Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer e-ISSN: 2548-964X
Vol. 1, No. 1, Januari 2017, hlm. x-x http://j-ptiik.ub.ac.id

Studi Perbandingan pada Metode CNN-LSTM dan LSTM dalam Mendeteksi Emosi pada Data Teks Berbahasa Indonesia pada Media Sosial Twitter

Alwan Fauzul Azhim Kustiwa[1], Muhammad Aminul Akbar[2], Aryo Pinandito[3]

Program Studi Sistem Informasi, Fakultas Ilmu Komputer, Universitas Brawijaya
Email: [1]alwanfa @student.ub.ac.id, [2]muhammad.aminul@ub.ac.id, , [3]aryo@ub.ac.id

Abstrak

Model deteksi emosi pada data teks memiliki banyak fungsi seperti sentimen analisis hingga cara untuk mengekstrak emosi dari teks yang kemudian di konfigurasikan pada *text-to-voice*. Salah satu model yang banyak digunakan dalam NLP deteksi emosi adalah LSTM. Menurut beberapa penelitian performa dari LSTM ini bisa ditingkatkan dengan menggabungkannya dengan CNN menjadi CNN-LSTM. Dalam penelitian ini kami akan meneliti perbandingan antara CNN-LSTM dan LSTM dalam mendeteksi emosi ketika dilatih dan dievaluasi menggunakan data Bahasa Indonesia sehari-hari. Penelitian dilakukan dengan melatih dan mengevaluasi model CNN-LSTM dan LSTM dengan menggunakan metode *stratified K-Fold Cross Validation* untuk melihat konsistensi performa model. Dataset yang digunakan merupakan Bahasa Indonesia sehari-hari yang bersumber dari media sosial. CNN-LSTM dan LSTM akan dilatih dan diuji menggunakan pasangan train-test yang sama pada setiap fold sehingga data performa dari kedua model akan saling berpasangan pada setiap fold. Performa yang diuji antara lain akurasi, *precision*, *recall*, *f1-score* dan *loss* yang berupa *cross entropy loss*. Distribusi data performa dari setiap model dilakukan uji hipotesis beda antara kedua model pada setiap metrik. Dari hasil pengujian statistik yang dilakukan, dari 5 metrik yang diuji menunjukan CNN-LSTM hanya memiliki *loss* yang signifikan lebih rendah daripada LSTM. Sedangkan pada 4 metrik lainnya menunjukan tidak ada perbedaan yang signifikan.

Kata kunci: *Deteksi Emosi, Klasifikasi, Bahasa Indonesia, CNN-LSTM, LSTM, Uji T Berpasangan, K-Fold Cross Validation*

Abstract

Emotion detection models on text data have many functions such as sentiment analysis to ways to extract emotions from text which are then configured in text-to-voice. One model that is widely used in NLP emotion detection is LSTM. According to some research, the performance of LSTM can be improved by combining it with CNN into CNN-LSTM. In this study we will examine the comparison between CNN-LSTM and LSTM in detecting emotions when trained and evaluated using everyday Indonesian data. The research is conducted by training and evaluating CNN-LSTM and LSTM models using the stratified K-Fold Cross Validation method to see the consistency of model performance. The dataset used is everyday Indonesian language sourced from social media. CNN-LSTM and LSTM will be trained and tested using the same train-test pair in each fold so that the performance data of the two models will be paired in each fold. The performance tested includes accuracy, precision, recall, f1-score and loss in the form of cross entropy loss. The distribution of performance data from each model is carried out hypothesis testing of the difference between the two models on each metric. From the results of statistical testing carried out, of the 5 metrics tested, CNN-LSTM only has a loss that is significantly lower than LSTM. While the other 4 metrics show no significant difference

Keywords: *Emotion Detection, Classification, Indonesian Language, CNN-LSTM, LSTM, Paired T Test, K-Fold Cross Validation*

1. PENDAHULUAN

Deteksi emosi pada teks selain berguna sebagai sarana untuk mendeteksi sentimen yang spesifik pada emosi dimedia sosial, deteksi emosi juga dapat digunakan untuk menambah

variasi dari *fitur text-to-voice. Text to voice* yang sekarang ini cenderung mengeluarkan suara yang netral tanpa emosi. Dengan mendeteksi emosi pada teks, kita dapat mengkonfigurasikan output suara dari *text-to-voice* sesuai dengan emosi yang terdeteksi. Sehingga output suara nantinya dapat memiliki nuansa yang lebih beremosi. Pemilihan model deteksi emosi dengan performa yang lebih baik menjadi sangat penting di sini karena semakin bagus performa model tersebut, semakin representatif suara yang dihasilkan fitur *text-to-voice* terhadap emosi yang dimaksud.

