

Algoritma LSTM-CNN untuk Sentimen Klasifikasi dengan Word2vec pada Media Online

LSTM-CNN Algorithm for Sentiment Clasification with Word2vec On Online Media

Dedi Tri Hermanto^{*1}, Arief Setyanto², Emha Taufiq Luthfi³

^{1,2,3}Universitas Amikom Yogyakarta

E-mail: ^{*1} dedi.hermanto@students.amikom.ac.id, ² arief_s@amikom.ac.id,

² emhataufiqluthfi@amikom.ac.id

Abstrak

Media online banyak menghasilkan berbagai macam berita, baik ekonomi, politik, kesehatan, olahraga atau ilmu pengetahuan. Di antara itu semua, ekonomi adalah salah satu topik menarik untuk dibahas. Ekonomi memiliki dampak langsung kepada warga negara, perusahaan, bahkan pasar tradisional tergantung pada kondisi ekonomi di suatu negara. Sentimen yang terkandung dalam berita dapat mempengaruhi pandangan masyarakat terhadap suatu hal atau kebijakan pemerintah. Topik ekonomi adalah bahasan yang menarik untuk dilakukan penelitian karena memiliki dampak langsung kepada masyarakat Indonesia. Namun, masih sedikit penelitian yang menerapkan metode deep learning yaitu Long Short-Term Memory dan CNN untuk analisis sentimen pada artikel finance di Indonesia. Penelitian ini bertujuan untuk melakukan pengklasifikasian judul berita berbahasa Indonesia berdasarkan sentimen positif, negatif dengan menggunakan metode LSTM, LSTM-CNN, CNN-LSTM. Dataset yang digunakan adalah data judul artikel berbahasa Indonesia yang diambil dari situs Detik Finance. Berdasarkan hasil pengujian memperlihatkan bahwa metode LSTM, LSTM-CNN, CNN-LSTM memiliki hasil akurasi sebesar, 62%, 65% dan 74%.

Kata Kunci — LSTM, sentiment analysis, CNN

Abstract

Online media produce a lot of various kinds of news, be it economics, politics, health, sports or science. Among them, economics is one interesting topic to discuss. The economy has a direct impact on citizens, companies, and even traditional markets depending on the economic conditions in a country. The sentiment contained in the news can influence people's views on a matter or government policy. The topic of economics is an interesting topic for research because it has a direct impact on Indonesian society. However, there are still few studies that apply deep learning methods, namely Long Short-Term Memory and CNN for sentiment analysis on finance articles in Indonesia. This study aims to classify Indonesian news headlines based on positive and negative sentiments using the LSTM, LSTM-CNN, CNN-LSTM methods. The dataset used is data on Indonesian language article titles taken from the Detik Finance website. Based on the test results, it shows that the LSTM, LSTM-CNN, CNN-LSTM methods have an accuracy of, 62%, 65% and 74%.

Keywords — LSTM, sentiment analysis, CNN

1. PENDAHULUAN

Berdasarkan laporan terbaru We Are Social, pada tahun 2020 disebutkan bahwa ada 175,4 juta pengguna internet di Indonesia. Dibandingkan tahun sebelumnya, ada kenaikan 17% atau 25 juta pengguna internet di negeri ini. Berdasarkan total populasi Indonesia yang berjumlah 272,1 juta jiwa, maka itu artinya 64% setengah penduduk Indonesia telah merasakan akses ke dunia maya. Salah satu kunci utama kesuksesan penetrasi internet di Indonesia adalah perkembangan infrastruktur yang sudah mencapai daerah pelosok di Indonesia [1]. Dengan adanya pertumbuhan teknologi informasi, terutama media online, banyak mengubah cara manusia untuk melakukan komunikasi dengan sesamanya. Penggunaan media online banyak dimanfaatkan oleh masyarakat untuk mengekspresikan opini, pengalaman maupun hal lain yang menjadi perhatian mereka.

