercepat penanganan dan tindak lanjut dari sebuah laporan. Tujuan dari penelitian ini adalah mengklasifikasikan setiap laporan pengaduan dari masyarakat untuk persiapan proses verifikasi setiap dokumen laporan masyarakat, yang nantinya diharapkan dapat berdampak pada proses percepatan penanganan dan tindak lanjut dari setiap Lembaga atau instansi yang terkait. Pada penelitian ini, *Long Short-Term Memory Recurrent Neural Network* digunakan untuk melakukan proses klasifikasi setiap dokumen laporan masyarakat. Model pembelajaran dievaluasi menggunakan *k-fold cross-validation* sebanyak 10 bagian data. Hasil evaluasi menunjukkan rata-rata persentase *f-measure* sebesar 85,69% untuk *dataset* seimbang dan 79,44% untuk *dataset* tidak seimbang, sedangkan nilai evaluasi tertinggi dari semua evaluasi menghasilkan *f-measure* sebesar 88,82%. Akurasi pemodelan yang cukup tinggi menunjukkan bahwa metode yang diusulkan dapat digunakan untuk mengklasifikasikan dokumen laporan masyarakat.

Kata Kunci: *klasifikasi, pengaduan, long short-term memory, recurrent neural network, laporan*

1 PENDAHULUAN

Perkembangan Teknologi Informasi dan Komunikasi (TIK) yang sangat cepat sangatlah berpengaruh pada kehidupan sosial manusia di berbagai negara, salah satunya negara Indonesia. Indonesia mencoba memanfaatkan peran TIK dalam sektor pemerintahan, yaitu dengan membentuk sistem pengelolaan pengaduan pelayanan publik nasional (SP4N) - layanan aspirasi dan pengaduan online rakyat. Layanan penyampaian semua aspirasi dan pengaduan masyarakat Indonesia melalui beberapa kanal pengaduan yaitu website www.lapor.go.id, SMS 1708, twitter @lapor1708 dan aplikasi android [1].

Proses pelaporan pada SP4N LAPOR! diawali dengan registrasi pelapor, setelah itu pelapor melakukan pelaporan dengan cara mengisi *form* pelaporan dan memilih kategori laporan. Laporan yang telah dikirim membutuhkan waktu dalam 3 hari untuk proses verifikasi oleh administrator LAPOR! [2]. Portal SP4N LAPOR! dapat mempermudah pemerintah dalam hal pengawasan maupun pengontrolan pelayanan publik, serta mempermudah masyarakat untuk melakukan penyampaian laporan berupa aspirasi dan pengaduan kepada pemerintah. Kemudahan tersebut mengakibatkan banyak sekali laporan dalam bentuk teks dan juga jumlah laporan bertambah semakin cepat. Jumlah laporan yang telah masuk per Januari 2019 adalah sebanyak 1.389.891 laporan [1]. Jumlah data laporan yang bertambah setiap hari juga membutuhkan waktu cukup lama dalam mempersiapkan verifikasi data laporan [3]. Persiapan verifikasi data laporan dilakukan untuk bisa memilah laporan yang sesuai untuk diteruskan atau didisposisikan kepada instansi kementerian atau lembaga atau departemen terkait.

Penelitian [3] terkait yaitu tentang klasifikasi pengaduan pada sistem LAPOR! menggunakan metode C4.5 dengan menggunakan seleksi *fitur forward selection*. Penggunaan jumlah *dataset* tidak berpengaruh pada kenaikan akurasi pada klasifikasi dengan seleksi *fitur*. Dalam penelitian ini telah menghasilkan akurasi sebesar 85,27% [3]. Penelitian [4] lainnya yaitu tentang klasifikasi pengaduan pada sambat *online* menggunakan metode *n-gram* dan *neighbor weighted k-nearest neighbor* (NW-KNN). Dalam penelitian ini dilakukan perhitungan jarak terlebih dahulu menggunakan *cosine similarity* dan ekstraksi fitur menggunakan *n-gram*, kemudian dilakukan perhitungan jarak menggunakan metode NW-KNN. Dalam penelitian ini, nilai tetangga *k* terdekat yang paling optimal adalah 3, yang dapat menghasilkan nilai *f-measure* tertinggi sebesar 75.25% [4].

Berbeda dengan penelitian-penelitian tersebut, penelitian ini akan menggunakan pendekatan *natural language processing* atau sering disingkat sebagai NLP. NLP merupakan sebuah metode pembentukan model komputasi bahasa sebagai bentuk interaksi antara manusia dan komputer dengan perantara bahasa alami. NLP berupaya agar dapat memecahkan masalah untuk memahami bahasa alami manusia. NLP diyakini dapat memberikan pendekatan pada pemrosesan teks. Dalam pengerjaan tugas NLP terdapat salah satu metode yang dapat digunakan yaitu *Long Short-Term Memory* yang merupakan variasi dari permodelan *Recurrent Neural Network* atau sering disingkat RNN. RNN mampu memodelkan dependensi sekuensial pada data masukan dan memiliki sebuah memori data yang berisikan informasi yang dihasilkan sebelumnya, kemampuan ini disebabkan oleh adanya koneksi simpul (*recurrent*) pada model dan memproses data masukan secara sekuensial. 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 [5].

Berdasarkan hal-hal di atas mengenai klasifikasi teks laporan masyarakat berdasarkan instansi yang terkait pada di situs LAPOR! akan dilakukan menggunakan metode *Long Short-Term Memory Recurrent Neural Network*. Diharapkan dengan adanya penelitian ini dapat mempercepat proses verifikasi data laporan masyarakat, mengetahui kinerja dari metode *Recurrent Neural Network* dalam klasifikasi teks laporan masyarakat serta memperoleh hasil akurasi yang baik.

2 TINJAUAN PUSTAKA

2.1 Text Mining

Text mining atau *text analytics* adalah istilah yang mendeskripsikan sebuah teknologi yang mampu menganalisis data teks semi-terstruktur maupun tidak terstruktur, hal inilah yang membedakannya dengan data *mining* dimana data *mining* mengolah data yang sifatnya terstruktur. Pada dasarnya, *text mining* merupakan bidang interdisiplin yang mengacu pada perolehan informasi (*information retrieval*), data *mining*, pembelajaran mesin (*machine learning*), statistik, dan komputasi

Imam Fahrur Rozi dkk: Klasifikasi Teks Laporan Masyarakat pada Situs LAPOR! Menggunakan Recurrent Neural Network

linguistic [6]. Perbedaan mendasar antara *text mining* dan data *mining* terletak pada sumber data yang digunakan. Pada data *mining*, pola-pola diekstrak dari basis data yang terstruktur. Sedangkan pada *text mining*, pola-pola diekstrak dari data tekstual [7]. Diantara proses yang dapat dilakukan dalam *text mining* adalah klasifikasi teks. Klasifikasi teks dapat didefinisikan sebagai proses untuk menentukan suatu dokumen teks ke dalam suatu kelas tertentu. Untuk melakukan proses klasifikasi teks, ada beberapa algoritma yang dapat digunakan diantaranya *Support Vector Machine (SVM)*, *Naive Bayes*, *k-Nearest Neighbor (KNN)*, *Decision Tree*, dan *Artificial Neural Networks (ANN)* [8].