Salah satu cara yang dapat digunakan untuk mendeteksi emosi yaitu dengan menggunakan pembelajaran mesin. Data teks yang menjadi objek deteksi merupakan sebuah jenis data sekuential berupa kata yang berurutan. Dalam mendeteksi informasi pada data sekuential salah satu metode yang banyak digunakan yaitu pembelajaran berbasis *recurrent learning* seperti RNN dan LSTM dimana setiap kata pada sebuah kalimat akan mempengaruhi informasi dari kata-kata lainnya pada kalimat. Banyaknya metode yang dapat digunakan untuk mendeteksi emosi ini

Pada penelitian Nisa et al (2021) yang membandingkan performa antara model-model *machine learning* konvensional dengan model-model *deep learning* dalam mengklasifikasikan emosi dalam berbagai dataset. Pada salah satu peneltiannya pada dataset *Tweet Pemilu* bahasa Indonesia, LSTM dan GRU memiliki akurasi tertinggi dengan angka 90,90% dan 90,91%. Pada dataset *CrowdFlower* yang berisi data teks dari berbagai sumber, LSTM, RNN, dan GRU mendapat tingkat akurasi tertinggi dengan angka 92,30%. Dari penelitian ini, secara keseluruhan metode *deep learning* terutama LSTM dan GRU memiliki akurasi yang lebih tinggi jika dibandingkan dengan metode lain dalam mengklasifikasikan emosi pada data teks Bahasa Indonesia. Penelitian lain dalam klasifikasi emosi bahasa Indonesia dilakukan oleh Riza dan Charibaldi(2021) menggunakan data hasil pengumpulan pribadi dari sumber twitter menggunakan algoritma LSTM menghasilkan performa akurasi tertinggi 73,15%.

Dari beberapa penelitian sebelumnya, performa LSTM dalam model klasifikasi emosi menunjukan kinerja yang baik dalam segi akurasi, *precision, recall,* dan *f-1 score*. LSTM yang memiliki kemampuan yang baik dalam menangkap informasi secara kontekstual membuat LSTM memiliki performa yang cukup baik ketika diterapkan pada kasus pemrosesan bahasa alami. Tetapi dalam proses penangkapan informasi secara kontekstual pada data teks dapat mengarahkan model pada bias(Yang et al., 2019). Menurut Yang et al (2019), RNN-*based* model merupakan sebuah model yang cukup bias karena semakin mendekati akhir teks, informasi yang ditangkap akan semakin banyak dibandingkan dengan kata-kata sebelumnya. Pada kasus deteksi emosi kata, fitur kunci seperti kata emosi dapat muncul dimana saja tidak hanya diakhir kata. Ketika RNN atau LSTM menangkap informasi semantik dari keseluruhan kalimat, perubahan posisi dari fitur penting dapat mengakibatkan penurunan efektifitas, bahkan mungkin untuk mengabaikan informasi penting dari kata tersebut.

Salah satu solusi yang dapat dilakukan untuk meningkatkan performa dari LSTM agar dapat lebih menangkap kata penting pada data teks yaitu dengan mengkombinasikannya dengan *convolutional layer* pada CNN. *Convolutional layer* dapat menangkap lokal fitur yang cukup penting. Sehingga nantinya model hasil kombinasi CNN dan LSTM dapat menangkap informasi dari kata penting atau secara spasial dan makna kontekstual sekaligus. Pada penelitian yang dilakukan oleh Ankita et al (2022) yang membandingkan model sentimen analisis berdasarkan emosi mengenai topik *black lives matter* di beberapa negara bagian Amerika Serikat, CNN-LSTM mendapat tingkat akurasi yang lebih tinggi dari LSTM, dan beberapa model lainnya. Penelitian lain dilakukan oleh Ullah et al (2022) mengenai deteksi emosi dan analisis sentimen pada Bahasa Urdu menggunakan CNN-LSTM. Performa dari CNN-LSTM ini lebih baik setelah dibandingkan dengan model LSTM dan beberapa model lainnya.

Dari beberapa hasil penelitian sebelumnya, performa CNN-LSTM dalam mengklasifikasikan emosi pada dataset bahasa Inggris dan dataset Bahasa Urdu lebih baik daripada menggunakan LSTM. Hal ini menarik untuk diteliti apakah hal serupa akan terjadi pada dataset dengan bahasa Indonesia. Pasalnya

setiap bahasa memiliki karakteristik yang berbeda-beda. Seperti struktur kalimat, penggunaan kata, *stopwords*, *slang*, konteks kebudayaan, dan lain sebagainya. Oleh karena itu pada penelitian ini akan membandingkan performa model kombinasi dari CNN dan LSTM dengan model LSTM, apakah CNN-LSTM akan memiliki performa yang lebih baik daripada LSTM seperti pada kasus penelitian sebelumnya pada penelitian Ankita et al (2022) dan Ullah et al (2022).