Perkembangan teknologi saat ini tumbuh sangat pesat, hal ini membuat penyebaran informasi semakin mudah dan cepat melalui media online (facebook, twitter), blog, atau situs resmi suatu lembaga [2]. Dengan kemudahan dan kecepatan media online, mampu mengubah cara konsumsi masyarakat terhadap suatu berita. Koran, majalah fisik telah tergantikan oleh media online seperti berita online, weblog.

Di zaman sekarang orang ingin mengkonsumsi sebanyak – banyaknya berita, dari berbagai sumber yang menurut mereka penting, atau yang sesuai dengan minat mereka. Media online telah mengembangkan strategi yang efektif untuk menarik perhatian orang. Media online mengungkapkan pendapat tentang suatu entitas, yang dapat terdiri dari orang, tempat atau bahkan benda ketika melaporkan peristiwa yang terjadi [3]. Untuk alasan ini layanan penilaian emosi interaktif ditawarkan oleh berbagai saluran dari beberapa situs web berita, berita dapat digolongkan menjadi negatif dan positif. Media online banyak menghasilkan berbagai macam berita, baik ekonomi, politik, kesehatan, olahraga atau ilmu pengetahuan. Di antara itu semua, ekonomi adalah salah satu topik menarik untuk dibahas. Ekonomi memiliki dampak langsung kepada warga negara, perusahaan, bahkan pasar tradisional tergantung pada kondisi ekonomi di suatu negara [4].

Dalam penelitian Joshi, (2018) memaparkan bahwa tren saham dapat di prediksi dengan artikel yang berkaitan dengan harga saham, artikel tersebut akan mempengaruhi secara positif ataupun negatif harga saham pasaran [5]. Sentimen yang terkandung dalam berita dapat mempengaruhi pandangan masyarakat terhadap suatu hal atau kebijakan pemerintah. Topik ekonomi adalah bahasan yang menarik untuk dilakukan penelitian karena memiliki dampak langsung kepada masyarakat Indonesia [4]. Oleh sebab itu penelitian ini dikhususkan untuk melakukan analisis sentimen berita ekonomi yang didapat dari di media online berbahasa Indonesia, Detik Finance.

Sentiment Analysis (SA) sering dikenal sebagai opinion mining yang digunakan untuk menganalisis atau mengklasifikasikan pengguna dari kata, kalimat atau dokumen [6]. Beberapa penelitian telah melakukan penelitian mengenai sentiment analisis menggunakan model – model yang berbeda. Penelitiann Pra-pemrosesan Data Twitter Indonesia untuk Pengakuan Emosi menggunakan deep learning model MLP dan LSTM dimana hasil akurasi kedua model tidak jauh berbeda yaitu 71,74% namun MLP menunjukkan proses yang lebih cepat [7]. Kochkina et al., dengan menggunakan pendekatan LSTM dan Branch LSTM untuk menentukan sentimen atau komentar di twitter yang hanya mendapat skor akurasi 78% [8]. Penelitian lain dari Augenstein et al., menggunakan *condition LSTM encoding* untuk mengolah pendapat dari media social [9]. Jelodar & Huang, menggunakan pendekatan LSTM untuk menentukan sentiment publik terhadap COVID-19 menghasilkan akurasi 81,15% dalam penelitiannya [10]. Murty, melakukan penelitian dengan judul “*Text based Sentiment Analysis using LSTM*” mampu mendapatkan hasil akurasi sebesar 85% [11].