2.2 Preprocessing Text

Persiapan dokumen teks (*preprocessing text*) merupakan tahapan proses yang sangat penting dalam menentukan kualitas tahap selanjutnya yakni tahap klasifikasi [9]. Teks *online* biasanya mengandung banyak *noise* dan bagian tidak informatif seperti *tag HTML*, *script* dan iklan [10]. Tahap ini penting dalam memilih kata kunci yang signifikan dan membuang kata-kata yang tidak berkontribusi dalam membedakan antar dokumen. Tahap *preprocessing* mengonversi data tekstual dalam struktur data *mining* [8]. Dalam proses ini terdapat beberapa tahap pembersihan dokumen antara lain:

a. Case folding

Case folding merupakan proses pengubahan huruf dalam dokumen menjadi satu bentuk, misalnya huruf kapital menjadi huruf kecil dan sebaliknya [9].

b. Tokenizing

Tokenization merupakan proses pemisahan teks menjadi potongan kalimat dan kata yang disebut token [9].

c. Stemming

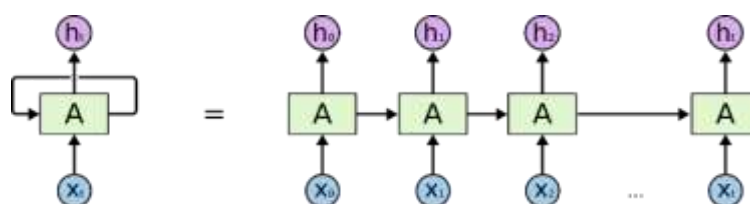
Salah satu algoritma untuk *stemming* adalah Algoritma Nazief & Adriani, algoritma ini dikembangkan pertama kali oleh Bobby Nazief dan Mirna Adriani. Algoritma ini berdasarkan pada aturan morfologi bahasa Indonesia yang luas, yang dikumpulkan menjadi satu grup dan di-enkapsulasi pada imbuhan atau *affixes* yang diperbolehkan (*allowed affixes*) dan imbuhan atau *affixes* yang tidak diperbolehkan (*disallowed affixes*). Algoritma ini menggunakan kamus kata dasar dan mendukung *recoding*, yakni penyusunan kembali kata-kata yang mengalami proses *stemming* berlebih [11].

d. Filtering

Filtering merupakan proses membuang atau menghapus kata-kata serta tanda-tanda yang tidak bermakna secara signifikan, seperti url, *emoticon*, tanda baca dan lainnya [9].

2.3 Recurrent Neural Network

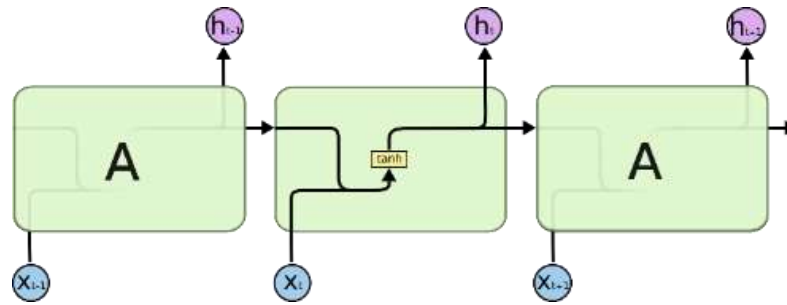
Recurrent neural network adalah bagian dari *neural network* yang memproses sejumlah data yang bersambung (*sequential data*). Secara umum, manusia tidak membuat keputusan dari awal setiap saat. Manusia selalu memperhitungkan informasi masa lalu dalam membuat keputusan. Seperti halnya manusia, *recurrent neural network* menyimpan informasi dari masa lalu dengan melakukan *looping* dalam arsitekturnya seperti ilustrasi dalam Gambar 1 yang secara otomatis membuat informasi dari masa lalu tetap tersimpan [5].



Gambar 1. Arsitektur RNN [5]

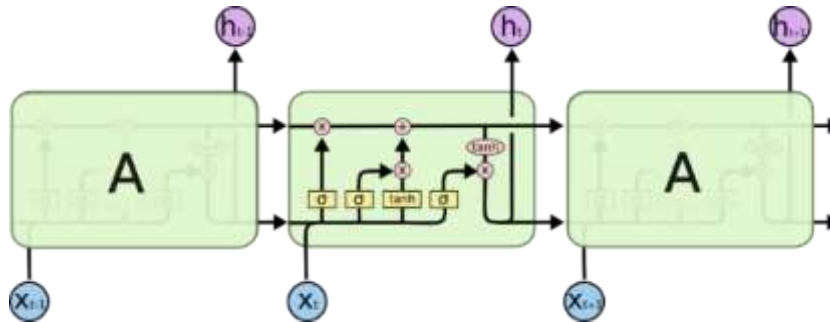
2.4 Long Short-Term Memory

Long Short-Term Memory (LSTM) adalah salah satu variasi dari *Recurrent Neural Network* yang dibuat untuk menghindari masalah ketergantungan jangka Panjang pada *Recurrent Neural Network (RNN)*. LSTM dapat mengingat informasi jangka panjang. Pada RNN perulangan jaringan hanya menggunakan satu *layer* sederhana [5], yaitu *layer tanh* seperti pada Gambar 2.



Gambar 2. Layer Pada Recurrent Neural Network [12]

Long Short-Term Memory (LSTM) juga mempunyai model seperti pada RNN, tetapi memiliki struktur yang berbeda, LSTM memiliki layer pada setiap perulangan [5], seperti pada Gambar 3.



Gambar 3. Layer Pada Setiap Cell LSTM [12]

Dalam setiap cell jaringan LSTM terdapat sebuah proses aliran data, aliran data tersebut dapat diwakilkan dalam notasi yang terdapat pada 4.



Gambar 4. Notasi Aliran Data Pada Cell LSTM [12]

LSTM memiliki 3 jenis gates diantaranya adalah *forget gate*, *input gate* dan *output gate*. *Forget gate* adalah gate yang memutuskan informasi mana yang akan dihapus dari cell. *Input gate* adalah gate yang memutuskan nilai dari input untuk diperbarui pada state memory. *Output gate* adalah gate yang memutuskan apa yang akan dihasilkan output sesuai dengan input dan memori pada cell [5].