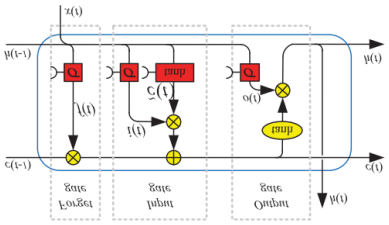
2. LANDASAN TEORI

2.1 Fasttext

Fasttext merupakan salah satu metode *word embedding* yang representasi vektornya tidak berdasarkan pada satuan kata melainkan *n-gram* karakter dari suatu kata. Kata akan direpresentasikan sebagai hasil penjumlahan vektor *dari n-gram* yang menyusun kata tersebut (Bojanowski et al., 2016). *Fasttext* dapat mengelola kata kata unik, tidak standar, ataupun salah pengetikan selama n-gram yang menyusun kata tersebut ada dalam kamus model *fasttext*. Oleh karena itu *fasttext* dianggap penulis cocok untuk data kata yang tidak standar seperti data yang bersumber dari media sosial.

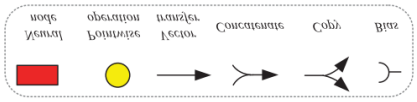
2.2 Long Short Term Memories (LSTM)

LSTM merupakan salah satu jenis dari Recurrent Neural Network (RNN) yang memungkinkan untuk mempelajari ketergantungan jangka panjang yang dikemukakan oleh Hochreiter & Schmidhubber (1997). LSTM di buat untuk mengendalikan masalah dari *vanishing gradient* dalam kasus *long-term dependancies* pada RNN (Yu et al., 2019). Struktur sebuah segmen LSTM dapat dilihat pada Gambar 1 dan 2.



Gambar 1 Arsitektur LSTM dengan *forget gate*

Sumber : Yu et al (2019)

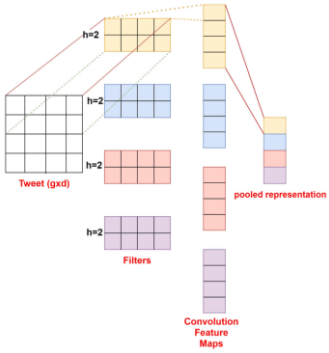


Gambar 2 Keterangan simbol pada gambar 3

Sumber : Yu et al (2019)

2.3 *Convolutional Neural Network* (CNN)

CNN merupakan arsitektur *deep learning* yang banyak digunakan untuk *image processing*. Namun saat ini cukup banyak penelitian yang menggunakan CNN untuk *text processing* karena hasilnya yang cukup menjanjikan. Hal ini terbukti dari beberapa penelitian yang dilakukan sebelumnya oleh Pradana Rachman et al (2021), CNN mendapatkan tingkat akurasi paling optimal. Hal ini dikarenakan pada CNN terdapat berbagai lapisan konvolusional yang dapat menangkap lokal atau global fitur pada sebuah vektor embedding (Ankita et al., 2022).



Gambar 3. Arsitektur CNN pada vektor *Natural Language*

Sumber : Ankita et al. (2022)

Pada pemrosesan teks pada penelitian ini akan menggunakan 2 layer CNN yakni *Convolutional layer* dan *Pooling layer*. Pada *Convolutional Layer* akan dilakukan ekstraksi fitur pada vektor embedding berdasakkran filternya.

$$m_r = L \cdot r_i \quad (2.1)$$

m_r merupakan nilai skalar dari hasil operasi dot antara vektor input dengan filternya. L merupakan vektor filter dan r merupakan sub vektor dari vektor embed sesuai ukuran dengan filter yang di geser sesuai stridenya. Himpunan dari mr nantinya akan menjadi *convolution*

feature map m.

Setelah menjadi *feature map*, dimensi dari fitur map akan direduksi menggunakan pooling layer. Salah satu dari jenis pooling layer yaitu max-pooling. dimana B merupakan nilai maksimal dari semua anggota vektor m.

$$b = max\,[m[i]] \quad (2.2)$$

2.4 K-Fold Cross Validation

K-Fold cross validation (KCV) merupakan teknik yang digunakan untuk evaluasi model. KCV dilakukan dengan membagi dataset menjadi K jumlah segmen dataset (Anguita et al., 2012). Setelah itu secara iteratif K-1 segmen data akan digunakan sebagai bahan pembelajaran model dan 1 segmen akan digunakan untuk melakukan validasi model.

3. METODOLOGI PENELITIAN

3.1 Pengumpulan Data

Data utama yang akan digunakan pada penelitian ini merupakan data sekunder dari hasil pengumpulan Riccosan et al. (2022). Dataset ini berisi 7080 buah opini publik berbahasa Indonesia dari media sosial Twitter yang telah dianotasikan. Anotasi dari dataset ini terdiri dari 6 label emosi yaitu *anger*, *fear*, *joy*, *love*, *sadness*, dan *neutral*. Data ini telah dilakukan preprocessing sederhana seperti normalisasi *hashtag*, normalisasi duplikasi, *casefolding*.

