Analisis Sentimen Pengguna TikTok Terhadap Progres Pembangunan IKN Menggunakan LSTM dan FastText

I Gusti Agus Sakah Aditia[a1], Luh Arida Ayu Rahning Putri[a2],

[a]Program Studi Informatika, Fakultas Matematika dan Ilmu Pengetahuan Alam
Universitas Udayana, Bali
Jln. Raya Kampus UNUD, Bukit Jimbaran, Kuta Selatan, Badung, 80361, Bali, Indonesia
[1]aditia.2308561012@student.unud.ac.id
[2]rahningputri@unud.ac.id

Abstract

The development of the Capital City of the Archipelago (IKN) is one of the national strategic projects that has generated various reactions and public opinions. Social media, especially TikTok, has become an important platform for people to voice their views through video content and interactive comments. This research aims to analyze the sentiment of TikTok users towards the progress of IKN construction using a deep learning approach, namely the Long Short-Term Memory (LSTM) algorithm. Sentiment data was collected from Kaggle as many as 1472 Indonesian comments, then processed through the stages of normalization, tokenization, and conversion into word embeddings. The LSTM model was designed and trained to classify sentiment into positive, negative, and neutral. The results of the analysis are expected to provide a comprehensive picture of public perception towards IKN, identify critical issues that are often discussed, and measure the level of public acceptance or rejection of this project. This research is expected to contribute to a better understanding of the dynamics of public opinion in the digital era.

Keywords: *IKN, Capital City of the Archipelago, TikTok, Sentiment Analysis, Deep Learning, LSTM.*

1. Pendahuluan

Pembangunan Ibu Kota Nusantara (IKN) merupakan inisiatif strategis pemerintah Indonesia yang didasarkan pada Undang-Undang Nomor 3 Tahun 2022 tentang Ibu Kota Negara. Proyek ini bertujuan untuk mewujudkan pemerataan ekonomi dan pembangunan serta mewujudkan tata kelola pemerintahan yang baik [1]. Beberapa negara yang pernah melakukan pemindahan ibu Kota adalah Korea Selatan yang memindahkan pusat ibu kota dari Seoul ke Senjong, Malaysia dari Kuala Lumpur ke Putrajaya, Amerika Serikat yang memindahkan pusat pemerintahan ke Washington dengan tetap menjadikan New York sebagai pusat perekonomian dan bisnis, Brazil yang memindahkan ibu kota dari Rio de Janeiro ke Brasilia serta Turki yang memindahkan ibu kotanya dari Istanbul ke Ankara. [2]. Negara - negara tersebut juga telah melakukan pemindahan ibu kota dengan tujuan serupa, menunjukkan bahwa ini adalah strategi pembangunan yang telah diterapkan secara global. Keberhasilan pembangunan IKN di Indonesia sangat bergantung pada dukungan dan persepsi positif dari seluruh elemen masyarakat.

Di era digital, platform media sosial seperti TikTok telah menjadi wadah utama bagi masyarakat untuk berbagi opini terkait isu-isu nasional, termasuk progres pembangunan IKN. Topik pemindahan ibu kota ini telah menjadi perbincangan intens dan menyita perhatian luas, memicu munculnya berbagai pandangan. Sebuah penelitian sebelumnya yang menganalisis sentimen pengguna TikTok terhadap perpindahan ibu kota menunjukkan bahwa polaritas negatif masih mendominasi komentar, yaitu sebesar 42,4%, diikuti sentimen positif 30,9% dan netral 26,7%. Hasil ini mengindikasikan adanya keraguan masyarakat terhadap keberlangsungan dan dampak proyek tersebut pada periode pengumpulan data [3].

Adanya dominasi sentimen negatif pada periode awal perbincangan IKN ini menunjukkan adanya kesenjangan antara harapan pemerintah terhadap dukungan publik dan realitas persepsi masyarakat. Seiring berjalannya waktu, progres pembangunan fisik IKN terus berlangsung, informasi baru disebarkan, dan sosialisasi semakin gencar dilakukan. Perubahan-perubahan ini berpotensi memengaruhi opini dan sentimen publik. Oleh karena itu, penting untuk mengetahui apakah sentimen yang mayoritas negatif pada periode awal tersebut telah bergeser menjadi lebih positif atau tetap konsisten, setelah masyarakat melihat progres nyata pembangunan IKN.