Langkah pertama adalah LSTM memutuskan informasi apa yang akan dihapus dari cell state. Keputusan ini dibuat oleh sigmoid layer yang bernama “forget gate layer”. Forget gate layer akan memproses h_{t-1} dan X_t sebagai input, dan menghasilkan output berupa angka 0 atau 1 pada cell state C_{t-1} [5]. Persamaan forget gate layer dapat diuraikan dalam persamaan 1.

$$f_t = \sigma(W_f.[h_{t-1}, X_t] + b_f) \tag{1}$$

Dimana: f_t = forget gate σ = fungsi sigmoid
 W_f = nilai weight untuk forget gate h_{t-1} = nilai output sebelum orde ke t
 X_t = nilai input pada orde ke t b_f = nilai bias pada forget gate

Langkah kedua adalah memutuskan informasi apa yang akan disimpan di cell state. Untuk langkah ini terdapat dua bagian. Bagian pertama, sigmoid layer yang bernama “input gate layer” memutuskan nilai mana yang akan diperbarui. Selanjutnya, tanh layer membuat satu kandidat dengan nilai baru yaitu \tilde{C}_t , yang dapat ditambahkan ke cell state. Tahap selanjutnya adalah output dari input gate layer dan tanh layer akan digabungkan untuk memperbarui cell state [5]. Persamaan input gate dapat diuraikan pada persamaan 2.

$$i_t = \sigma(W_i.[h_{t-1}, X_t] + b_i) \quad (2)$$

Dimana: i_t = input gate σ = fungsi sigmoid
 W_i = nilai weight untuk input gate X_t = nilai input pada orde ke t
 b_i = nilai bias untuk input gate

Sedangkan, persamaan kandidat baru dapat diuraikan pada persamaan 3.

$$\tilde{C}_t = \tanh(W_c.[h_{t-1}, X_t] + b_c) \quad (3)$$

Dimana: \tilde{C}_t = nilai baru yang dapat ditambahkan ke cell state \tanh = fungsi tanh
 W_c = nilai weight untuk cell state h_{t-1} = nilai output sebelum orde ke t
 X_t = nilai input pada orde ke t b_c = nilai bias untuk cell state

Langkah ketiga adalah memperbarui cell state yang lama yaitu C_{t-1} , menjadi cell state baru yaitu C_t . Dengan mengkalikan state lama dengan f_t , untuk menghapus informasi yang sudah ditentukan sebelumnya pada langkah forget gate layer. Selanjutnya, ditambahkan dengan $i_t * \tilde{C}_t$, yang merupakan nilai baru dan digunakan untuk memperbarui nilai state [5]. Persamaan cell state baru dapat diuraikan pada persamaan 4.

$$C_t = f_t * C_{t-1} + i_t * \tilde{C}_t \quad (4)$$

Dimana: C_t = cell state f_t = forget gate
 C_{t-1} = cell state sebelum orde ke t i_t = input gate
 \tilde{C}_t = nilai yang dapat ditambahkan ke cell state

Langkah keempat adalah langkah terakhir dalam metode LSTM yang bertujuan untuk memutuskan hasil output. Output harus sesuai dengan cell state yang telah diproses terlebih dahulu. Pertama sigmoid layer memutuskan bagian dari cell state yang menjadi output. Selanjutnya, output dari cell state dimasukkan kedalam tanh layer (untuk mengganti nilai menjadi diantara -1 dan 1) dan dikalikan dengan sigmoid gate, agar output yang dihasilkan sesuai dengan apa yang kita putuskan sebelumnya [5]. Persamaan output gate diuraikan pada persamaan 5.

$$O_t = \sigma(W_o.[h_{t-1}, X_t] + b_o) \quad (5)$$

Dimana: O_t = output gate σ = fungsi sigmoid
 W_o = nilai weight untuk output gate h_{t-1} = nilai output sebelum orde ke t
 X_t = nilai input sebelum orde ke t b_o = nilai bias pada output gate

Sedangkan, persamaan nilai output orde ke t diuraikan pada persamaan 6.

$$h_t = O_t * \tanh(C_t) \quad (6)$$

Dimana: h_t = nilai output orde ke t O_t = output gate
 \tanh = fungsi tanh C_t = cell state

2.5 Tensorflow

TensorFlow merupakan open source library perangkat lunak yang dikembangkan oleh tim Google Brain dalam organisasi penelitian mesin cerdas google, tujuannya untuk melakukan pembelajaran mesin dan penelitian jaringan syaraf dalam. Tensorflow menggabungkan aljabar komputasi teknik pengoptimalan kompilasi untuk mempermudah perhitungan banyak ekspresi matematis [13].

3 METODE PENELITIAN

Pengumpulan data dilakukan dengan memanfaatkan data laporan masyarakat yang terdapat pada halaman laporan situs SP4N LAPOR! dengan periode januari 2015 – desember 2019, seperti pada Gambar 5. Pada halaman tersebut akan dilakukan scrape oleh sistem guna mengekstrak konten HTML yang mengandung informasi yang dibutuhkan dalam penelitian, seperti id pelapor, nama pelapor, judul laporan, tanggal laporan, isi laporan, link laporan dan instansi terdisposisi. Web scraping merupakan proses pengambilan sebuah dokumen-dokumen semi terstruktur dari internet, umumnya berupa halaman web dalam bahasa markup seperti html atau xhtml, dan menganalisa dokumen tersebut untuk diambil data tertentu dari halaman tersebut untuk digunakan bagi kepentingan lain [14].

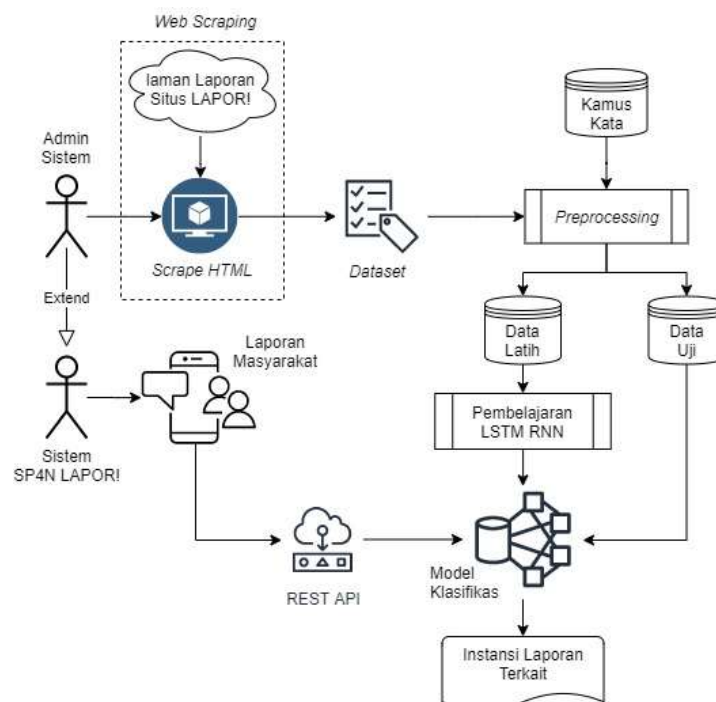
Imam Fahrur Rozi dkk: Klasifikasi Teks Laporan Masyarakat pada Situs LAPOR! Menggunakan Recurrent Neural Network



Gambar 5. Data Laporan Masyarakat

Pengambilan data dilakukan berdasarkan banyaknya halaman atau *pagination* pada halaman laporan SP4N LAPOR!, setiap *request scraping* akan memproses 1 halaman yang terdapat 10 data laporan masyarakat, pengambilan data dilakukan terus menerus sampai tidak menemukan data laporan lagi pada halaman tersebut atau diakhiri secara manual dari halaman *scraping* sistem. Data hasil *scraping* akan disimpan dan dijadikan *dataset* untuk proses pembuatan model klasifikasi pada penelitian ini.

