

Sentiment Analysis On Reviews Of Beach Tourism Objects On Google Maps Using Long-Short Term Memory Method

Niken Oktaviana¹, Heru Cahya Rustamaji², Herry Sofyan³

^{1,2,3} Informatika, Universitas Pembangunan Nasional Veteran Yogyakarta, Indonesia

¹nikenoktaviana@gmail.com, ^{2*}herucr@upnyk.ac.id, ^{3*}herry.s@upnyk.ac.id

*: *Penulis korespondensi (corresponding author)*

Keywords: classification, sentiment analysis, deep learning, LSTM

Kata kunci: klasifikasi, analisis sentimen, *deep learning*, LSTM

Abstract

Purpose: Can find out the performance level of the Long-Short Term Memory algorithm in conducting sentiment analysis on beach reviews in Yogyakarta.

Design/methodology/approach: This research uses Long-Short Term Memory (LSTM) algorithm with word2vec to handle sequential data. The data used are sourced from google maps reviews and are labeled manually. The process carried out is to collect data, preprocessing, sentence conversion, word2vec, then do the classification. Finally, testing using confusion matrix and ROC curves to find the accuracy, precision, recall, and AUC values. Tests use the percentage of training data 80% and test data 20% with the K-fold cross validation.

Findings/result: The test results with confusion matrix produce an average accuracy value of 84%, 76% precision, and a 0.73% recall for the sentiment model. Meanwhile, for the category model, an average accuracy of 76%, 75% precision and 0.74% recall is obtained. The test results with the ROC curve obtained an average AUC value of 0.73 (fair classification) for the sentiment model. Meanwhile, for the category model, the AUC average score was 0.83 (good classification).

Originality/value/state of the art: This research implements one of the deep learning algorithms LSTM with word embedding weighting method to test the performance of text classification. The test carried out is to compare the performance of the LSTM implementation for binary class and multiclass classifications.

Abstrak

Tujuan: Dapat mengetahui tingkat performa algoritma Long-Short Term Memory dalam melakukan analisis sentimen ulasan pantai di Yogyakarta.

Perancangan/metode/pendekatan: Penelitian ini menggunakan algoritma *Long-Short Term Memory* (LSTM) dengan kombinasi *word2vec* untuk menangani data *sequential*. Data yang digunakan bersumber dari ulasan

google maps dan diberi label secara manual. Proses yang dilakukan adalah mengumpulkan data, *preprocessing*, *sentence conversion*, *word2vec*, kemudian dilakukan klasifikasi. Terakhir, dilakukan pengujian menggunakan *confusion matrix* dan kurva ROC untuk mencari nilai akurasi, presisi, *recall*, dan nilai AUC. Pengujian menggunakan persentase data latih 80% dan data uji 20% dengan pendekatan *K-fold cross validation*.

Hasil: Hasil pengujian dengan *confusion matrix* menghasilkan nilai akurasi rata-rata 84%, presisi 76%, dan *recall* 0.73% untuk model sentimen. Sedangkan untuk model kategori diperoleh akurasi rata-rata sebesar 76%, presisi 75%, dan *recall* 0.74%. Hasil pengujian dengan kurva ROC diperoleh nilai rata-rata AUC 0.73 (*fair classification*) untuk model sentimen. Sedangkan untuk model kategori diperoleh nilai rata-rata AUC 0.83 (*good classification*).

Keaslian/ *state of the art*: Penelitian ini mengimplementasikan salah satu algoritma *deep learning* yaitu LSTM dengan metode pembobotan *word embedding* untuk menguji performa klasifikasi teks. Pengujian yang dilakukan adalah membandingkan performa implementasi LSTM untuk klasifikasi *binary class* dan *multiclass*.