3.2 Preprocessing Data

Sebelum data dapat digunakan untuk melatih model CNN-LSTM dan LSTM, data sebelumnya harus dinormalisasi terlebih dahulu. Beberapa normalisasi yang akan dilakukan antara lain menghapus tanda baca dan angka, tokenisasi teks, dan normalisasi *slang*, *POS tagging* hingga *stopwords removal*.

3.2.1 Penghapusan tanda baca

Pada dataset yang digunakan dalam penelitian ini sebelumnya telah dilakukan pembersihan karakter berupa hashtag. Tetapi pada dataset ini masih tersisa beberapa tanda tanya seperti titik, koma, tanda tanya, hingga tanda emoticon. Hal ini dilakukan untuk menyederhanakan teks supaya model nantinya fokus pada pembelajaran pada teks saja.

3.2.2 Normalisasi kata slang

Dataset penelitian ini memiliki varian kata yang sangat beragam karena dataset bersumber dari tweet masyarakat yang tidak ada batasan jenis kata yang digunakan. Dari hasil penelusuran, dataset ini memiliki 17589 kata unik. Kata yang digunakan pun beragam mulai dari kata gaul, salah ketikan, karakter awal kata atau akhir yang berulang seperti “haloooo”, hingga ragam cara tertawa masyarakat yang beragam seperti “wkwk”, “hahah”, “hehe” dan lain sebagainya. Hal ini membuat suatu makna dapat memiliki kata yang beragam. Hal ini akan menyulitkan model dalam mengekstrak fitur pada kata karena bisa jadi pada beberapa kata memiliki arti yang sama tetapi memiliki vektor kata yang berbeda. Selain itu, banyaknya kata unik dapat membuat suatu kata sedikit muncul yang membuat model menjadi *underfitting* terhadap suatu kata tersebut.

3.2.3 Part of Speech (POS) *Tagging*

POS Tagging dilakukan untuk menandai kata berdasarkan jenis kata apakah itu subjek, predikat, objek atau keterangan. Sehingga kata yang memiliki bentukan kata yang sama dapat dibedakan berdasarkan posisi kata di suatu kalimat. POS Tagging dilakukan menggunakan pretrained model *POS Tagging* yang berasal dari platform *Huggingface*.

3.2.4 *Stopwords Removal*

Proses ini dilakukan untuk mengurasi *noise* pada data dengan menghilangkan stopwords yang dianggap tidak memiliki makna penting dalam penentuan emosi. Data *stopwords* diambil dari Pustaka *Natural Language Toolkit* (NLTK). Tidak seluruh kata *stopwords* dari NLTK pada dataset dihapus. Beberapa kata seperti kata ganti orang, kata emosi, negasi, hingga kata pengandaian tidak dihapus dari dataset karena mempengaruhi dalam penentuan emosi.

3.2.5 Penghapusan Kalimat Panjang

Pada dataset yang telah diolah sebelumnya, terdapat data row yang memiliki panjang hingga 706 kata pada 1 tweet. Karena ketika proses

training data harus memiliki jumlah data input yang sama dan untuk mengurangi dimensi data yang terlalu besar, beberapa tweet yang panjang akan dihapus untuk mengurangi jumlah input yang terlalu banyak. Pada penelitian ini ditetapkan jumlah input maksimal adalah 100 kata.

3.2.6 Tokenisasi dan *Encoding* Kata

Proses tokenisasi dilakukan dengan menggunakan Tokenizer. Tokenizer merupakan sutu utilitas yang berasal dari librari keras. Utilitas Tokenizer selain digunakan untuk memecah kalimat menjadi token yang lebih kecil, tool ini juga dapat digunakan untuk membuat kamus token atau kata beserta indeks yang nantinya akan berfungsi untuk membuat *embedding matrix* untuk inisialisasi *weight* pada *embedding layer*.

3.2.7 Penambahan *Padding* pada data

Penambahan *padding* dilakukan untuk menyeragamkan panjang dari setiap data. Pada proses ini dilakukan dengan menggunakan salah satu utilitas dari keras yaitu keras.preprocessing.sequence.pad_sequences. Pada penelitian ini, panjang maximum yang digunakan adalah 100 kata per baris. Data yang memiliki jumlah kata kurang dari 100 akan ditambahkan *padding* angka 0. Argumen *padding* 'pre' menjadikan padding pada bagian awal kalimat diisi menggunakan angka 0.

3.3 Word Embedding

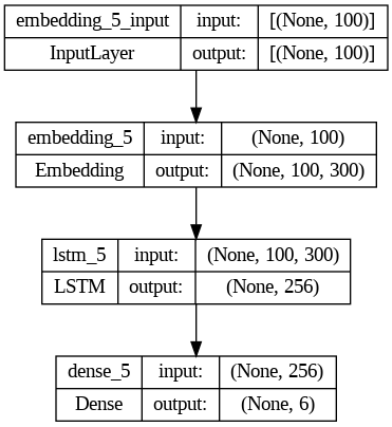
Proses Word Embedding dilakukan dengan membuat embedding matrix yang setiapData embedding setiap kata akan diambil dari model *fasttext* yang telah dimuat sebelumnya. Setiap embedding kata nantinya akan dimuat pada matrix dengan indeks yang sesuai dengan kamus tokenizer.