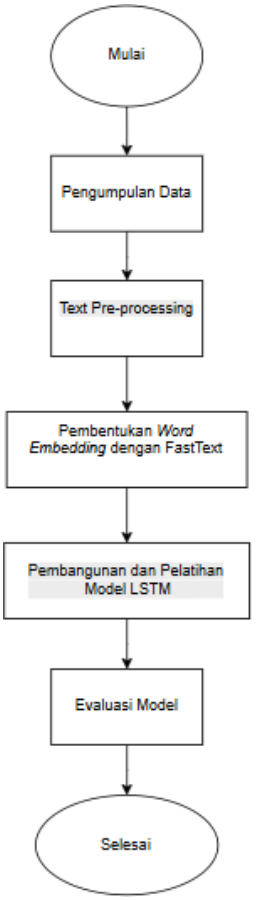
Analisis sentimen menggunakan pendekatan Deep Learning, khususnya *Long Short-Term Memory* (LSTM), sangat relevan untuk tugas ini karena kemampuannya memproses dan memahami konteks dalam data teks sekuensial. LSTM merupakan salah satu algoritma *Deep Learning* yang dapat dimanfaatkan untuk analisis sentiment LSTM memiliki kemampuan untuk menyimpan atau mengingat informasi pada sel memori dengan ruang yang sangat besar. Metode LSTM juga dirancang untuk menyelesaikan permasalahan *vanishing gradient* yang terjadi pada RNN [4]. Untuk merepresentasikan kata-kata sebagai vektor numerik yang padat dan bermakna, metode FastText akan digunakan. FastText menjadi pilihan unggul karena mempertimbangkan informasi subword dalam proses pembelajaran. Tidak seperti model lainnya yang hanya merepresentasikan kata sebagai satu entitas, FastText memecah kata menjadi n-gram sehingga dapat mengenali hubungan antar kata berdasarkan komponen penyusunnya. Keunggulan ini memungkinkan FastText menangani masalah kata *out-of-vocabulary* (OOV), yaitu kata-kata yang tidak ditemui selama proses pelatihan, yang sering muncul pada ulasan yang berbahasa informal atau tidak terstruktur [5]. Dengan menggunakan FastText, model deep learning dapat memanfaatkan informasi lebih kaya dari data teks, meningkatkan akurasi dalam analisis sentimen.

Penelitian lain telah membuktikan potensi tinggi metode LSTM dan FastText dalam memahami opini publik terkait kurikulum merdeka, dengan tingkat akurasi sebesar 89,5%, presisi sebesar 89,7%, recall sebesar 89,5%, dan skor-f1 sebesar 89,6%. [6]. Penelitian lain juga menunjukkan efektivitas kombinasi LSTM dengan FastText, penelitian itu menghasilkan accuracy sebesar 0,859, precision sebesar 0,847, *recall* sebesar 0,875, dan f1-score sebesar 0,863. Hasil metode ini lebih unggul dibandingkan tanpa menggunakan FastText dengan selisih 0,053 pada nilai *f1- score*. Hal ini dikarenakan FastText dapat merepresentasikan kata-kata yang sebelumnya tidak pernah dijumpai saat pelatihan [7]. Selain itu, penelitian lain juga menganalisis sentimen kendaraan listrik di Twitter menggunakan LSTM yang memberikan akurasi 89,7% [8]. Hasil dari penelitian-penelitian tersebut semakin memperkuat argumen bahwa kombinasi LSTM dan FastText adalah metode yang tepat untuk melakukan analisis sentimen pada data teks media sosial.

2. Metode Penelitian

Pada tahap metode penelitian, penulis akan menyusun gambaran alur tahapan penelitian. Pada tahapan ini input yang digunakan penulis merupakan data reguler yang dapatkan dari platform Kaggle dengan jumlah 1472 komentar dari platform Tiktok mengenai Progres Pembangunan IKN. Data tersebut telah dilabeli dengan 2 kelas label yaitu kelas positif dan negative untuk mengetahui pandangan masyarakat terhadap progress perkembangan IKN [9]. Output yang akan diperoleh yakni perbandingan kinerja antara model *word embedding* LSTM tanpa FastText dan model yang menggunakan *word embedding* FastText, kemudian diuji melalui 10 percobaan untuk memastikan validitas hasil.

p-ISSN: 2986-3929
e-ISSN: 3032-1948



Gambar 1. Bagan Alur Penelitian

Pada gambar 1, Menjelaskan alur penelitian ini, yang pertama dilakukan adalah pengumpulan data, prapemrosesan data, pembentukan word embedding menggunakan FastText, pembangunan dan pelatihan model LSTM, evaluasi model, hingga analisis sentimen.