Klasifikasi dilakukan sebelum proses verifikasi data laporan masyarakat yang masuk untuk mempercepat proses verifikasi setiap laporan yang akan di disposisikan oleh admin pengelola SP4N LAPOR!. Dalam sistem ini di mulai dari pengambilan data laporan masyarakat yang terdapat pada halaman laporan situs LAPOR! dengan melakukan *web scraping*, data tersebut dipilah oleh admin dan dijadikan sebagai *dataset* dalam pembentukan model klasifikasi. Dalam proses pembentukan model klasifikasi dilakukan *preprocessing* terlebih dahulu. Di dalam proses *preprocessing* terdapat beberapa tahapan, diantaranya *case folding*, *tokenizing*, *stemming*, *filtering*. Kemudian data dibagi menjadi data latih dan data uji. Untuk data latih akan dilakukan perhitungan menggunakan metode LSTM RNN untuk menghasilkan model klasifikasi, sedangkan data uji digunakan untuk mengetahui tingkat akurasi dari model klasifikasi yang dihasilkan oleh metode LSTM RNN tersebut. Setelah menemukan model klasifikasi yang baik menggunakan evaluasi pengujian *f-measure*, model klasifikasi digunakan oleh sistem SP4N LAPOR!, yaitu dengan melakukan *request web API* pada sistem dengan membawa sebuah parameter teks laporan masyarakat, maka sistem akan memberikan respon berupa klasifikasi teks laporan masyarakat tersebut berdasarkan instansi kementerian atau lembaga atau departemen terkait, untuk ilustrasi gambaran sistem terdapat pada Gambar 6.



Gambar 6. Gambaran Sistem

Imam Fahrur Rozi dkk: Klasifikasi Teks Laporan Masyarakat pada Situs LAPOR! Menggunakan Recurrent Neural Network

Kinerja sistem klasifikasi diukur dengan menggunakan metode *confusion matrix*. *Confusion matrix* mengandung informasi yang membandingkan hasil klasifikasi yang dilakukan oleh sistem dengan hasil klasifikasi yang seharusnya. Tempat istilah representasi hasil proses klasifikasi pada pengukuran kerja menggunakan *confusion matrix*, yaitu: *True Positives* (TP), *True Negatives* (TN), *False Positives* (FP), dan *False Negatives* (FN) [15], seperti yang terdapat pada Gambar 7.

		Predicted Class	
		P	N
Actual Class	P	True Positives (TP)	False Negatives (TN)
	N	False Positives (FP)	True Negatives (FN)

Gambar 7. *Confusion Matrix*

True Positives (TP) merupakan jumlah data positif yang terdeteksi benar, sedangkan *False Negatives* (FN) merupakan jumlah data positif yang terdeteksi sebagai data negatif. *True Negatives* (TN) merupakan jumlah data *negative* yang terdeteksi secara benar, sedangkan *False Positives* merupakan data negatif yang terdeteksi sebagai data positif [15].

Berdasarkan nilai *True Positives* (TP), *True Negatives* (TN), *False Positives* (FP), dan *False Negatives* (FN) dapat diperoleh nilai *precision*, *recall* dan *f-measure*. *Precision*, yang disebut juga *Positive Predictive Value* (PPV), merupakan jumlah data positif yang diklasifikasikan dengan benar dibagi dengan jumlah kelas bernilai positif yang diprediksi. *Precision* membantu jika nilai *false positives* tergolong tinggi. Seperti rumus yang dapat dilihat pada Persamaan 7, *precision* merupakan jumlah dari *True Positives* dibagi dengan jumlah *True Positives* (TP) dan *False Positives* (FP) [15].

$$Precision = \frac{TruePositives}{TruePositives + FalsePositives} \tag{7}$$

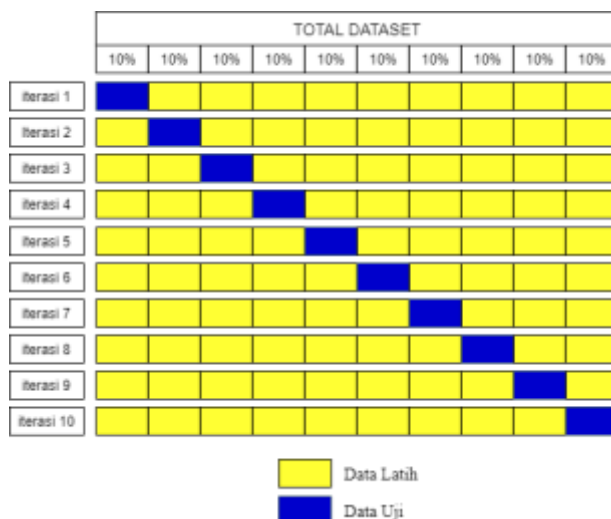
Recall, yang disebut juga *Sensitivity* atau *True Positive Rate*, menunjukkan persentase data kategori positif yang diprediksi dengan benar oleh sistem. *Recall* merupakan jumlah prediksi positif yang dibagi dengan jumlah kelas yang bernilai positif dalam data *testing*. *Recall* sangat membantu ketika nilai *false negatives* tergolong tinggi. Persamaan 8 menunjukkan persamaan untuk mendapatkan nilai *recall* [15].

$$Recall = \frac{TruePositives}{TruePositives + FalseNegatives} \tag{8}$$

F-measure, yang juga disebut *F1 score* atau *F-score*, merupakan sebuah ukuran akurasi dari *testing* yang digunakan untuk menghasilkan rata-rata harmonik berbobot dari *precision* dan *recall*. *F-measure* mencapai nilai terbaiknya, yang berarti *precision* dan *recall* yang sempurna, pada nilai 1. Sedangkan *F-measure* dengan *precision* dan *recall* terendah akan menjadi nilai 0 [15]. Persamaan *f-measure* dapat dilihat pada Persamaan 9.

$$F\ Measure = 2 * \frac{1}{\frac{1}{precision} + \frac{1}{recall}} \tag{9}$$

Validasi adalah salah satu proses pengujian suatu algoritma, pada umumnya pengujian dilakukan menggunakan data secara acak dengan beberapa kali pengujian hingga dihasilkan akurasi yang baik dari data tersebut. Pada validasi menggunakan *k-fold cross-validation* dilakukan pembagian *dataset* menjadi 2 bagian, yaitu data latih dan data uji dengan persentase pembagian yang sama pada setiap k. Seperti contoh yang akan diterapkan pembagian data latih sebanyak 90% dan data uji sebanyak 10% dari total keseluruhan *dataset* dengan membagi menjadi 10 bagian. Proses ini diulang sebanyak 10 kali hingga semua data mendapat bagian menjadi data uji, seperti yang terdapat pada Gambar 8.