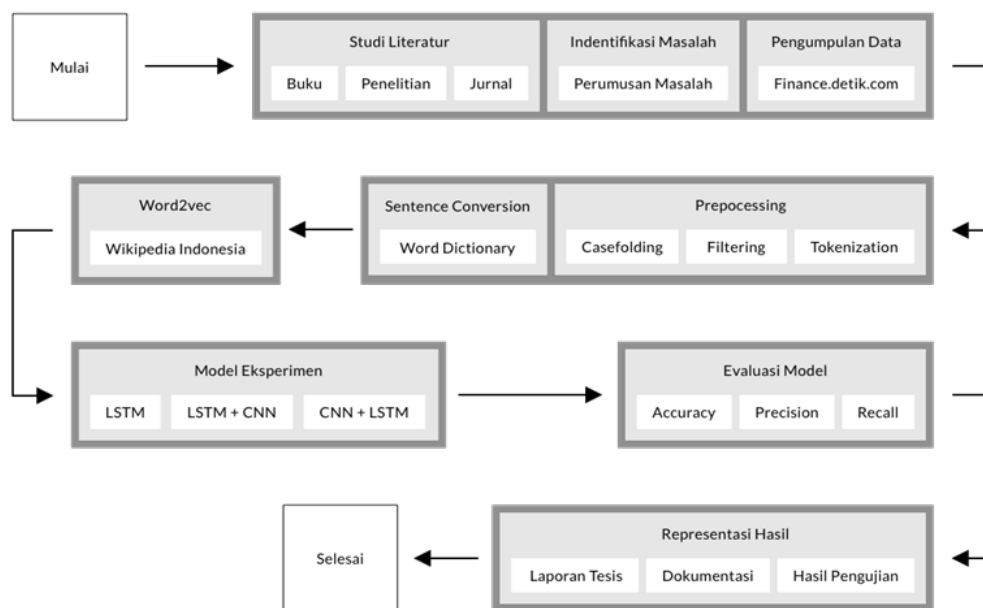
Dari hasil penelitian – penelitian di atas masih didapat perbedaan hasil akurasi untuk menentukan analisis sentimen, hal ini disebabkan oleh perbedaan model dan banyaknya dataset yang digunakan. Oleh sebab itu menarik bagi peneliti melakukan penelitian ulang untuk menentukan analisis sentiment. Penelitian ini terinspirasi dari penelitian Jelodar yang

menggunakan metode LSTM untuk menentukan sentimen, LSTM adalah metode yang lebih baik dibandingkan dengan konvensional, metode LSTM sangat cocok untuk diterapkan pada sentiment analisis [10]. Peneliti akan mengkombinasikan metode LSTM dengan CNN, Dengan adanya kombinasi LSTM dan CNN akan meningkatkan performa analisis sentimen [12]. Penelitian ini juga akan membandingkan hasil dari kombinasi LSTM dan CNN dengan metode LSTM pada media online Detik Finance.

2. METODE PENELITIAN

2.1. Methodology Penelitian

Methodology yang digunakan pada penelitian ini terdiri beberapa tahapan yang selanjutnya dapat dilihat pada Gambar 1 di bawah ini.