2.1 Pengumpulan Data

Teknik pengumpulan data pada penelitian kali ini menggunakan data regular yang diambil melalui platform Kaggle, dengan kata kunci yang digunakan saat pencarian adalah “analisis sentimen”. Dataset yang digunakan adalah dataset komentar dari platform Tiktok mengenai progress Pembangunan IKN. Dataset ini terdiri dari 1472 komentar pengguna TikTok yang relevan dengan topik IKN. Komentar-komentar tersebut telah dilabeli dengan 2 kelas sentimen utama, yaitu positif dan negatif, yang akan digunakan untuk menganalisis pandangan masyarakat terhadap perkembangan IKN.

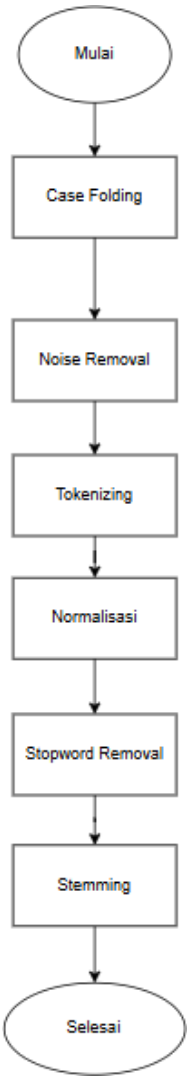
Tabel 1. Contoh Dataset

Text	Label
ikn ibukota koruptor nepotisme	negatif
serta seluruh pekerja di ikn yg gak pernah berhenti bergantian menuntaskan mimpi besar bangsa ini	positif
jadi bendera singapore ndak	negatif
tapera gw bakal lari kesini	negatif
ketika indonesia berada di tangan yang tepat	positif

p-ISSN: 2986-3929
e-ISSN: 3032-1948

Text	Label
sehat sehat para pekerja di sana kalian menjadi sejarah bangsa ini	positif

2.2 Text Pre-processing



Gambar 2. Bagan Alur tahap Text Pre-processing

Tahap pertama pada Text Pre-processing adalah case folding yang digunakan untuk mengubah semua data menjadi huruf kecil. Setelah itu dilakukan noise removal yaitu menghapus karakter yang tidak relevan seperti tautan URL, angka, tanda baca yang berlebihan, emotikon, atau karakter khusus lain yang tidak memberikan informasi sentiment. Dilanjutkan dengan *tokenizing* membagi teks menjadi unit-unit kata atau token. Kemudian dilakukan yaitu tahap normalisasi untuk menyeragamkan tulisan, termasuk mengubah slang menjadi bentuk standar, misalnya “yg” menjadi “yang”, atau “baikk” menjadi “baik”. Lalu dilakukan stopword removal yang akan menghapus kata-kata umum yang tidak memiliki makna yang signifikan dalam analisis sentiment

seperti kata yang, dan, di, ke. Lalu yang selanjutnya yaitu dilakukan *stemming* atau *lemmatization* yang akan mengubah kata menjadi bentuk dasarnya seperti “membangun” menjadi “bangun”.

2.3 Pembentukan Word Embeding dengan FastText

Word embedding adalah teknik esensial dalam metode deep learning seperti LSTM. Ini bertujuan untuk merepresentasikan kata-kata sebagai vektor angka dalam ruang multidimensi, di mana kata-kata dengan makna serupa akan memiliki representasi vektor yang dekat satu sama lain. Dalam penelitian ini, FastText akan digunakan. FastText, yang diperkenalkan oleh Facebook pada tahun 2017, bertujuan untuk mengubah kata-kata dari data teks menjadi representasi numerik yang padat dan bermakna yang dapat diproses oleh jaringan saraf. Keunggulan FastText terletak pada kemampuannya untuk menangkap informasi sub-kata (karakter N-gram), sehingga efektif dalam menangani kata-kata yang jarang muncul (OOV - out-of-vocabulary) dan bahasa dengan morfologi kompleks seperti bahasa Indonesia [6].

Word embedding FastText akan diimplementasikan sebagai lapisan embedding awal pada model LSTM. *Word embedding* ini dapat dilatih dari awal menggunakan data TikTok yang telah diproses, atau menggunakan model pre-trained FastText yang telah dilatih pada korpus bahasa Indonesia yang besar dan kemudian disesuaikan (fine-tuned) dengan data penelitian.