Gambar 8. Representasi Dari 10-Fold Cross-Validation

Untuk evaluasi pengujian metode LSTM dilakukan menggunakan *k-fold cross-validation* dengan setiap *training* model menggunakan *hyperparameter batch size, epochs, optimizer* dan *activation*. *Hyperparameter tuning* yang digunakan pada penelitian ini terdapat pada Tabel 1.

Tabel 1. Hyperparameter Tuning

Kasus	Parameter			
	Batch Size	Epoch	Optimizer	Activation
Dataset tidak seimbang	512	10	Adam	Softmax
Dataset seimbang	512	10	Adam	Softmax

4 HASIL DAN PEMBAHASAN

Untuk mengetahui kinerja dari sistem yang telah dibuat, terdapat beberapa jenis percobaan yang dilakukan antara lain percobaan yang berkaitan dengan *dataset* dan percobaan REST API. Percobaan *dataset* digunakan untuk mengetahui perbedaan nilai *f-measure* yang dihasilkan dengan berbagai jenis *dataset*. Selanjutnya percobaan REST API dilakukan untuk memastikan semua layanan yang disediakan oleh sistem dapat berjalan dengan baik.

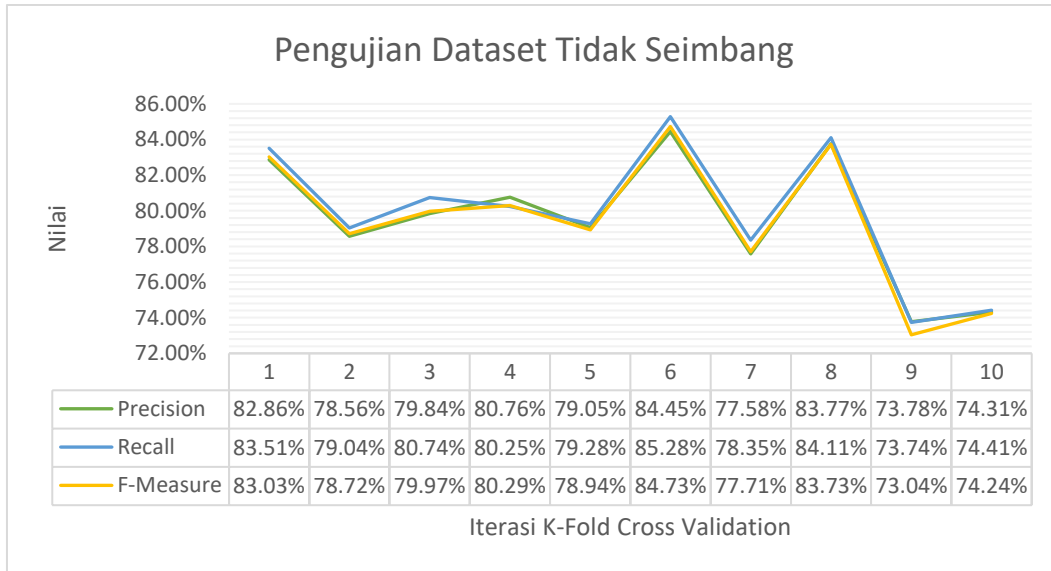
4.1 Percobaan menggunakan dataset tidak seimbang

Percobaan pertama dilakukan pada kasus *dataset* tidak seimbang, yaitu dengan menggunakan jumlah data yang tidak sama pada setiap label atau kelas, sehingga terdapat beberapa kelas yang lebih dominan pada *dataset*. Pengujian ini dilakukan untuk mengukur kinerja dari model yang dihasilkan saat dihadapkan dengan *dataset* tidak seimbang, jumlah data setiap kelas dapat dilihat pada Tabel 2.

Tabel 2. Perbandingan Dataset Tidak Seimbang

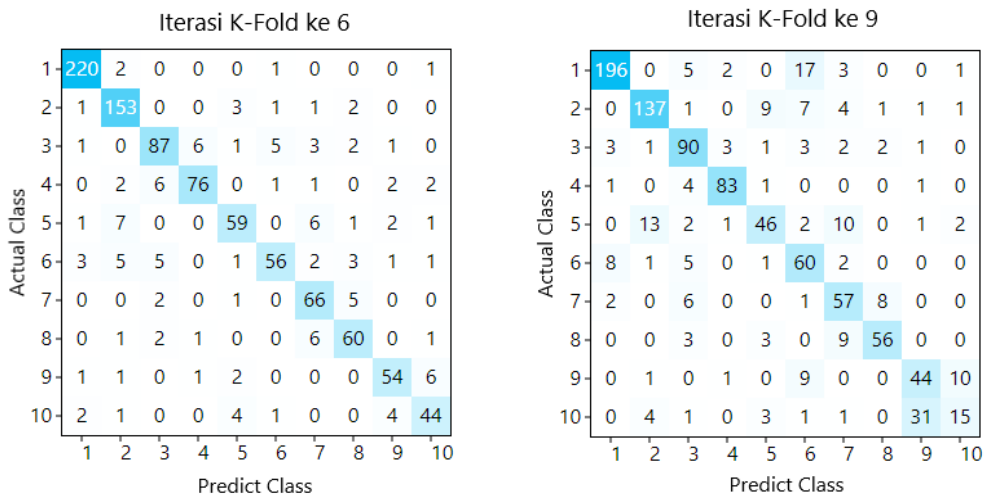
No	Nama Kelas	Jumlah
1	Direktorat Jenderal Imigrasi	2239
2	Kementerian Pendidikan dan Kebudayaan	1607
3	Kepolisian Negara Republik Indonesia	1055
4	Otoritas Jasa Keuangan	895
5	Kementerian Sosial	773
6	Kementerian Dalam Negeri	771
7	PT PLN (Persero)	740
8	Kementerian Energi dan Sumber Daya Mineral (ESDM)	712
9	Badan Penyelenggara Jaminan Sosial Ketenagakerjaan	652
10	Badan Penyelenggara Jaminan Sosial Kesehatan	556
Jumlah		10.000

Metode pengujian dilakukan dengan menggunakan metode pengujian *f-measure* yang dapat menghasilkan nilai *precision*, *recall* dan *f-measure* dengan menggunakan validasi *k-fold* 10 bagian data, dengan rata-rata nilai *precision* 79,50%, nilai *recall* 79,87% dan nilai *f-measure* 79,44%. Pada pengujian ini, nilai tertinggi terdapat pada iterasi *k-fold* ke 6 dengan nilai *precision* 84,45%, nilai *recall* 85,28% dan nilai *f-measure* 84,73%, sedangkan nilai terendah terdapat pada iterasi *k-fold* ke 9 dengan nilai *precision* 73,78%, nilai *recall* 73,74% dan nilai *f-measure* 73,04%, seperti yang terdapat pada Gambar 9.