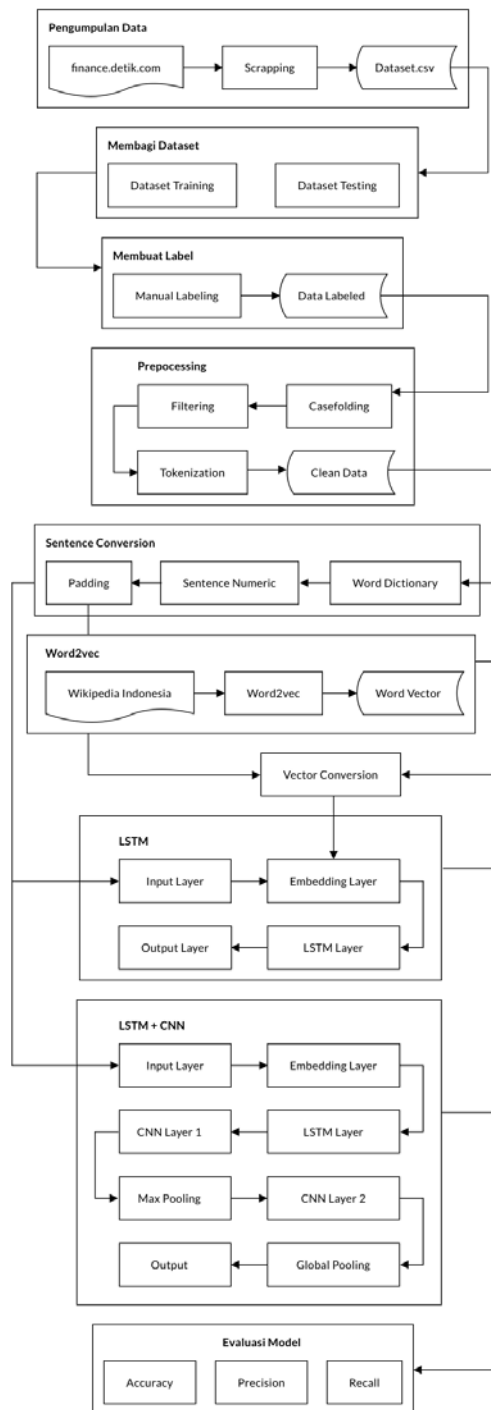
Gambar 1. Alur Penelitian

Penelitian ini dimulai dari tahapan studi literatur. Pada tahap ini peneliti melakukan studi literatur dengan membaca buku, jurnal, makalah dan laporan penelitian yang terkait dengan topik penelitian. Kemudian melakukan pengumpulan data dan informasi seperti melakukan wawancara, dokumentasi dan observasi untuk mengambil kebutuhan data. Selanjutnya dilakukan proses definisi kebutuhan dengan melakukan identifikasi data yang dibutuhkan, melihat prosedur yang sedang berjalan, menganalisis sistem yang sedang berjalan serta membuat hasil evaluasi sistem tersebut. Tahapan kedua adalah preprocessing, banyaknya data ulasan yang didapat banyak yang menggunakan kata yang tidak berstruktur seperti singkatan, emotikon, simbol dan angka hal ini diperlukan teknik preprocessing. Preprocessing berguna untuk mengekstrak informasi dari ulasan, mengubah kata-kata yang tidak terstruktur itu menjadi bentuk standar. Preprocessing sangat penting dan kritis dalam data mining [13]. Setelah dilakukan preprocessing tahapan yang dilakukan adalah sentence conversion. Konversi kalimat dilakukan dalam beberapa langkah: membuat kata-kata kamus, mengonversi kata menjadi angka, dan padding. Proses ini dilakukan untuk digunakan sebagai input ke sistem [14]. Tahapan selanjutnya adalah Word2vec, salah satu metode embedding word yang berguna untuk merepresentasikan kata menjadi sebuah vektor. Pada tahap selanjutnya ini berisikan proses desain sistem dan software seperti pemodelan metode LSTM, LSTM-CNN dan CNN-LSTM. Setelah itu dilakukan evaluasi model, evaluasi model dilakukan perhitungan, tingkat akurasi, presisi, dan *recall*. Confusion Matrix digunakan untuk

mengukur kinerja yang memiliki beberapa parameter. Confusion Matrix juga dikenal sebagai error matrix. Parameter yang digunakan adalah TP (true positive), FN (false negative), TN (true negative), dan FP (false positive) pada data training [15]. Pada tahap akhir ini dilakukan proses dokumentasi dan publikasi thesis berisi hasil penelitian yang telah dilakukan.

2.2. Rancangan Flowchart Pemodelan LSTM-CNN

Dari Gambar 2 dapat dilihat bahwa peneliti akan melakukan uji terhadap 3 kombinasi Algoritma, yang pertama LSTM, kedua LSTM-CNN dan yang ketiga CNN-LSTM.



Gambar 2. Rancangan Flowchart Pemodelan LSTM-CNN

2.3. Pengacuan Pustaka

Berikut ini adalah beberapa teori yang digunakan dalam penelitian ini.

2.3.1. LSTM

LSTM menggunakan tiga gerbang yaitu gerbang input, gerbang forget dan gerbang output. Dalam LSTM satu komponen gate digunakan saat mengontrol informasi yang masuk kedalam memori yang bertugas memecahkan masalah dari gradien kehilangan dan pembagian. Koneksi yang berulang menambah keadaan atau memori ke jaringan dan memungkinkannya untuk memanfaatkan pengamatan yang terurut. Memori internal berarti output network yang bergantung dengan konteks terakhir pada atrian input bukan input yang disajikan sebagai jaringan [9].