2.4 Long Short-Term Memory (LSTM)

Deep learning dalam Natural Language Processing telah terbukti efektif dalam menyelesaikan berbagai permasalahan klasifikasi teks. Dalam penelitian ini, metode *Long Short-Term Memory* (LSTM) yang diperkenalkan oleh Hochreiter dan Schmidhuber (1997) digunakan untuk melakukan klasifikasi sentimen pada komentar TikTok terkait Ibu Kota Negara (IKN). Metode *Long Short-Term Memory* (LSTM) digunakan karena memiliki beberapa keunggulan, antara lain LSTM mendapatkan akurasi yang baik saat pemrosesan data teks dan LSTM mampu memproses data yang relatif panjang [7].

Secara internal, LSTM terdiri dari enam komponen utama, yaitu *forget gate, input gate, candidate gate, cell state, hidden state, dan output gate.* Tahapan-tahapan proses internal LSTM dijelaskan sebagai berikut:

a. *Forget Gate*: Menentukan informasi apa yang perlu dilupakan dari *cell state* sebelumnya. Perhitungannya menggunakan fungsi sigmoid seperti pada Persamaan (1):

$$f_t = \sigma\,(Wf \cdot [h_t{-}1, x_t] + bf) \quad (1)$$

Keterangan:

f_t = *forget gate*
σ = fungsi sigmoid
Wf = bobot *forget gate*
$h_t{-}1$ = *hidden state* saat t-1
x_t = *input*
bf = bias *forget gate*

b. *Input Gate*: Menentukan informasi baru yang akan disimpan di *cell state*. Dihitung menggunakan Persamaan (2):

$$it = \sigma\,(Wi \cdot [ht{-}1, xt] + bi) \quad (2)$$

Keterangan:

it = *input gate*
Wi = bobot *input gate*

bi = bias *input gate*

c. Candidate Gate: Membentuk vektor kandidat untuk memperbarui *cell state* dengan fungsi aktivasi tanh, sebagaimana terlihat pada Persamaan (3):

$$\tilde{c}_t = \tanh(Wc \cdot [h_t{-}1, x_t] + bc) \quad (3)$$

Keterangan:

$\tilde{c}_t$ = kandidat *cell state* saat waktu ke-t
tanh = fungsi tanh
Wc = bobot kandidat *cell state*
bc = bias kandidat *cell state*

d. Cell State: Menggabungkan hasil dari *forget gate* dan *input gate* untuk menghasilkan *cell state* baru, seperti pada Persamaan (4):

$$c_t = f_t * c_t{-}1 + i_t * \tilde{c}_t \quad (4)$$

Keterangan:

c_t = *cell state*
$c_t{-}1$ = *cell state* pada saat t-1

e. *Output Gate* dan *Hidden State*: Menentukan output dari unit LSTM saat ini dengan menghitung *output gate* dan *hidden state* menggunakan Persamaan (5) dan (6):

$$o_t = \sigma(Wo \cdot [h_t{-}1, x_t] + bo) \quad (5)$$

$$h_t = ot * \tanh(c_t) \quad (6)$$

Keterangan:

o_t = *output gate*
Wo = bobot *output gate*
bo = bias *output gate*
h_t = *hidden state*

2.5 Confusion Matrix

Confusion matrix digunakan sebagai teknik evaluasi pada penelitian ini. *Confusion matrix* merupakan metode evaluasi yang terdiri dari hasil prediksi oleh sistem dengan hasil yang aktual atau sebenarnya [7]. Pada confusion matrix akan dilakukan 4 buah perhitungan, yaitu perhitungan akurasi, presisi, *recall*, dan juga skor-f1. Berikut merupakan rumus untuk melakukan perhitungan confusion matrix.

$$Accuracy = \frac{TP+TN}{TP+TN+FP+FN} \quad (7)$$

$$Precision = \frac{TP}{TP+FP} \quad (8)$$

$$Recall = \frac{TP}{TP+FN} \quad (9)$$

$$f1 - Score = \frac{2.Precision.Recall}{Precision+Recall} \quad (10)$$

3. Hasil dan Diskusi

3.1 Pre-processing

Pada gambar 3, tahap text preprocessing yang diterapkan pada 1472 data komentar dilakukan menggunakan Bahasa pemrograman Python. Proses *stowords* dilakukan menggunakan *library* NLTK dan juga menggunakan tambahan *stopword* menggunakan file stopwords.txt yang diperoleh dari GitHub untuk meningkatkan akurasi model dengan membuang kata-kata yang tidak berkontribusi terhadap makna. Dataset akan dinormalisasi menggunakan file slangwords.txt yang diperoleh dari Kaggle [10] untuk membantu menyamakan kata tidak baku ke bentuk standar agar model lebih mudah mengenali pola dan makna. Proses *stemming* dalam penelitian ini menggunakan library Sastrawi. Berikut contoh data yang telah melalui tahap pre-processing.