1. Pendahuluan

Demi menjaga dan meningkatkan kualitas serta menyusun strategi dalam pengelolaan obyek wisata di Yogyakarta, diperlukan berbagai upaya termasuk peran Dinas Pariwisata. Salah satu upaya penting yaitu melakukan analisis kondisi objek wisata yang ada untuk mengetahui evaluasi serta menyiapkan solusi demi terwujudnya sektor pariwisata yang maju. Di era digital saat ini, banyak fasilitas digital yang disediakan untuk masyarakat dalam menyampaikan opini yang berkaitan dengan obyek wisata, salah satunya kolom ulasan pada *google maps*. Ulasan-ulasan yang dituliskan secara *online* dinilai memiliki peran lebih, dibuktikan dengan hasil *local consumer review survey* yang menyebutkan bahwa 78% konsumen lebih mempercayai ulasan *online* daripada rekomendasi dari seseorang [1]. Hasil dari opini yang dituliskan berperan penting bagi Dinas Pariwisata untuk mengetahui evaluasi dan menyiapkan strategi melalui tingkat kepuasan pengunjung [2]. Namun kalimat ulasan yang ada pada *google maps* belum cukup membantu untuk mengetahui kekurangan dan kelebihan suatu tempat wisata karena ulasannya mengandung berbagai aspek penilaian [3]. Maka dari itu perlu adanya analisis sentimen ulasan *google maps* pada tempat wisata khususnya pantai sehingga dapat diketahui penilaian sentimen dari masing-masing tempat.

Tantangan yang dihadapi dalam proses analisis sentimen yaitu adanya penilaian positif maupun negatif dalam satu ulasan yang menyebabkan sulit menemukan pola klasifikasinya. Maka dari

itu, untuk meminimalisir kendala tersebut dapat dilakukan dengan menggunakan metode *deep learning*. Hal tersebut karena metode *deep learning* sangat efektif dan lebih mudah dalam mengidentifikasi pola-pola dari data yang diproses. *Deep learning* memiliki algoritma dengan struktur dan jumlah jaringan syaraf yang banyak sebagai jaringan tersembunyi (*hidden layer*) yang dapat digunakan untuk mengidentifikasi pola untuk klasifikasi ulasan. Selain itu, kelebihan lain dari *deep learning* yaitu mampu menangani data dengan skala yang besar dan mengekstraksi fitur dari data tersebut secara otomatis. Salah satu algoritma yang termasuk dalam metode *deep learning* dan mampu mengatasi permasalahan tersebut adalah *Long-Short Term Memory* (LSTM).

LSTM merupakan salah satu variasi dari *Recurrent Neural Network* (RNN) yang dibuat untuk menghindari masalah ketergantungan jangka panjang. Selain itu, LSTM terbukti memiliki tingkat akurasi yang tinggi dalam mengatasi masalah analisis sentimen. Seperti penelitian yang dilakukan oleh Li dan Qian [4] tentang analisis sentimen dengan LSTM yang hasilnya lebih tinggi tingkat akurasinya dibandingkan dengan RNN konvensional. Metode LSTM juga pernah dibandingkan dengan metode *Naive Bayes Classifier* dan menghasilkan akurasi yang lebih tinggi yaitu selisih 4,97% [5].

Berdasarkan permasalahan yang telah dipaparkan di atas, maka penerapan metode *deep learning* dengan menggunakan LSTM menjadi solusi yang ditawarkan pada penelitian ini. Data yang digunakan bersumber dari ulasan *google maps*, dimana data tersebut akan menjadi data latih dan juga data uji.

2. Metode/Perancangan

Metode penelitian yang digunakan adalah metode kualitatif dengan tahapan wawancara bersama pihak Dinas Pariwisata DIY. Sedangkan metode pengembangan sistem menggunakan *prototyping*.