Gambar 9. Grafik Pengujian Dataset Tidak Seimbang

Berdasarkan hasil pengujian setiap iterasi *k-fold* pada Gambar 9, akan dilakukan perbandingan data pada hasil pengujian *f-measure* tertinggi yang terdapat pada iterasi *k-fold* ke 6 dan hasil pengujian *f-measure* terendah yang terdapat pada iterasi *k-fold* ke 9. Tujuan dari perbandingan ini adalah mencari sebab nilai *f-measure* yang terpaut 11,69% dari nilai terendah ke nilai yang tertinggi. *Confusion matrix* nilai tertinggi terdapat pada iterasi *k-fold* ke 6, pada iterasi ini terdapat sedikit penyebaran data yang bernilai *true negative* dan *false negative*, sedangkan nilai terendah terdapat pada iterasi *k-fold* ke 9, pada iterasi ini terjadi lebih banyak penyebaran data yang bernilai *true negative* dan *false negative*, jika dilihat dari contoh data terendah yaitu pada iterasi *k-fold* ke 9. Hal ini disebabkan oleh adanya data uji yang bersifat ambigu, selain itu banyak kata singkatan, kata daerah dan juga mempunyai panjang kalimat lebih dari rata-rata dari data latih, selain itu faktor jumlah *dataset* yang berbeda pada setiap kelas juga mempengaruhi hasil pengujian setiap nilai *f-measure*, perbandingan tersebut terdapat pada Gambar 10.



Gambar 10. Confusion Matrix Dataset Tidak Seimbang

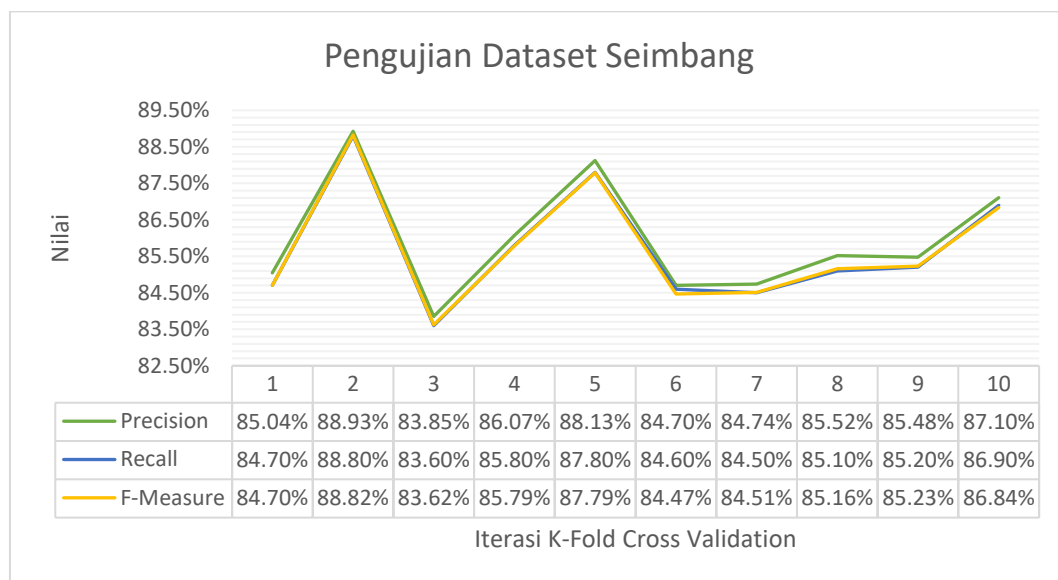
4.2 Percobaan menggunakan *dataset* seimbang

Percobaan kedua dilakukan pada kasus *dataset* seimbang, yaitu dengan menggunakan jumlah data yang sama pada setiap label atau kelas. Pengujian ini dilakukan untuk mengukur kinerja dari model yang dihasilkan saat dihadapkan dengan jumlah *dataset* yang sama antar kelas, jumlah data setiap kelas dapat dilihat pada Tabel 3.

Tabel 3. Perbandingan Jumlah *Dataset* Seimbang

No	Nama Kelas	Jumlah
1	Badan Penyelenggara Jaminan Sosial Ketenagakerjaan	1000
2	Kementerian Dalam Negeri	1000
3	Kementerian Pendidikan dan Kebudayaan	1000
4	Kepolisian Negara Republik Indonesia	1000
5	PT PLN (Persero)	1000
6	Otoritas Jasa Keuangan	1000
7	Badan Penyelenggara Jaminan Sosial Kesehatan	1000
8	Direktorat Jenderal Imigrasi	1000
9	Kementerian Energi dan Sumber Daya Mineral (ESDM)	1000
10	Kementerian Sosial	1000
Jumlah		10.000

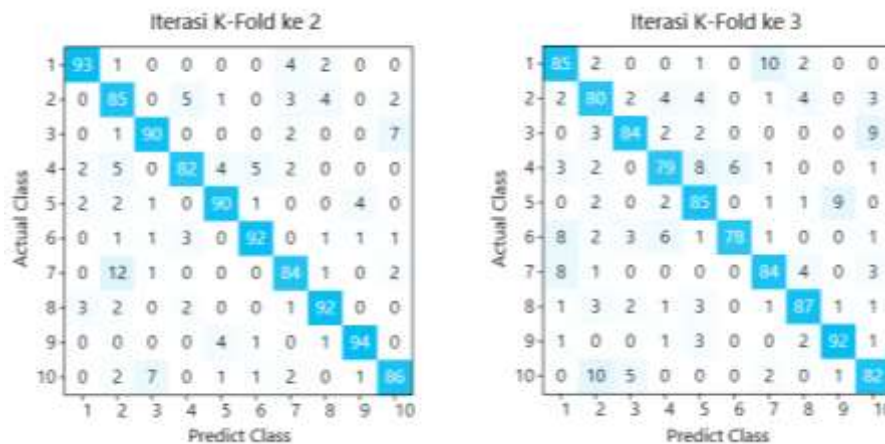
Metode pengujian percobaan kedua juga dilakukan dengan menggunakan metode pengujian *f-measure*. Pengujian tersebut mendapatkan nilai *precision*, *recall* dan *f-measure* dengan menggunakan validasi *k-fold* 10 bagian data, dengan rata-rata nilai *precision* 85,96%, nilai *recall* 85,70% dan nilai *f-measure* 85,62%. Pada pengujian ini, nilai tertinggi terdapat pada iterasi *k-fold* ke 2 dengan nilai *precision* 88,93%, nilai *recall* 88,80% dan nilai *f-measure* 88,82%, sedangkan nilai terendah terdapat pada iterasi *k-fold* ke 3 dengan nilai *precision* 83,85%, nilai *recall* 83,60% dan nilai *f-measure* 83,62%, seperti yang terdapat pada grafik Gambar 11.