3.4 Perancangan Model

3.4.1 LSTM

Arsitektur model LSTM terdiri atas *input layer*, kemudian masuk ke embedding layer yang memetakan setiap kata menjadi vector berdimensi 300. Keluaran dari *embedding layer* akan menjadi input pada LSTM layer. Terakhir sebuah dense layer sebagai lapisan keluaran yang menggunakan *activation function softmax* supaya keluaran dari model berupa probabilitas deteksi tiap label emosi. Arsitektur LSTM dapat dilihat pada Gambar 4.

Setelah rancangan arsitektur dilakukan, proses hyperparameter eksperimen dilakukan untuk mendapatkan set parameter yang memiliki performa baik. Eksperimen parameter dilakukan dengan menggunakan *train test* pada dataset dengan rasio 8:2 dan pengaturan *hyperparameter* dilakukan secara manual dengan mengubah *hyperparameter* satu per satu. Hasil optimasi dari LSTM dapat dilihat pada Tabel 1.



Gambar 4. Arsitektur model LSTM

Tabel 1. Parameter setup LSTM

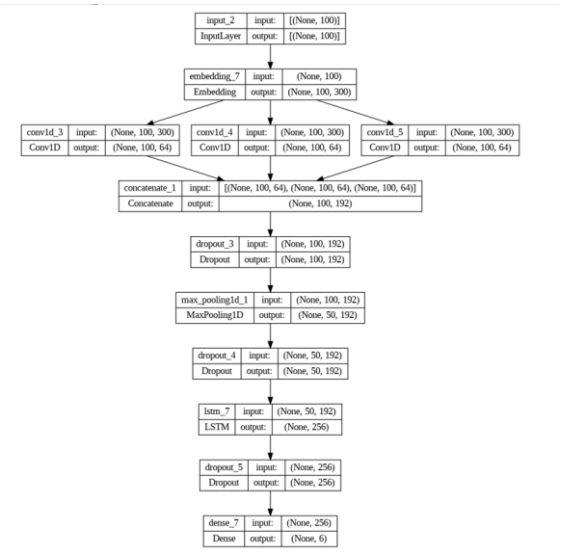
Parameter	Value
LSTM Unit	256
Dropout Rate	0.3
Optimizer	Adam
Learning rate	0.001
Batch size	64
Epochs	50

3.4.2 CNN-LSTM

Struktur model CNN-LSTM yang digunakan pada penelitian ini terinspirasi dari penelitian yang dilakukan oleh Ullah et al., (2022). Output dari embedding layer akan menjadi input 3 buah *convolution layer* yang memiliki ukuran filter yang berbeda. Keluaran dari 3 *buah convolution layer* yang memiliki dimensi yang sama digabung menjadi 1 buah matriks dengan menggunakan *concatenate* yang merupakan salah satu utilitas pada *keras*. Matriks output yang telah digabung kemudian

akan masuk ke dalam *pooling layer*. Dari *pooling layer* akan masuk ke dalam lapisan LSTM yang kemudian akan diteruskan ke dalam dense layer *softmax* untuk mendapat output berupa vektor probabilitas kelas.

Setelah rancangan arsitektur dilakukan, proses hyperparameter eksperimen dilakukan untuk mendapatkan set parameter yang memiliki performa baik. Eksperimen parameter dilakukan dengan menggunakan *train test* pada dataset dengan rasio 8:2 dan pengaturan *hyperparameter* dilakukan secara manual dengan mengubah *hyperparameter* satu per satu. Hasil optimasi dari CNN-LSTM dapat dilihat pada Tabel 2.



Gambar 5. Arsitektur model LSTM

Tabel 2. Parameter setup CNN-LSTM

Parameter	Value
LSTM Unit	256
Kernel size	4, 6, 8
Number of kernel	64
Dropout Rate	0.5
Optimizer	Adam
Learning rate	0.0001

3.5 Evaluasi Model

Proses evaluasi model pada penelitian ini dilakukan dengan menggunakan metode *Stratified K-Fold Cross Validation*. Dataset yang setelah dibersihkan berjumlah 7066 baris data akan dibagi menjadi 30 *fold* dengan setiap *fold* berjumlah 236 atau 235 baris data. Model akan dilatih menggunakan 29 *fold* dan akan dievaluasi menggunakan 1 *fold* sisa. *Fold* evaluasi akan diiterasi hingga setiap *fold* pernah menjadi data evaluasi. Data yang dihasilkan pada proses evaluasi ini berupa 30 buah nilai untuk setiap metrik performa yang dihitung.