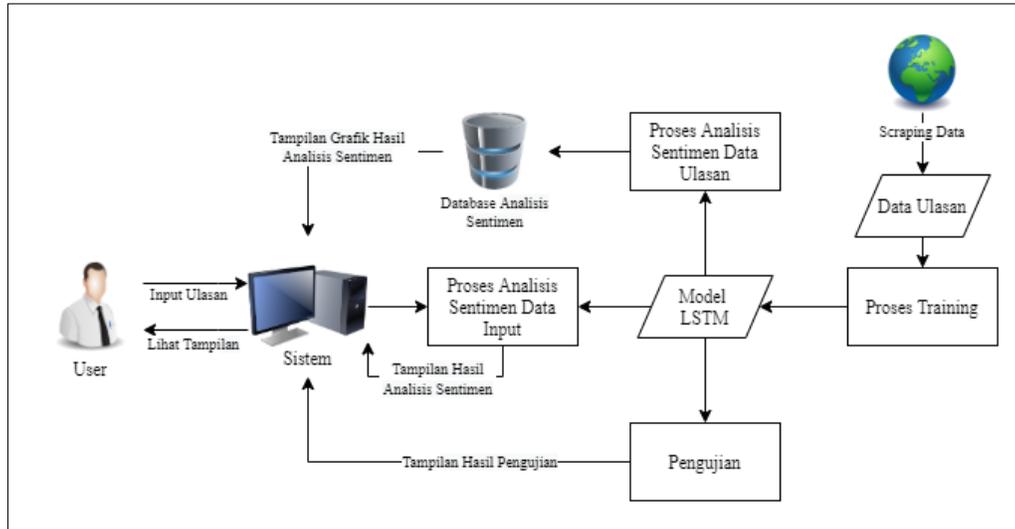
2.1. Pengumpulan Data

Data pada penelitian ini menggunakan data primer yang berupa teks. Data didapatkan dari ulasan 35 pantai di Provinsi Daerah Istimewa Yogyakarta dari *google maps*. Data yang sudah terkumpul, selanjutnya diberi label secara manual baik label sentimen (positif atau negatif) maupun label kategori (daya tarik, aksesibilitas, kebersihan, atau fasilitas). Sehingga diperoleh *dataset* dengan total 3135 ulasan yang terdiri dari 2466 ulasan sentimen positif dan 669 ulasan sentimen negatif. Sedangkan untuk label kategori, 1352 ulasan daya tarik, 526 ulasan aksesibilitas, 536 ulasan kebersihan, dan 721 ulasan fasilitas. Sehingga data yang digunakan pada penelitian ini merupakan data *imbalanced* atau tidak seimbang.

Tabel 1. Contoh Dataset

No	Ulasan	Label Sentimen	Label Kategori
1.	Kurang suka .. Masih sepi bgt.. Cocok buat camp kyknya..."	Negatif	Daya Tarik
2.	Pantainya bersih, indah ombaknya besar sekali	Positif	Kebersihan
3.	Fasilitas minim, kebanyakan toilet, jajanan mahal	Negatif	Fasilitas

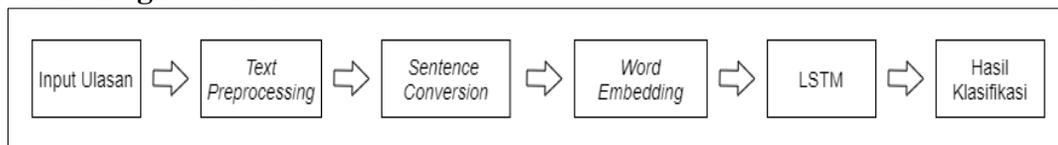
2.2. Arsitektur Sistem



Gambar 1. Arsitektur Sistem

Secara garis besar, *user* akan melakukan *input* ulasan pada halaman klasifikasi. Setelah proses *input*, *user* dapat menekan *button* “prediksi” untuk memberikan perintah pada sistem untuk melakukan proses analisis sentimen terhadap teks ulasan yang telah diinputkan sebelumnya. Sehingga ketika *button* “prediksi” sudah dipilih maka lanjut ke proses teks *preprocessing*, *sentence conversion*, dan proses klasifikasi dengan LSTM. Setelah rangkaian proses tersebut selesai maka akan ditampilkan sebagai *output* kepada *user* berupa hasil klasifikasi sentimen dan kategori. Selain itu, *user* juga dapat melihat data yang telah tersimpan dalam *database*. Data yang ditampilkan berupa grafik prosentase jumlah data ulasan pantai bernilai positif, yang mana sebelumnya merupakan hasil dari klasifikasi ulang seluruh dataset. Terakhir, *user* dapat melihat hasil pengujian berupa akurasi, presisi, dan *recall*.