2.3.2. Tensorflow

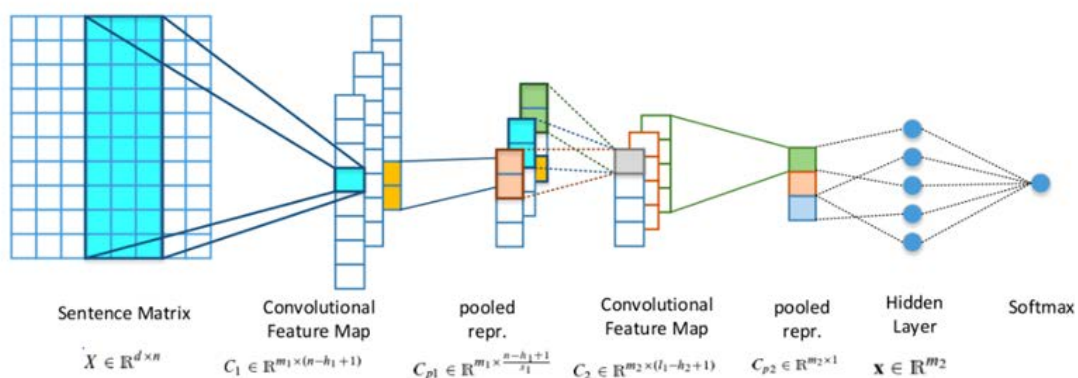
Tensorflow dirilis oleh Google pada November 2015 yang telah digunakan pada sebagian besar produk Google seperti Google Search, deteksi spam, pengenalan suara, Google Assistant, Google Now, dan Google Photos. TensorFlow memiliki kemampuan untuk melakukan perhitungan sub-grafik parsial sehingga memungkinkan pelatihan terdistribusi dengan bantuan mempartisi *neural networks*. Sehingga TensorFlow memungkinkan pemrosesan *parallelism* model dan *parallelism* data. TensorFlow menyediakan beberapa API. Pada level terendah API yaitu TensorFlow Core memberikan kontrol dengan pemrograman yang lengkap.

2.3.3. Word2vec

Word2vec adalah salah satu metode *embedding word* yang berguna untuk merepresentasikan kata menjadi sebuah vektor [14]. Misalnya sebuah kata “Indonesia” di representasikan menjadi sebuah vektor dengan panjang 5 yaitu: [0.2, 0.4, -0.8, 0.9, -0.5]. Vektor tersebut tidak hanya merepresentasikan kata secara sintaktik tapi juga secara semantik atau secara makna.

2.3.4. CNN

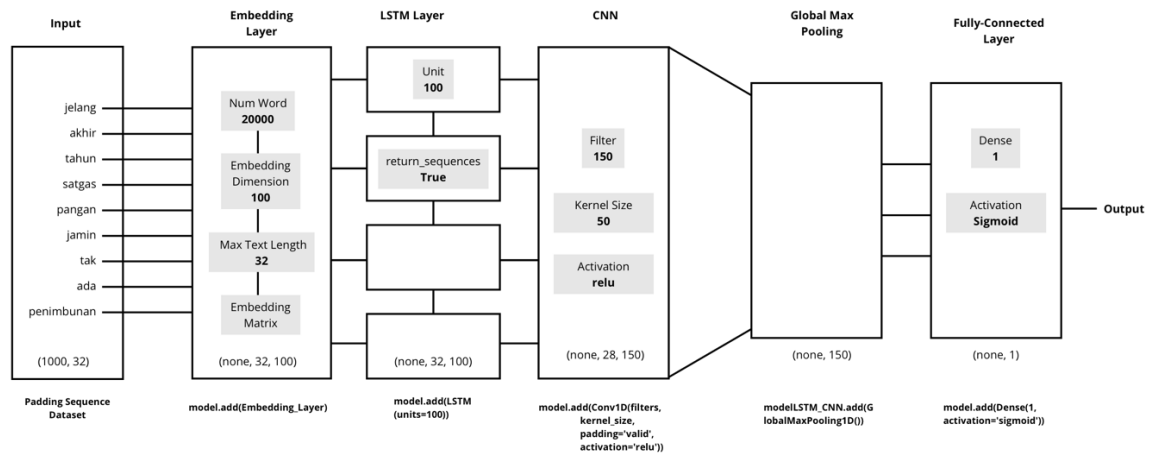
Convolutional Neural Network (CNN) merupakan jaringan saraf tiruan yang memberi umpan balik dimana jaringan saraf mempertahankan struktur hierarki dengan mempelajari representasi fitur internal dan menggeneralisasi fitur-fitur dalam masalah gambar secara umum seperti pengenalan objek dan masalah *computer vision*. Penggunaanya tidak terbatas pada gambar, bisa juga digunakan untuk mencapai hasil dalam masalah pemrosesan bahasa alami dan pengenalan ucapan. Arsitektur dari CNN tergambar pada Gambar 3.