	Sebelum	Sesudah
0	investor	['investor']
1	ini yg bikin smua di pajakin	['bikin', 'pajakin']
2	wkwkwk pegang itu kata semuanya selesai sebelu...	['wkwkwk', 'pegang', 'selesai', 'agustus']
3	lah anis di kasih anggaran 271t	['anis', 'kasih', 'anggar', 't']
4	semoga dapat undangan untuk mengikuti upacara ...	['moga', 'undang', 'ikut', 'upacara', 'bendera...
5	insya allah mogaterwujud dapat undangan agustus	['insya', 'allah', 'mogaterwujud', 'undang', '...
6	benerƎ dikebut tmn ku ada yg kerja disana jam ...	['bener', 'kebut', 'tmn', 'ku', 'kerja', 'sana...
7	bang mau nanya dahnanti monas di jakarta di ba...	['bang', 'nanya', 'dahnanti', 'monas', 'jakart...
8	ibu kota kini jauh d kaltim kami yg d plosok d...	['kota', 'kaltim', 'pelosok', 'daerh', 'jabar'...
9	bangganya yg lolos paskibraka nasional	['bangga', 'lolos', 'paskibraka', 'nasional']

Gambar 3. Contoh data yang telah melalui tahap pre-processing.

3.2 Pembentukan Word Embeding dengan FastText

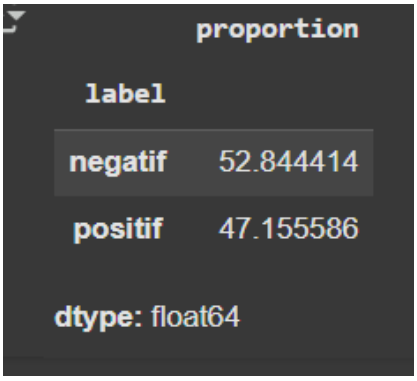
```
1. Kata: ikn
Vektor: [-0.00863107 -0.03941524 -0.04574469 -0.11421788  0.0466078   0.04200456
  0.02675633  0.08746155 -0.          0.04257996]
--------------------------------------------------
2. Kata: indonesia
Vektor: [ 0.09537704 -0.02384426  0.067631    0.12225603 -0.02882988 -0.20050856
  0.02687899  0.06004418 -0.00997124 -0.0344658 ]
--------------------------------------------------
3. Kata: iya
Vektor: [ 0.06363782 -0.06804946  0.08988705  0.13951793  0.00623143 -0.03700259
 -0.07588011 -0.04555012 -0.02795873 -0.06782888]
--------------------------------------------------
4. Kata: jokowi
Vektor: [ 0.0313789  -0.07124978 -0.01077031  0.04774149  0.0092169  -0.07031773
 -0.08678389 -0.01273797 -0.00486735 -0.04691301]
--------------------------------------------------
5. Kata: nya
Vektor: [ 0.08997708 -0.06187304 -0.04568746  0.1455966  -0.09534775 -0.07055439
  0.01934912  0.0118449  -0.00286926 -0.12131824]
--------------------------------------------------
```

Gambar 4. Contoh Hasil Pembentukan Word Embeding dengan FastText

p-ISSN: 2986-3929
e-ISSN: 3032-1948

Pada gambar 4, tahap ekstraksi fitur, digunakan metode FastText untuk mengubah teks menjadi representasi numerik (vektor). Model FastText yang digunakan merupakan pretrained Bahasa Indonesia dari Facebook, dengan dimensi vektor sebesar 300. Proses embedding dijalankan menggunakan Python di Google Colab, dan diterapkan pada 1.472 data komentar hasil preprocessing. Dari proses tokenisasi, diperoleh total kosakata (vocabulary) sebanyak 5.000 kata terbanyak, yang kemudian digunakan untuk membentuk embedding matrix.