2.3. Perancangan Proses



Gambar 2. Tahapan Proses

Pada tahap pertama adalah proses *input* ulasan yang dilakukan pada halaman klasifikasi. Kemudian akan dilakukan proses *text preprocessing*, yang terdiri dari tahapan-tahapan dalam pembersihan dan penyeleksian data. Hasil dari *preprocessing* teks akan diubah menjadi numerik agar dapat diproses oleh algoritma atau biasa disebut dengan *sentence conversion*. Selain itu, hasil dari *text preprocessing* juga dilakukan proses pembobotan fitur menggunakan metode *word embedding* dengan *word2vec*. Hasil dari proses *word embedding* tersebut akan digunakan sebagai bobot pada *hidden layer* saat proses *training* [6]. Klasifikasi dilakukan dua kali, yang pertama untuk klasifikasi dalam menentukan nilai sentimen (positif atau negatif) dan yang kedua untuk klasifikasi dalam menentukan nilai kategori (daya tarik, aksesibilitas, kebersihan, atau fasilitas). *Output* dari sistem ini adalah hasil klasifikasi ulasan baik sentimen maupun kategori dari suatu ulasan.

2.4. Text Preprocessing

Preprocessing dilakukan guna membersihkan dan mempersiapkan data sebelum dilakukannya proses klasifikasi [7]. Tahapan yang dilakukan pada *text preprocessing* ini yaitu *case folding* (mengubah huruf besar menjadi huruf kecil), *remove punctuation* (menghapus simbol), *remove number* (menghapus angka), *spelling correction* (memperbaiki kata-kata yang salah ejaan), *negation word conversion* (mengganti kata-kata yang negasi berdasarkan kamus yang telah dibuat sebelumnya), *tokenizing* (memotong kalimat menjadi kata-kata atau token), *stopword removal* (menghapus kata-kata yang tidak dibutuhkan berdasarkan kamus *stopword* yang dibuat), dan *stemming* (mengubah kata yang masih memiliki imbuhan menjadi kata dasar).

Tabel 2. Contoh *Text Preprocessing*

Preprocessing	Teks
Input	Pantainya Indrayanti benar2 indah dan tdk kotor!!
Case Folding	pantainya indrayanti benar2 indah dan tdk kotor!!
Remove Punctuation	pantainya indrayanti benar2 indah dan tdk kotor
Remove Number	pantainya indrayanti benar indah dan tdk kotor
Spelling Correction	pantainya indrayanti benar indah dan tidak kotor
Negation Word Conversion	pantainya indrayanti benar indah dan tidak bersih
Tokenizing	'pantainya', 'indrayanti', 'benar', 'indah', 'dan', 'tidak', 'bersih'
Stopword Removal	'pantainya', 'indrayanti', 'indah', 'bersih'
Stemming	'pantai', 'indrayanti', 'indah', 'bersih'

2.5. Sentence Conversion

Setelah melalui tahapan *text preprocessing*, selanjutnya kata-kata yang dihasilkan akan dikonversi menjadi bentuk angka atau disebut dengan *sentence conversion*. Perubahan ini dilakukan karena algoritma yang digunakan tidak dapat melakukan pemrosesan terhadap teks secara langsung. Terdapat tiga proses dalam bagian ini antara lain: pembentukan *vocabulary*, *text to sequence* (mengubah kata menjadi index angka sesuai *vocabulary* yang dibuat sebelumnya), dan *padding sequence* (menambahkan *padding* angka nol untuk menyamakan panjang data [5]. Pada penelitian ini, *padding* ditambahkan dengan panjang 50 kata.

Tabel 3. Contoh Sentence Conversion

Hasil preprocessing	Text to sequence	Padding sequence
pantai indrayanti indah bersih	[1 8 5 3]	[0 0 0 0 0 0 1 8 5 3]

2.6. Word Embedding

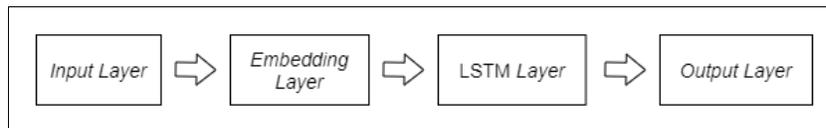
Proses ini akan mengubah suatu kata menjadi vektor. Vektor yang dihasilkan dari proses ini akan merepresentasikan dari setiap kata yang ada. Kata yang menjadi *input* pada *word embedding* berasal dari hasil akhir keseluruhan *preprocessing*. Vektor yang dihasilkan memiliki panjang atau *dimensi* tertentu yang dapat ditentukan, semakin panjang suatu vektor maka akan semakin teliti representasi yang dihasilkan [8]. Pada penelitian ini metode yang digunakan pada *word embedding* adalah *word2vec*. Hasil dari *word2vec* akan dijadikan sebagai parameter *weight* untuk *embedding layer*. Pada penelitian ini, panjang dimensi yang digunakan sebesar 100. Karena panjang maksimum ulasan dibuat 50 kata, maka dimensi yang dihasilkan adalah $n*m$ yaitu $50 \times 100 = 5000$ dimensi.