Gambar 3. Arsitektur CNN

2.3.5. LSTM-CNN

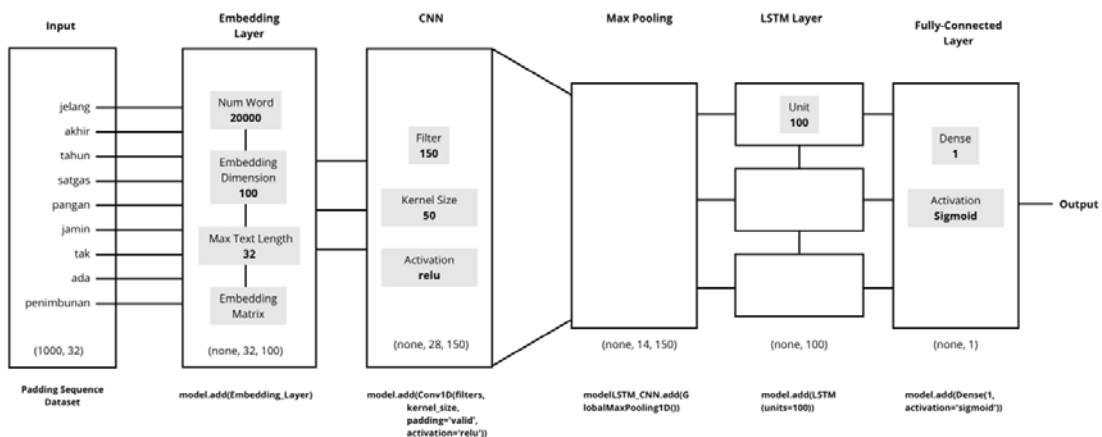
Model LSTM-CNN terdiri dari lapisan LSTM awal yang akan menerima *embeddings* kata untuk setiap token dalam judul sebagai input. Intuisi adalah dimana token outputnya akan menyimpan informasi tidak hanya dari token awal, tetapi juga token sebelumnya. Dengan kata lain, lapisan LSTM menghasilkan pengkodean baru untuk input asli. Output dari layer LSTM kemudian diumpungkan ke dalam lapisan konvolusi yang diharapkan akan mengekstrak fitur lokal. Akhirnya Output lapisan konvolusi akan dikumpulkan ke dimensi yang lebih kecil dan pada akhirnya dikeluarkan sebagai label positif atau negatif [17]. Pada Gambar 4 menunjukkan arsitektur LSTM-CNN



Gambar 4. Arsitektur LSTM-CNN

2.3.6. CNN-LSTM

Model CNN-LSTM terdiri dari lapisan *Convolutional* awal yang akan menerima *embeddings* kata untuk setiap token dalam judul sebagai input. Output yang dihasilkan dari lapisan CNN akan dikumpulkan ke dimensi yang lebih kecil kemudian diumpungkan ke dalam lapisan LSTM yang diharapkan akan mengekstrak fitur lokal. Pada akhirnya dikeluarkan sebagai label positif atau negatif [17][17][17][16][16][16][13]. Pada Gambar 5 menunjukkan asitektur CNN-LSTM.