4. Hasil dan Pembahasan

4.1 Data performa

Data yang telah dikumpulkan merupakan performa akurasi, *cross entropy loss*, *precision*, *recall*, dan *F1-score* dari 30 model CNN-LSTM dan LSTM yang telah dilatih menggunakan pasangan *train test* dari 30 *fold*. Seluruh data penelitian terdapat pada Lampiran A Data rata-rata dan standar deviasi dari tiap performa dapat dilihat pada Tabel 3.

Tabel 3. Rata-rata performa model CNN-LSTM dan LSTM

Metrik	Mean	
	LSTM	CNN-LSTM
Akurasi	0.7072	**0.7099**
Loss	0.8212	**0.7993**
Precision	**0.7187**	0.7183
Recall	0.7244	**0.7273**
F1-score	0.7165	**0.7182**

4.2 Analisis Data

4.2.1 Uji Normalitas

Uji normalitas dilakukan menggunakan metode *Shapiro-Wilk* yang tersedia pada pustaka *scipy* pada bahasa pemrograman *python*. Uji normalitas ini dilakukan untuk menguji apakah sampel data berdistribusi normal.

Tabel 4. Hasil Uji Normalitas

Metrics	p-value	
	LSTM	CNN-LSTM
Akurasi	0.8239	0.6099
Loss	0.3129	0.4289
Macro Precision	0.5636	0.9698
Macro Recall	0.6755	0.7174
Macro F1-score	0.9298	0.7013

Dalam uji normalitas, H_0 ditolak apabila nilai p lebih kecil daripada alpha atau level signifikansi. Pada penelitian ini, level

signifikansi yang digunakan yaitu 0,05. Dari data pada Tabel 4 seluruh distribusi data memiliki nilai p dari di atas 0,05 yang berarti H_0 gagal ditolak dan dapat diasumsikan bahwa seluruh data terdistribusi normal.

4.2.2 Uji Beda

Setelah seluruh data dapat diasumsikan terdistribusi normal, uji beda dilakukan menggunakan uji T berpasangan untuk membuktikan apakah ada perbedaan signifikan antara data masing-masing metrik performa dari model LSTM dan CNN-LSTM. Hasil dari uji beda dapat dilihat pada Tabel 5.

Tabel 5. Hasil Uji T Berpasangan

Metrik	Selisih data berpasangan							
	Mean	*Std. Deviation*	*Std. Error Mean*	*95% Confidence Interval of the Differemce*		*T*	*df*	*p-value (two tailed)*
				Lower	Upper			
Akurasi	-0.0027	0.0262	0.0048	-0.0126	0.0072	-0.5557	29	0.5826
Loss	0.0218	0.0381	0.0071	0.0074	0.0363	3.0955	29	0.0043
Macro Avg Precision	0.0005	0.0274	0.0051	-0.0099	0.0109	0.0905	29	0.9285
Macro Average Recall	-0.0029	0.0258	0.0048	-0.0126	0.0069	0.5979	29	0.5545
Macro Average F1-Score	-0.0017	0.0252	0.0047	-0.0113	0.0078	-0.3708	29	0.7135

4.2.3 Pembahasan

Pada data performa model CNN-LSTM dan LSTM yang telah dilatih menggunakan dataset yang telah dinormalisasi dilakukan uji beda menggunakan uji T berpasangan. Pada Tabel 5 menunjukan bahwa dari seluruh metrik klasifikasi berdasarkan *confusion matrix* seperti akurasi, *macro average precision*, *recall*, dan f1-score memiliki p-value di atas alpha yaitu 0,05. Hal menunjukan bahwa untuk dari uji beda pada metrik metrik tersebut tidak memiliki bukti yang cukup untuk menolak H_0 yaitu kedua model tidak memiliki perbedaan signifikan.

Sedangkan pada metrik *loss* yang menggunakan *cross entropy loss* menghasilkan nilai p 0.0043 yang lebih kecil dari *alpha* yaitu 0.05 yang menunjukan ada bukti yang cukup untuk menolak H_0 pada metrik *loss*. Sehingga untuk metrik *loss* antara CNN-LSTM dan LSTM terdapat perbedaan signifikan dengan rata-rata selisih antara *loss* LSTM dan CNN-LSTM yaitu 0.0218 dengan CNN-LSTM lebih rendah daripada LSTM. Loss yang lebih rendah pada CNN-LSTM daripada LSTM menunjukan bahwa model hanya lebih confidence ketika mengklasifikasikan emosi dibandingkan LSTM. Tetapi CNN-LSTM tidak membuat model memprediksi emosi yang benar lebih banyak daripada LSTM.