Tabel 4. Contoh word embedding dimensi 5

Kata	Dimensi				
	1	2	3	4	5
pantai	0.702900668	-0.13690501	0.651611363	0.725284718	0.580436679
indrayanti	0.8127969	2.364095	-0.6084724	0.6790205	-0.65735227
indah	2.7809677	3.9450598	-1.4909214	0.26488346	-0.89886343
bersih	2.928585	2.9040217	-1.6556750	-0.00391697	-2.49438300

2.7. Long-Short Term Memory (LSTM)

LSTM merupakan variasi dari algoritma *Reccurent Neural Network* (RNN) yang memperbaiki kekurangan RNN yaitu ketergantungan memori jangka panjang. LSTM memiliki lapisan tersembunyi untuk menyimpan dan memperbarui informasi sebelumnya. Lapisan tersembunyi terdiri dari 3 gerbang atau *gate*, yaitu *forget gate*, *input gate* dan *output gate* [9]. Dalam penelitian ini dibuat sebuah arsitektur model jaringan atau *network* dengan menggunakan *word embedding* dan LSTM layer untuk mengklasifikasikan sentimen dan kategori. Layer yang disusun untuk pemodelan ditunjukkan pada **Gambar 3**.



Gambar 3. Network Layer

1. Input Layer

Layer pertama merupakan *input* atau masukan yang terdiri dari *sequence* ulasan dengan ukuran tertentu. Pada penelitian ini, input yang digunakan berukuran 50. Artinya panjang maksimum 50 adalah *sequence* dari data ulasan yang telah melalui *preprocessing* dan *sentence conversion*.

2. Embedding Layer

Layer ini akan mengubah indeks bilangan bulat positif pada *input* menjadi vektor. Ukuran dimensi ditentukan berdasarkan dimensi vektor dari kamus kata *word2vec* yang telah dibuat sebelumnya. *Input* berubah menjadi vektor berdimensi $n \times m$ dimana n adalah panjang kalimat maksimum dan m adalah dimensi vektor. Sehingga *input* pada *embedding layer* adalah $50 \times 100 = 5000$ dimensi.

3. LSTM Layer

Langkah pertama LSTM adalah memutuskan informasi yang dihapus dari *cell state*. Keputusan ini dibuat oleh *sigmoid layer* yang disebut *forget gate*. Output 1 berarti “biarkan lewat”, dan 0 berarti “lupakan informasi” [10]. Perhitungan nilai *forget gate* dilihat pada **Persamaan 1**.

$$f_t = \sigma(x_t W_f + h_{t-1} U_f + b_f) \quad (1)$$

Langkah selanjutnya adalah menentukan informasi baru yang akan disimpan pada *cell state*. Terdapat dua proses, pertama menghitung *input gate* yang menentukan bagian mana yang

akan diperbarui. Kedua yaitu menghitung nilai kandidat konteks (kandidat baru) yang ditambahkan ke *cell state*. Berikutnya, kedua hasil tersebut digabungkan untuk memperbarui nilai *cell state*. Rumus menghitung *input gate* dilihat pada **Persamaan 2** sedangkan rumus menghitung kandidat konteks dilihat pada **Persamaan 3**.

$$i_t = \sigma(x_t W_i + h_{t-1} U_i + b_i) \quad (2)$$

$$\check{C}_t = \tanh(x_t W_c + h_{t-1} U_c + b_c) \quad (3)$$

Selanjutnya, memperbarui *cell state* lama menjadi baru dengan cara mengalikan *cell state* lama dengan *forget gate* kemudian ditambahkan dengan perkalian *input gate* dengan kandidat konteks. Proses menghitung nilai *cell state* yang baru terdapat pada **Persamaan 4**.