Gambar 5. Arsitektur CNN-LSTM

2.4. Pengumpulan Data

Dataset diambil dari media sosial Detik Finance pada bulan Desember 2017 sampai bulan Desember 2018 data yang diambil adalah judul dari artikel yang ada dilaman Detik Finance berjumlah 1200 data. Dataset hasil web scraping dari Detik Finance disimpan dengan format CSV. Sugiyono, 2012 dalam (Pradana & Reventiary, 2016) menyatakan bahwa sampel adalah bagian dari jumlah dan karakteristik yang dimiliki oleh populasi tersebut. Karena populasi dari yang data tweet diketahui maka Teknik sampel dalam penelitian ini adalah *probability sampling* dengan menggunakan sampling acak sederhana (*Simple Random Sampling*). Jumlah data yang akan digunakan untuk fase learning ditentukan menggunakan rumus Slovin sedangkan sisanya akan digunakan sebagai data testing. Adapun perhitungan jumlah sampel berdasarkan persamaan Slovin sebagai berikut:

$$n = \frac{N}{1 + N e^2} \quad (1)$$

dimana: N = ukuran populasi

n = ukuran sampel

e = batas toleransi kesalahan

Pembuatan data train dalam menggunakan algoritma *Machine Learning* sangat di perlukan untuk metrain kinerja machine learning. Total data yang tersedia ada 1200 judul berita. Data latih juga dilakukan dengan cara membagi data yang tersedia menggunakan rumus slovin dengan menggunakan tingkat signifikansi sebesar 5%.

$$n = \frac{1200}{1 + 1200 \cdot 0,05^2} = 300$$

Kemudian didapatkan jumlah data sampel yang digunakan untuk data train atau latih sebanyak 300 judul artikel dan kemudian diambil acak. setelah tahapan tersebut dilakukan pelabelan secara manual. Jumlah data set yang digunakan sebanyak 1200 yang kemudian akan dibagi menjadi data training dan data testing. Data training sebesar 900 data dan data testing sebesar 300 data. Setelah dilakukan proses preprocessing data akan dilakukan pelabelan secara manual.

2.5. Pelabelan Data

Jumlah data set yang digunakan sebanyak 1200 yang kemudian akan dibagi menjadi data *training* dan data *testing*. Data *training* sebesar 900 data dan data *testing* sebesar 300 data. Setelah dilakukan proses preprocessing data akan dilakukan pelabelan secara manual. Pelabelan dilakukan dengan memberikan label negatif atau positif pada data, hal ini dilakukan karena suatu dokumen memiliki opini dan sentiment positif atau negatif [10]. Pelabelan akan dilakukan oleh tiga *reviewer*. Teknik yang dilakukan untuk pelabelan diperlukan metode khusus, seperti metode Majority Voting [4]. Dari berbagai metode tersebut metode yang sederhana dan banyak digunakan adalah metode Majority Voting. Metode ini menggunakan konsep pengambilan keputusan hasil vote yang diperoleh dari jumlah terbesar dari masing-masing pilihan vote yang ada. Pelabalean Dasar pemberian pelabelan Jumlah data label dapat dilihat pada Tabel 1 berikut.

Tabel 1. Hasil Pelabelan

| Jenis Data | Sentimen | Jumlah Data |
|---------------|----------|-------------|
| Data Training | Positif | 530 |
| | Negatif | 370 |
| Data Testing | Positif | 222 |
| | Negatif | 78 |

2.6. Preprocessing

Pada tahap ini dataset yang disimpan dalam bentuk CSV dilakukan preprocessing, proses ini dilakukan untuk membersihkan data teks yang tidak terstruktur tahapan yang dilakukan diantaranya adalah sebagai berikut:

2.6.1. Casefolding

Langkah pertama dalam melakukan pembersihan data yaitu dengan melakukan casefolding atau membuat semua huruf menjadi huruf kecil [13].