Dari hasil analisis sebelumnya dapat ditarik kesimpulan bahwa pada dataset Riccosan et al (2022) CNN-LSTM hanya memiliki nilai loss yang lebih rendah sedikit dibandingkan dengan LSTM. Hal ini berbeda dari hasil penelitian sebelumnya dimana CNN-LSTM dapat mengungguli LSTM di berbagai metrik. Perbedaan ini mungkin terjadi karena karakteristik dataset yang digunakan dalam proses pelatihan dan evaluasi. Pada penelitian ini, dataset Bahasa Indonesia yang digunakan telah dinormalisasi, sehingga memiliki karakteristik bahasa yang berbeda dibandingkan dengan dataset yang digunakan dalam penelitian sebelumnya. Selain itu, perbedaan dalam jumlah data juga dapat menjadi penyebab performa yang serupa antara CNN-LSTM dan LSTM. Dukungan untuk hal ini dapat ditemukan dalam penelitian Brigato & Iocchi (2020), di mana jaringan saraf tiruan dengan kompleksitas rendah dapat berfungsi dengan baik atau bahkan lebih baik daripada model-model mutakhir ketika bekerja dengan dataset yang terbatas.

5. KESIMPULAN DAN SARAN

Dari hasil penelitian menggunakan data

evaluasi 30 fold cross validation didapatkan 30 data performa akurasi, precision, recall, f1-score, dan juga loss dari kedua model, didapatkan bahwa hasil pengujian statistik menggunakan uji T berpasangan, dari kelima metrik yang diuji hanya 1 metrik yang menunjukan adanya perbedaan signifikan yaitu loss. Secara perbandingan CNN-LSTM menunjukan tidak ada perbedaan performa pada metrik klasifikasi dibandingkan dengan model LSTM yang sama-sama dilatih menggunakan data Bahasa Indonesia sehari-hari yang telah dinormalisasi.

Agar penelitian dapat menghasilkan data perbandingan yang lebih baik dan lebih representatif terhadap performa pada data tak terlihat, sebaiknya dilakukan optimasi *hyperparameter* dengan metode yang sama pada setiap model yang dibandingkan tanpa melibatkan data uji dalam proses optimasi. Keterbatasan waktu dan sumber daya menjadi penyebab pada penelitian ini, proses optimasi masih dilakukan secara manual dan optimasi hyperparameter masih dilakukan dengan melibatkan data tes.

Selain itu untuk penelitian selanjutnya juga dapat melakukan studi perbandingan antara model CNN-LSTM dan LSTM pada berbagai kasus klasifikasi teks lainnya. Keterbatasan sumber dataset dalam penelitian ini membatasi ruang lingkup penelitian hanya pada klasifikasi emosi menggunakan dataset bahasa Indonesia dari media sosial Twitter. Penelitian lanjutan yang melibatkan berbagai jenis dataset teks dapat memberikan pemahaman yang lebih komprehensif mengenai performa model-model tersebut dalam konteks yang berbeda.

6. DAFTAR PUSTAKA

Akbarianto Wibowo, H., Nindyatama Nityasya, M., Feyza AkyürekAky, A., Suci Fitriany, A., Fikri Aji, A., Eko Prasojo, R., & Tanti Wijaya, D. (2021). IndoCollex: A Testbed for Morphological Transformation of Indonesian Colloquial Words. www.kaggle.com/grikomsn/lazada-indonesian-reviews

Alfat, L., Nasucha, M., Uddin, N., Salleh, K. A., & Baharun, N. (2023). Sentiment Classification of Indonesian Emotion Related to Vaccination Event using LSTM. 2023 IEEE World AI IoT Congress, AIIoT 2023, 825–829. https://doi.org/10.1109/AIIoT58121.2023.10174581

Anguita, D., Ghelardoni, L., Ghio, A., Oneto, L., & Ridella, S. (2012). The "K" in K-fold Cross Validation. http://www.i6doc.com/en/livre/?GCOI=28001100967420.

Ankita, Rani, S., Bashir, A. K., Alhudhaif, A., Koundal, D., & Gunduz, E. S. (2022). An efficient CNN-LSTM model for sentiment detection in #BlackLivesMatter. Expert Systems with Applications, 193. https://doi.org/10.1016/j.eswa.2021.116256

Bojanowski, P., Grave, E., Joulin, A., & Mikolov, T. (2016). Enriching Word Vectors with Subword Information. http://arxiv.org/abs/1607.04606

Brigato, L., & Iocchi, L. (2020). A close look at deep learning with small data. Proceedings - International Conference on Pattern Recognition, 2490–2497. https://doi.org/10.1109/ICPR48806.2021.9412492

Chopra, A., Prashar, A., & Sain, C. (2013). Natural Language Processing. INTERNATIONAL JOURNAL OF TECHNOLOGY ENHANCEMENTS AND EMERGING ENGINEERING RESEARCH, 1(4). http://en.wikipedia.org/wiki/

Haddi, E., Liu, X., & Shi, Y. (2013). The role of text pre-processing in sentiment analysis. Procedia Computer Science, 17, 26–32. https://doi.org/10.1016/j.procs.2013.05.005

Hanusz, Z., Tarasinska, J., & Zielinski, W. (2016). SHAPIRO-WILK TEST WITH KNOWN MEAN. In REVSTAT-Statistical Journal (Vol. 14, Issue 1).