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

Terakhir adalah memutuskan hasil *output*. Pertama, lapisan *sigmoid* memutuskan bagian sel yang menjadi *output*. Selanjutnya, *output* dari gerbang *sigmoid* (O_t) akan dikalikan dengan nilai baru yang dihasilkan oleh lapisan *tanh* dari *cell state* (C_t), nilai yang dihasilkan berada antara -1 sampai 1. Proses menentukan hasil *output* dijelaskan pada **Persamaan 5** dan **Persamaan 6**.

$$O_t = \sigma(x_t W_o + h_{t-1} U_o + b_o) \quad (5)$$

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

4. Output Layer

Output layer atau disebut *dense layer* yaitu memproses data hasil dari perhitungan LSTM *layer* untuk memperoleh nilai probabilitas dari setiap kelas [11]. Pada model klasifikasi sentimen terdiri dari dua kelas (positif dan negatif), sedangkan model klasifikasi kategori terdiri dari empat kelas (daya tarik, aksesibilitas, kebersihan, dan fasilitas). Output dari *layer* LSTM pada periode terakhir akan digunakan sebagai input yang kemudian akan dihitung dengan bobot serta *bias* pada *dense layer*. Hasil pada perhitungan tersebut kemudian akan menjadi input pada fungsi *sigmoid* sesuai pada **persamaan 7**.

$$Y = \sigma(h_t W_y + b_y) \quad (7)$$

3. Hasil dan Pembahasan

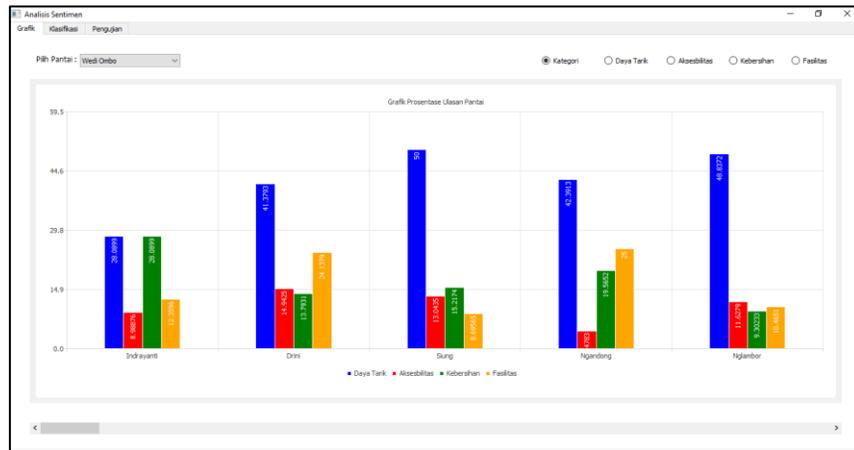
3.1. Hasil Penelitian

Hasil penelitian ini mencakup implementasi sistem berupa halaman grafik persentase ulasan positif, halaman klasifikasi, dan halaman pengujian akurasi, presisi, *recall*. Di dalam halaman klasifikasi terjadi proses *text preprocessing*, *sentence conversion*, *word embedding*, dan klasifikasi LSTM. Implementasi sistem dibuat dalam bentuk aplikasi dekstop menggunakan modul *pyqt5* dengan bahasa *python* yang penyimpanannya menggunakan *MySQL*.

3.2. User Interface

Halaman grafik merupakan halaman awal ketika aplikasi dijalankan. Pada halaman ini *user* dapat melihat grafik ulasan pantai untuk masing-masing kategori secara keseluruhan dari 35

pantai yang ada. Grafik menampilkan persentase jumlah ulasan positif dari data pantai. Data persentase diperoleh dari proses klasifikasi ulang datasets berdasarkan model *training*.



Gambar 4. Halaman Grafik

Halaman klasifikasi digunakan *user* untuk memasukkan data uji berupa ulasan. Data uji baru itulah yang nanti akan diprediksi oleh sistem untuk diketahui sentimen dan kategorinya. Ketika *user* menekan tombol prediksi maka data uji tersebut akan diproses dalam sistem sehingga menghasilkan keluaran berupa hasil klasifikasi sentimen dan kategori. Hasil dari *preprocessing text* sebagai langkah awal dalam melakukan klasifikasi juga akan ditampilkan pada halaman klasifikasi.