3.3 Pemodelan LSTM



Gambar 5. Proporsi Sentimen dalam Dataset

Berdasarkan gambar 5, sebanyak 52,84% data termasuk kategori sentimen negatif, dan 47,16% termasuk sentimen positif, sehingga data dapat dikatakan relatif seimbang.

Arsitektur Model LSTM tanpa FastText:
Model: "sequential_3"

Layer (type)	Output Shape	Param #
embedding_3 (Embedding)	(None, 100, 300)	1,500,000
lstm_3 (LSTM)	(None, 64)	93,440
dropout_6 (Dropout)	(None, 64)	0
dense_6 (Dense)	(None, 64)	4,160
dropout_7 (Dropout)	(None, 64)	0
dense_7 (Dense)	(None, 1)	65

Total params: 4,792,997 (18.28 MB)
Trainable params: 1,597,665 (6.09 MB)
Non-trainable params: 0 (0.00 B)
Optimizer params: 3,195,332 (12.19 MB)

Gambar 6. Arsitektur Model LSTM tanpa FastText

Model LSTM tanpa FastText pada gambar 6, terdiri dari enam lapisan, dimulai dengan Embedding layer berdimensi 300 yang menghasilkan 1.500.000 parameter. Dilanjutkan dengan satu lapisan LSTM satu arah berukuran 64 unit dengan 93.440 parameter. Dua lapisan Dropout digunakan untuk mencegah overfitting, diselingi dengan Dense layer berisi 64 unit, serta Dense output layer berisi 1 neuron untuk klasifikasi biner. Total parameter dalam model ini adalah 4.792.997, dengan 1.597.665 parameter yang dapat dilatih

p-ISSN: 2986-3929
e-ISSN: 3032-1948

Arsitektur Model LSTM dengan FastText:
Model: "sequential_5"

Layer (type)	Output Shape	Param #
embedding_5 (Embedding)	?	672,900
lstm_5 (LSTM)	?	0 (unbuilt)
dropout_10 (Dropout)	?	0
dense_10 (Dense)	?	0 (unbuilt)
dropout_11 (Dropout)	?	0
dense_11 (Dense)	?	0 (unbuilt)

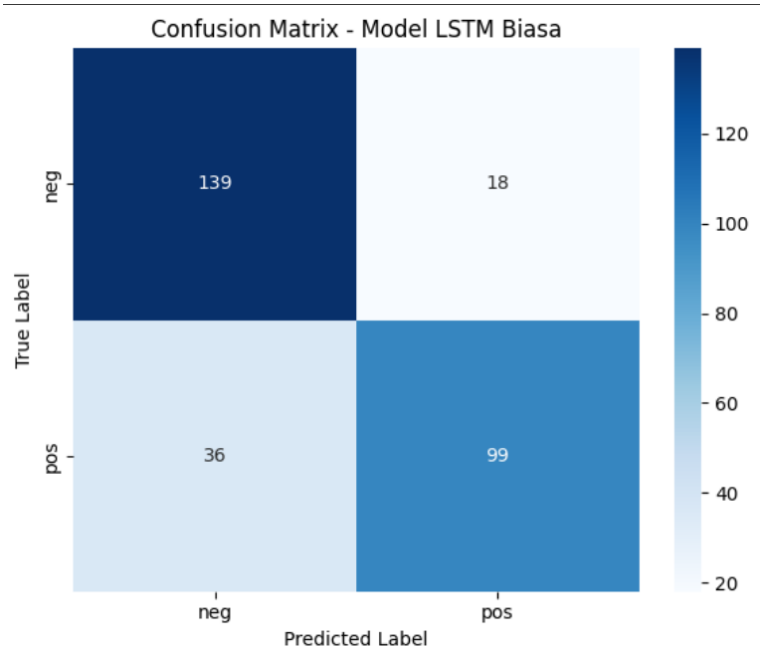
Total params: 672,900 (2.57 MB)
Trainable params: 672,900 (2.57 MB)
Non-trainable params: 0 (0.00 B)

Gambar 7. Arsitektur Model LSTM dengan FastText

Pada gambar 7, menunjukkan arsitektur model LSTM dengan FastText yang terdiri dari enam lapisan, dimulai dengan Embedding layer berdimensi 300 dengan total parameter 672.900. Lapisan berikutnya meliputi LSTM, dua Dropout, dan dua Dense layer, namun belum terbangun sepenuhnya (unbuilt) karena model belum dilatih atau belum diberi input shape. Saat ini, hanya embedding layer yang sudah memiliki parameter terhitung, sedangkan lapisan lainnya akan terisi setelah proses training atau saat model menerima input pertama.