Heldiansyah, M. F., & Winarko, E. (2022). Emotion Detection on Indonesian Tweets Using CNN and Contextualized Word Embedding. Proceedings of 2022 International Conference on Data and Software Engineering, ICoDSE 2022, 53–58. https://doi.org/10.1109/ICoDSE56892.2022.9972229

Ho, Y., & Wookey, S. (2020). The Real-World-Weight Cross-Entropy Loss Function: Modeling the Costs of Mislabeling. IEEE Access, 8, 4806–4813. https://doi.org/10.1109/ACCESS.2019.2962617

Hochreiter, S., & Schmidhubber, J. (1997). LONG SHORT-TERM MEMORY. Neural Computation, 9(8), 1735–1780.

Hsu, H., & Lachenbruch, P. A. (2005). Paired t Test.

Mrhar, K., Benhiba, L., Bourekkache, S., & Abik, M. (2021). A Bayesian CNN-LSTM Model for Sentiment Analysis in Massive Open Online Courses MOOCs. International Journal of Emerging Technologies in Learning, 16(23), 216–232. https://doi.org/10.3991/ijet.v16i23.24457

Nisa, R., Amriza, S., & Supriyadi, D. (2021). Komparasi Metode Machine Learning dan Deep Learning untuk Deteksi Emosi pada Text di Sosial Media Klasifikasi Loyalitas Pengguna Sistem E-Learning Menggunakan Net Promoter Score dan Machine Learning View project. JUPITER (Jurnal Pendidikan Teknik Elektro). https://doi.org/10.5281/3603.jupiter.2021.10

Pradana Rachman, F., Santoso, H., & History, A. (2021). Jurnal Teknologi dan Manajemen Informatika Perbandingan Model Deep Learning untuk Klasifikasi Sentiment Analysis dengan Teknik Natural Languange Processing Article Info ABSTRACT. 7(2), 103–112. http://http://jurnal.unmer.ac.id/index.php/jtmi

Riccosan, Saputra, K. E., Pratama, G. D., & Chowanda, A. (2022). Emotion dataset from Indonesian public opinion. Data in Brief, 43. https://doi.org/10.1016/j.dib.2022.108465

Riza, M. A., & Charibaldi, N. (2021). Emotion Detection in Twitter Social Media Using Long Short-Term Memory (LSTM) and Fast Text. International Journal of Artificial Intelligence & Robotics (IJAIR), 3(1), 15–26. https://doi.org/10.25139/ijair.v3i1.3827

Rong, X. (2014). word2vec Parameter Learning Explained. http://arxiv.org/abs/1411.2738

Salsabila, N. A., Winatmoko, Y. A., Septriandri, A. A., & Jamal, A. (2018). Colloquial Indonesian Lexicon. International Conference on Asian Language Processing (IALP).

Savigny, J., & Purwarianti, A. (2017). Emotion Classification on Youtube Comments using Word Embedding.

Sehu Mohamad, Z., & Mohd Hashim, I. H. (2020). EMOTIONAL EXPERIENCES DURING HAJJ: A LITERATURE. International Journal of Education, Psychology and Counseling, 5(34), 01–09. https://doi.org/10.35631/ijepc.534001

Tri Hermanto, D., Setyanto, A., & Luthfi, E. T. (2021). Algoritma LSTM-CNN untuk Sentimen Klasifikasi dengan Word2vec pada Media Online. Citec Journal, 8.

Ullah, F., Chen, X., Shah, S. B. H., Mahfoudh, S., Hassan, M. A., & Saeed, N. (2022). A Novel Approach for Emotion Detection and Sentiment Analysis for Low Resource Urdu Language Based on CNN-LSTM. Electronics (Switzerland), 11(24). https://doi.org/10.3390/electronics11244096

Vujović, Ž. (2021). Classification Model Evaluation Metrics. International Journal of Advanced Computer Science and Applications, 12(6), 599–606. https://doi.org/10.14569/IJACSA.2021.0120670

Wang, B., Wang, A., Chen, F., Wang, Y., & Kuo, C. C. J. (2019). Evaluating word embedding models: Methods and experimental results. In APSIPA Transactions on Signal and Information Processing (Vol. 8). Cambridge University Press. https://doi.org/10.1017/ATSIP.2019.12

Yang, F., Du, C., Huang, L., Yang, F., & Huang, L. (2019). Ensemble Sentiment Analysis Method based on R-CNN and C-RNN with Fusion Gate. In INTERNATIONAL JOURNAL OF COMPUTERS COMMUNICATIONS & CONTROL (Vol. 14, Issue 2).

Yu, Y., Si, X., Hu, C., & Zhang, J. (2019). A review of recurrent neural networks: Lstm cells and network architectures. In Neural Computation (Vol. 31, Issue 7, pp. 1235–1270). MIT Press Journals. https://doi.org/10.1162/neco_a_01199