KLASIFIKASI ULASAN GOOGLE MAPS

Masukkan Ulasan :
 Mulai kotor daerah pantai namun untuk masakannya tetap wuemakkj

Hasil Preprocessing :
 mulai kotor daerah pantai masakan enak

Sentimen :
 NEGATIF

Kategori :
 KEBERSIHAN

Buttons: Prediksi, Detail, Hapus

Gambar 5. Halaman Klasifikasi

3.3. Pengujian Confusion Matrix

Pengujian *confusion Matrix* dilakukan dengan menggunakan pendekatan *K-Fold Cross Validation* dengan $K=5$. Menggunakan $K=5$ karena pada penelitian ini menggunakan perbandingan data latih : data uji sebesar 80:20.

1. Pengujian Model Klasifikasi Sentimen

Waktu yang dibutuhkan dalam pengujian ini setiap iterasinya rata-rata sebesar 277 detik. Berikut hasilnya disajikan pada Tabel 5.

Tabel 5. Pengujian model klasifikasi sentimen

Fold	Akurasi	Presisi	Recall
1.	84%	76%	75%
2.	85%	77%	76%
3.	81%	71%	71%
4.	83%	77%	69%
5.	85%	78%	75%
Rata-rata	84%	76%	73%

2. Pengujian Model Klasifikasi Kategori

Waktu yang dibutuhkan dalam pengujian ini setiap iterasinya rata-rata sebesar 279 detik. Berikut hasilnya disajikan pada **Tabel 6**.

Tabel 6. Pengujian model klasifikasi kategori

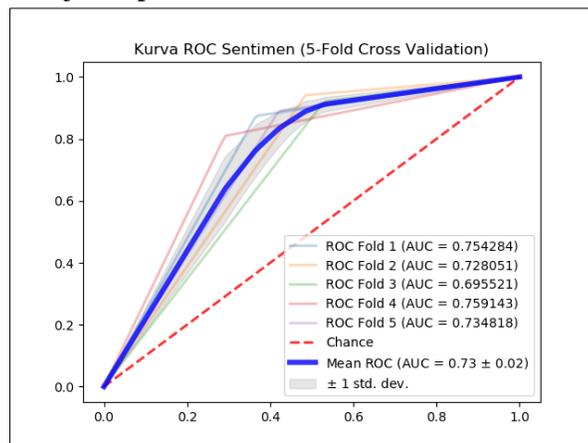
Fold	Akurasi	Presisi	Recall
1.	75%	76%	74%
2.	77%	77%	76%
3.	76%	76%	73%
4.	76%	74%	74%
5.	75%	74%	73%
Rata-rata	76%	75%	74%

3.1 Pengujian Kurva ROC

Pengujian dengan Kurva ROC juga menggunakan pendekatan *5-Fold Cross Validation*.

1. Pengujian Model Klasifikasi Sentimen

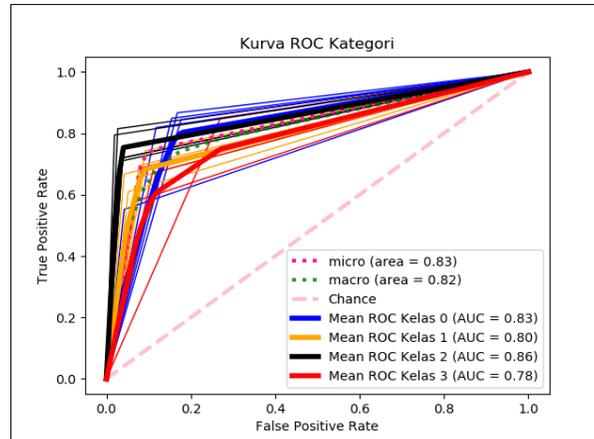
Waktu yang dibutuhkan dalam pengujian ini setiap iterasinya rata-rata sebesar 277 detik. Nilai AUC yang diperoleh untuk model sentimen rata-rata sebesar 0.73 (*fair classification*) [12]. Berikut plot kurva ROC disajikan pada **Gambar 6**.



Gambar 6. Kurva ROC model klasifikasi sentimen

2. Pengujian Model Klasifikasi Kategori

Waktu yang dibutuhkan dalam pengujian ini setiap iterasinya rata-rata sebesar 279 detik. Nilai AUC yang diperoleh untuk model kategori rata-rata sebesar 0.83 (*good classification*). Berikut plot kurva ROC disajikan pada **Gambar 7**.