3.4 Evaluasi Model

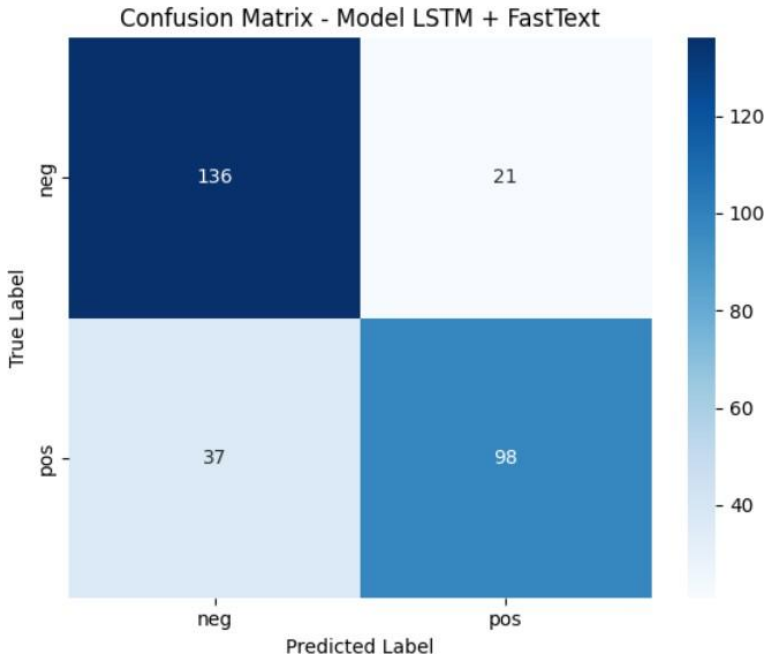
Model-model dibangun menggunakan library TensorFlow dan Keras dalam bahasa pemrograman Python. Data dibagi menjadi data training (80%) dan data testing (20%). Setelah model dilatih, evaluasi dilakukan menggunakan confusion matrix dan classification report untuk mendapatkan nilai presisi, akurasi, recall, dan F1-score. Berikut adalah hasil evaluasi untuk kedua model.



Gambar 8. Confusion Matrix LSTM Biasa (tanpa FastText)

p-ISSN: 2986-3929
e-ISSN: 3032-1948

Berdasarkan confusion matrix pada gambar 8, model LSTM Biasa(tanpa FastText) berhasil memprediksi sebanyak 139 data dengan sentimen negatif secara tepat. Namun, terdapat 18 data bersentimen negatif yang diprediksi sebagai positif. Selanjutnya, model juga berhasil memprediksi sebanyak 99 data dengan sentimen positif dengan benar, namun terdapat 36 data sentimen positif yang diprediksi sebagai negatif oleh model.



Gambar 9. Confusion Matrix LSTM dengan FastText

Kemudian berdasarkan gambar 9, model LSTM dengan FastText berhasil memprediksi sebanyak 136 data dengan sentimen negatif secara tepat. Tetapi, terdapat 21 data bersentimen negatif yang diprediksi sebagai positif. Kemudian, model LSTM dengan FastText juga berhasil memprediksi sebanyak 98 data dengan sentimen positif dengan benar, namun terdapat 37 data sentimen positif yang diprediksi sebagai negatif oleh model.

Evaluasi Model LSTM Biasa:

	precision	recall	f1-score	support
negatif	0.79	0.89	0.84	157
positif	0.85	0.73	0.79	135
accuracy			0.82	292
macro avg	0.82	0.81	0.81	292
weighted avg	0.82	0.82	0.81	292

Evaluasi Model LSTM + FastText:

	precision	recall	f1-score	support
negatif	0.79	0.87	0.82	157
positif	0.82	0.73	0.77	135
accuracy			0.80	292
macro avg	0.80	0.80	0.80	292
weighted avg	0.80	0.80	0.80	292

Gambar 10. Evaluasi Model LSTM Biasa (tanpa FastText) dan Model LSTM dengan FastText

Berdasarkan hasil evaluasi, model LSTM biasa memperoleh akurasi sebesar 82%, lebih tinggi dibanding model LSTM + FastText yang memiliki akurasi 80%. Selain itu, model LSTM biasa juga menunjukkan nilai F1-score yang lebih baik pada kedua kelas sentimen, khususnya pada kelas positif (0.79 dibanding 0.77). Hal ini menunjukkan bahwa model tanpa FastText justru lebih efektif dalam memahami gaya bahasa dalam komentar TikTok.