Gambar 11. Grafik Hasil Pengujian *Dataset* Seimbang

Berdasarkan hasil pengujian setiap iterasi *k-fold* pada Gambar 11, akan dilakukan perbandingan data pada hasil pengujian *f-measure* tertinggi yang terdapat pada iterasi *k-fold* ke 2 dan hasil pengujian *f-measure* terendah yang terdapat pada iterasi *k-fold* ke 3. Tujuan dari perbandingan ini adalah mencari sebab nilai *f-measure* yang terpaut 5,20% dari nilai terendah ke nilai yang tertinggi. *Confusion matrix* nilai tertinggi terdapat pada iterasi *k-fold* ke 2, pada iterasi ini terdapat sedikit penyebaran data yang bernilai *true negative* dan *false negative*, sedangkan nilai terendah terdapat pada

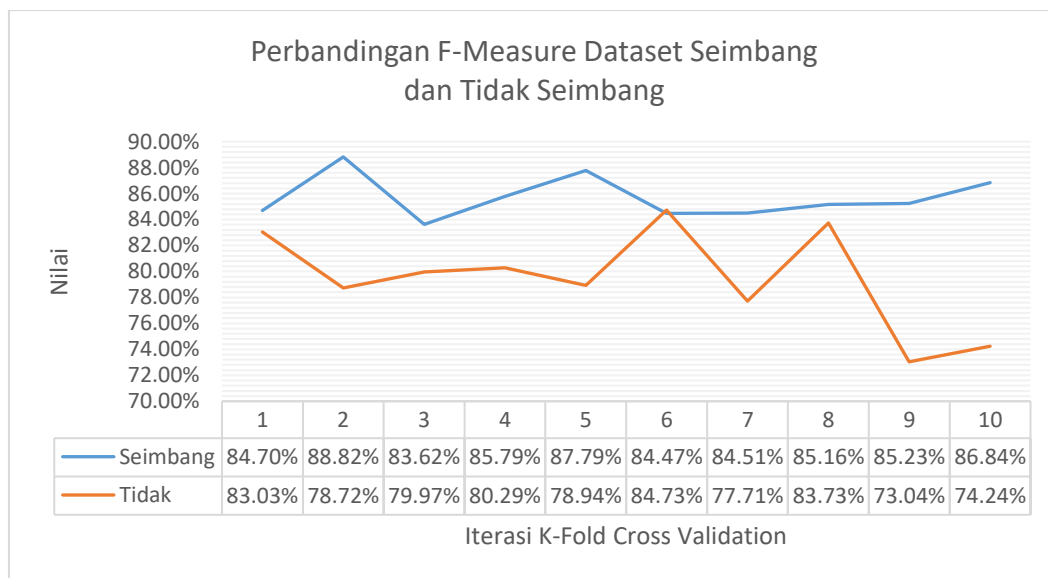
iterasi *k-fold* ke 3, pada iterasi terjadi lebih banyak penyebaran data yang bernilai *true negative* dan *false negative*, perbandingan *confusion matrix* terdapat pada 12.



Gambar 12. Confusion Matrix Dataset Seimbang

4.3 Perbandingan Bentuk Dataset

Untuk menghasilkan hasil yang maksimal di dalam proses klasifikasi data harus memiliki distribusi yang sama dengan data pelatihan. Namun, kenyataannya data seperti ini tidak selalu ditemukan, banyak juga data yang distribusinya tidak sama, dimana satu kelas mungkin diwakili oleh data dengan jumlah yang besar, sementara kelas yang lain diwakili oleh hanya beberapa data saja, hal ini dapat mempengaruhi hasil dari klasifikasi. Berdasarkan pengujian terhadap *dataset* seimbang diambil nilai *f-measure* tertinggi sebesar 88,82% yang terdapat pada iterasi *k-fold* ke 2, dan nilai *f-measure* tertinggi terhadap *dataset* tidak seimbang sebesar 84,73% yang terdapat pada iterasi *k-fold* ke 6, kedua nilai terdapat selisih 4,10%, seperti yang terdapat pada Gambar 13.



Gambar 13. Perbandingan Bentuk Dataset

4.4 Percobaan REST API

Postman adalah sebuah aplikasi yang berfungsi sebagai *REST CLIENT* untuk uji coba *REST API*. *Postman* biasa digunakan oleh *developer* pembuat API sebagai *tools* untuk menguji API yang telah dibuat, pengujian dalam penelitian ini dilakukan dengan melakukan *request* API dengan memasukkan URI (*ip/domain*) target dengan menggunakan metode *POST*, dalam *request* tersebut membawa parameter teks yang berisi teks laporan yang akan dilakukan prediksi kemudian mengirim *request*. Setelah sistem memberikan *respons*, pengujian mendapatkan *respons* berupa data *json*, data tersebut berisi teks awal, nilai prediksi, daftar kelas dan kelas prediksi. Kelas prediksi diambil berdasarkan

Imam Fahrur Rozi dkk: Klasifikasi Teks Laporan Masyarakat pada Situs LAPOR! Menggunakan Recurrent Neural Network

nilai prediksi yang mempunyai nilai tertinggi, setiap nilai prediksi mewakili peluang berdasarkan daftar kelas atau label klasifikasi, nilai tertinggi menandakan peluang paling besar dari klasifikasi tersebut, sehingga diambil juga kelas atau label berdasarkan posisi nilai prediksi yang tertinggi tersebut, hasil dari API terdapat pada Tabel 4.

Tabel 4. Hasil Pengujian Kelas Terdaftar

URI	http://malangdev.ddns.net/api/klasifikasi
Method	POST
Request body: teks	Saya mau tanya, apakah bisa saya meminta nomor kpj bpjstc saya? saya sudah lama resign dan sudah minta ke hrd utk nomor kpj bpjs blm ada tanggapan, apakah disini bisa meminta informasi tsb.
Nilai prediksi	[0.0028434619307518005, 0.00009904309990815818, 0.0000000386257397622, 0.001121978391893208, 0.000013629541172122117, 0.9816818833351135 , 0.000004221413746563485, 0.014232831075787544, 0,000000008747363722250, 0.000002458072685840]
Kelas sebenarnya	Badan Penyelenggara Jaminan Sosial Ketenagakerjaan
Kelas prediksi	Badan Penyelenggara Jaminan Sosial Ketenagakerjaan
Link LAPOR!	https://www.lapor.go.id/laporan/detil/tidak-tahu-nomor-kpj-bpjs

Pengujian juga dilakukan menggunakan kelas data laporan masyarakat yang tidak terdaftar pada model, hal ini dilakukan untuk mengetahui hasil dari model untuk mengenali semua jenis laporan yang akan diproses, dari hasil pengujian yang telah dilakukan menghasilkan kelas instansi Badan Penyelenggara Jaminan Sosial Kesehatan, sedangkan kelas sebenarnya adalah instansi RSUD H. Badaruddin Tanjung. Hal ini disebabkan karena dalam instansi Badan Penyelenggara Jaminan Sosial Kesehatan juga mempunyai wewenang untuk menindaklanjuti laporan tersebut, hanya saja kelas instansi RSUD H. Badaruddin Tanjung lebih spesifik dalam menindaklanjuti laporan masyarakat tersebut, akan tetapi instansi RSUD H. Badaruddin Tanjung tidak terdaftar dalam sistem, maka sistem memberikan kelas target instansi laporan tersebut ke Badan Penyelenggara Jaminan Sosial Kesehatan, hasil pengujian terdapat pada Tabel 5.