Gambar 7. Kurva ROC model klasifikasi kategori

4. Kesimpulan dan Saran

Berdasarkan dari hasil analisis, perancangan, dan pembahasan yang telah dilakukan, maka diperoleh kesimpulan bahwa hasil pengujian menggunakan *confusion matrix* dengan pendekatan *5-fold cross validation* menghasilkan rata-rata nilai akurasi untuk model klasifikasi sentimen sebesar 84%, presisi sebesar 76% dan *recall* sebesar 73% dalam waktu komputasi rata-rata selama 277 detik. Sedangkan untuk klasifikasi kategori diperoleh rata-rata nilai akurasi sebesar 76%, presisi sebesar 75% dan *recall* sebesar 74% dalam waktu komputasi rata-rata selama 279 detik.

Hasil pengujian menggunakan kurva ROC dengan pendekatan *5-fold cross validation* pada model klasifikasi sentimen dihasilkan nilai rata-rata AUC sebesar 0.73, sedangkan pada model klasifikasi kategori sebesar 0.83. Hasil AUC model klasifikasi sentimen berada pada nilai antara 0.70 – 0.80 sehingga masuk dalam kategori *fair classification*. Hasil AUC model klasifikasi kategori berada pada nilai antara 0.80 – 0.90 sehingga masuk dalam kategori *good classification* [13].

Saran yang diberikan pada penelitian ini yaitu menambahkan jumlah data latih dengan data yang seimbang agar dapat mengklasifikasikan teks lebih baik lagi. Selain itu dapat melakukan *hyperparameter tuning* atau pengaturan parameter dalam pembentukan model klasifikasi agar mendapatkan model yang lebih baik.

Daftar Pustaka

- [1] Brightlocal.com, “Local Consumer Review Survey 2020,” <https://www.brightlocal.com/research/local-consumer-review-survey/>, pp. 1–46, 2020.
- [2] M. T. Dewi, A. Herdiani, and D. S. Kusumo, “Multi-Aspect Sentiment Analysis Komentar Wisata TripAdvisor dengan Rule-Based Classifier (Studi Kasus : Bandung Raya),” *e-Proceeding Eng.*, vol. 5, no. 1, pp. 1589–1596, 2018.
- [3] D. Koesumaningrum, “Analisis Sentimen Ulasan TripAdvisor pada Tempat Wisata Menggunakan Ontology Supported Polarity Mining (OSPM)(Studi Kasus Bandung),” 2018.
- [4] D. Li and J. Qian, “Text sentiment analysis based on long short-term memory,” *2016 1st IEEE Int. Conf. Comput. Commun. Internet, ICCCI 2016*, pp. 471–475, 2016.
- [5] M. A. Nurrohmah and A. SN, “Sentiment Analysis of Novel Review Using Long Short-Term Memory Method,” *IJCCS (Indonesian J. Comput. Cybern. Syst.*, vol. 13, no. 3, p. 209, 2019.
- [6] W. Xing and D. Du, “Dropout Prediction in MOOCs : Using Deep Learning for Personalized Intervention,” *J. Educ. Comput. Res.*, no. March, 2018.
- [7] E. Haddi, X. Liu, and Y. Shi, “The Role of Text Pre-processing in Sentiment Analysis,” *Procedia Comput. Sci.*, no. 17, pp. 26–32, 2013.
- [8] A. H. Abdullah, “Word Embedding,” <https://rpubs.com/>, 2018. .
- [9] C. Olah, “Understanding LSTM Networks,” 2015.
- [10] M. Ryan, “PENGENALAN LSTM (LONG SHORT TERM MEMORY),” 2017.
- [11] Z. Wang and B. Song, “Research on hot news classification algorithm based on deep learning,” *2019 IEEE 3rd Inf. Technol. Networking, Electron. Autom. Control Conf.*, no. Itnc, pp. 2376–2380, 2019.
- [12] A. P. Chazhooor, “ROC Curve in Machine Learning,” 2019. [Online]. Available: <https://towardsdatascience.com/roc-curve-in-machine-learning-fea29b14d133>. [Accessed: 10-May-2020].
- [13] Suwarno and A. A. Abdillah, “Penerapan Algoritma Bayesian Regularization Backpropagation untuk Memprediksi Penyakit Diabetes,” 2016.