4. Kesimpulan

Berdasarkan hasil penelitian, dapat disimpulkan bahwa analisis sentimen terhadap pengguna TikTok mengenai progress IKN menggunakan LSTM dan FastText menunjukan pandangan masyarakat terhadap progress IKN masih negative yang menunjukan ada 52,84% data menunjukan sentiment negatif dan hanya 47.15% yang menunjukan sentiment positif. Kemudian hasil penelitia juga menunjukan jika model LSTM tanpa FastText memiliki performa yang lebih baik disbanding model LSTM dengan FastText. Ini terbukti dari model LSTM tanpa FastText menjunjukan hasil akurasi sebesar 82%, presisi 82%, recall 81%, dan skor-f1 81%. Sedangkan model LSTM dengan FastText menghasilkan akurasi sebesar 80%, presisi 80%, recall 80% dan skor-f1 80%. Hal ini menunjukkan bahwa model tanpa FastText justru lebih efektif dalam memahami gaya bahasa dalam komentar TikTok.

Daftar Pustaka

[1] Herdiana, D. (2022). Pemindahan Ibukota Negara: Upaya Pemerataan Pembangunan ataukah Mewujudkan Tata Pemerintahan yang Baik. Jurnal Transformative, 8(1), 1–30. https://doi.org/10.21776/ub.transformative.2022.008.01.1

[2] Amila, S., Nugraha, A. A., Sukron, A., & Rohmah, F. (2023). Analisis Dampak Dan Resiko Pemindahan Ibu Kota Negara Terhadap Ekonomi Di Indonesia. Jurnal Sahmiyya, 2(1), 10– 18.

[3] Rihastuti, S., & Rosyidi, A. (2025). Analisis Sentimen Pengguna Tiktok Tentang Progres Pembangunan Ikn Dengan Metode Random ForesT. Journal of Computer Science and Technology JCS-TECH, 5(1), 19–23.

[4] Pardede, J., & Pakpahan, I. (2023). Analisis Sentimen Penanganan Covid-19 Menggunakan Metode Long Short-Term Memory Pada Media Sosial Twitter. Jurnal Publikasi Teknik Informatika, 2(1), 12–25. https://doi.org/10.55606/jupti.v1i1.767

[5] Bojanowski, P., Grave, E., Joulin, A., & Mikolov, T. (2017). Enriching Word Vectors with Subword Information. Transactions of the Association for Computational Linguistics, 5, 135–146. https://doi.org/10.1162/tacl_a_00051/1567442/tacl_a_00051.

[6] Pangestu, A. F., Rahmat, B., & Sihananto, A. N. (2024). Analisis Sentimen Pada Media Sosial X Terhadap Implementasi Kurikulum Merdeka Menggunakan Metode Fasttext Dan Long Short-Term Memory (LSTM). JIPI (Jurnal Ilmiah Penelitian Dan Pembelajaran Informatika), 9(4), 2271–2280. https://doi.org/10.29100/jipi.v9i4.5665

[7] Islamy, M. A. A., Indriati, & Adikara, P. P. (2022). Analisis Sentimen IMDB Movie Reviews menggunakan Metode Long Short-Term Memory dan FastText. Jurnal Pengembangan Teknologi Informasi Dan Ilmu Komputer, 6(9), 4106–4115. http://j-ptiik.ub.ac.id

[8] Pipin, S. J., Kurniawan, H., Jurnal, |, & Mikroskil, S. (2022). Analisis Sentimen Kebijakan MBKM Berdasarkan Opini Masyarakat di Twitter Menggunakan LSTM. Jurnal SIFO Mikroskil, 23, 1–5.

[9] Najma Rafifah. (2025, February). Data Komentar Tiktok Progres IKN. Kaggle. https://www.kaggle.com/datasets/najmarafifah/analisis-sentimen-komentar-tiktok-progres-ikn

[10] Abadi, A. S. (2024). slangwords and stopwords bahasa indonesia. Kaggle. https://www.kaggle.com/datasets/ahmadseloabadi/slangwords-and-stopwords-bahasa-indonesia/data

p-ISSN: 2986-3929
e-ISSN: 3032-1948