Tabel 5. Pengujian Kelas Tidak Terdaftar

URI	http://malangdev.ddns.net/api/klasifikasi
Method	POST
Request body: teks	Kami mandangar habar di rs maburai ada pegawai yang kana corona, kami yang handak beobat jadi takut, kira-kira kami aman kah apabila beobat ka rumah sakit? #corona.
Nilai prediksi	[0.003222286468371749, 0.13194610178470612, 0.0031970287673175335, 0.2273382693529129, 0.0016881052870303392, 0.039412423968315125, 0.17141129076480865, 0.402165949344635 , 0.0037529265973716974, 0.015865672379732132]
Kelas sebenarnya	RSUD H. Badaruddin Tanjung
Kelas prediksi	Badan Penyelenggara Jaminan Sosial Kesehatan
Link LAPOR!	https://www.lapor.go.id/laporan/detil/aman-ke-rumah-sakit

5 KESIMPULAN

Berdasarkan hasil penelitian dan pengujian yang telah dilakukan dapat ditarik kesimpulan yang pertama, dalam pengumpulan data laporan masyarakat dilakukan menggunakan proses *scraping* pada situs LAPOR! dengan memperhatikan laporan yang sudah terverifikasi dan sudah berstatus selesai, serta dapat mengambil laporan berdasarkan periode yang dibutuhkan. Kedua, dalam proses klasifikasi teks laporan masyarakat dapat dilakukan dengan mengirim sebuah *request* ke REST API sistem yang telah dibuat pada penelitian. Ketiga, dalam implementasi metode *long short-term memory* (LSTM) *recurrent neural network* (RNN) dapat digunakan untuk proses klasifikasi laporan masyarakat

Imam Fahrur Rozi dkk: Klasifikasi Teks Laporan Masyarakat pada Situs LAPOR! Menggunakan Recurrent Neural Network

berdasarkan instansi yang terkait, dimana pengujian dilakukan menggunakan *k-fold cross-validation* sebanyak 10 bagian data, pengujian tersebut menghasilkan rata-rata persentase *f-measure* sebesar 85,69% untuk *dataset* seimbang dan 79,44% untuk *dataset* tidak seimbang, sedangkan dari total keseluruhan pengujian menghasilkan *f-measure* tertinggi sebesar 88,82%.

Pengembangan sistem yang dapat dilakukan penelitian selanjutnya adalah memperbaiki tahap *preprocessing* dengan menambah kamus kata yang berisikan kata singkatan, kata daerah dan kata gaul. Selain itu pada penelitian ini hanya menggunakan 10 instansi, sehingga untuk penelitian selanjutnya diharapkan dapat menggunakan lebih banyak instansi lagi, untuk mengetahui kestabilan kinerja metode LSTM RNN dalam klasifikasi teks laporan masyarakat.

REFERENSI

- [1] LAPOR!, "Tentang LAPOR!," Kantor Staf Presiden, 2019. [Online]. Available: <https://www.lapor.go.id/tentang>. [Accessed 23 December 2019].
- [2] LAPOR!, "Beranda LAPOR!," Kantor Staf Presiden, 2019. [Online]. Available: <https://www.lapor.go.id/>. [Accessed 23 December 2019].
- [3] A. Sofyan and S. Santosa, "Text Mining Untuk Klasifikasi Pengaduan Pada Sistem Laporan Menggunakan Metode C4.5 Berbasis Forward Selection," *Jurnal Teknologi Informasi*, pp. 74-83, 2016.
- [4] A. A. Prasanti, M. A. Fauzi and M. T. Furqon, "Klasifikasi Teks Pengaduan Pada Sambat Online Menggunakan Metode N-Gram dan Neighbor Weighted K-Nearest Neighbor (NW-KNN)," *Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer*, pp. 594-601, 2018.
- [5] I. Habibie, "Identifikasi Judul Berita Clickbait Berbahasa Indonesia Dengan Algoritma Long Short Term Memory (LSTM) Recurrent Neural Network," Universitas Sumatera Utara, Medan, 2018.
- [6] C. Megawati, "Analisis Aspirasi Dan Pengaduan Di Situs LAPOR! Dengan Menggunakan Text Mining," Universitas Indonesia, Depok, 2015.
- [7] D. Yuliana, Purwanto and C. Supriyanto, "Klasifikasi Teks Pengaduan Masyarakat Dengan Menggunakan Algoritma Neural Network," *KomTekInfo*, vol. V, pp. 92-116, 2019.
- [8] A. F. Hidayatullah and M. R. Ma'arif, "Penerapan Text Mining dalam Klasifikasi Judul Skripsi," *Seminar Nasional Aplikasi Teknologi Informasi*, pp. 33-36, 2016.
- [9] F. Handayani and F. S. Pribadi, "Implementasi Algoritma Naive Bayes Classifier dalam Pengklasifikasian Teks Otomatis Pengaduan dan Pelaporan Masyarakat melalui Layanan Call Center 110," *Jurnal Teknik Elektro*, pp. 19-24, 2015.
- [10] E. Indrayuni, "Klasifikasi Text Mining Review Produk Kosmetik Untuk Teks Bahasa Indonesia Menggunakan Algoritma Naive Bayes," *Khatulistiwa Informatika*, vol. VII, pp. 29-36, 2019.
- [11] D. Wahyudi, T. Susyanto and D. Nugroho, "Implementasi Dan Analisis Algoritma Stemming Nazief & Adriani Dan Porter Pada Dokumen Berbahasa Indonesia," *Jurnal Ilmiah SINUS*, pp. 49-56, 2017.
- [12] C. Olah, "Understanding LSTM Networks," Github, 27 August 2015. [Online]. Available: <https://colah.github.io/posts/2015-08-Understanding-LSTMs/>. [Accessed 15 December 2019].
- [13] P. N. Rena, "Penerapan Metode Convolutional Neural Network Pada Pendeteksi Gambar Notasi Balok," UIN Syarif Hidayatullah, Jakarta, 2019.
- [14] A. Oktora, "Implementasi Web Scraping Pada Indexing E-journal Universitas Islam Negeri (UIN)," Universitas Islam Negeri Sultan Syarif Kasim Riau, Pekanbaru, 2018.
- [15] Silvin, "Analisis Sentimen Media Twitter Menggunakan Long Short-Term Memory Recurrent Neural Network (Studi Kasus: Prediksi Polling Kandidat Presiden Indonesia 2019)," Universitas Multimedia Nusantara, Tangerang, 2019.