2.6.2. Filtering

Pada tahap ini dilakukan dengan menghapus karakter khusus pada dataset, adapun karakter khusus yang dihapus adalah tanda baca (poin (.), koma (,), tanda tanya, tanda seru, angka numerik dan karakter lainnya. Dalam proses ini, penyesuaian dilakukan dengan menghapus karakter khusus dalam ulasan tersebut sebagai tanda baca (poin (.), Koma (,), tanda tanya (?), tanda seru (!) dan seterusnya), angka numerik (0-9), dan karakter lainnya (\$,% , * , dll.). Proses ini juga dihapus kata-kata yang tidak cocok dengan hasil penguraian, seperti nama pengguna yang diawali dengan simbol "@", tagar "#", Uniform Resource Locator (URL), dan emotikon. Tanda / simbol atau angka ini dihilangkan karena tidak berpengaruh banyak pada penentuan label [13].

2.6.3. Tokenization

Pada prinsipnya proses ini adalah memisahkan setiap kata yang menyusun suatu dokumen. umumnya setiap kata terpisahkan dengan kata yang lain oleh karakter spasi, sehingga proses tokenisasi mengandalkan karakter spasi pada dokumen untuk melakukan pemisahan kata [13]. Setelah melalui proses tokenisasi maka kalimat tersebut menjadi sekumpulan array yang setiap selnya berisi kata-kata yang ada pada kalimat tersebut.

2.6.4. Sentence Conversion

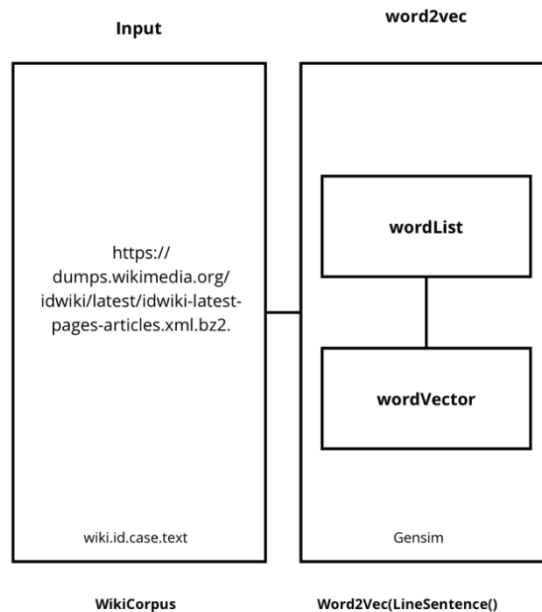
Pada prinsipnya proses ini adalah memisahkan setiap kata yang menyusun suatu dokumen. umumnya setiap kata terpisahkan dengan kata yang lain oleh karakter spasi, sehingga proses tokenisasi mengandalkan karakter spasi pada dokumen untuk melakukan pemisah kata [13]. Setelah melalui proses tokenisasi maka kalimat tersebut menjadi sekumpulan array yang setiap selnya berisi kata-kata yang ada pada kalimat tersebut. Hasil dari sentence conversion dapat dilihat pada Tabel 2

Tabel 2. Hasil Sentence Conversion

| Proses | Hasil |
|--------------------------|---|
| Sentence Word Dictionary | ['dihantam', 'isu', 'boikot', 'saham', 'starbucks', 'sudah', 'turun'] |
| Sentence Numeric | [1166, 493, 1167, 26, 694, 64, 29] |
| Padding Sequences | array([[0, 0, 0, ..., 181, 39, 46], [0, 0, 0, ..., 8, 47, 693], [0, 0, 0, ..., 694, 64, 29], ..., [0, 0, 0, ..., 2577, 1, 188], [0, 0, 0, ..., 2, 59, 243], [0, 0, 0, ..., 6, 86, 1]], dtype=int32) |

2.7. Word2vec

Word2vec adalah salah satu metode embedding word yang berguna untuk merepresentasikan kata menjadi sebuah vektor [14]. Pada tahap pembuatan kamus kata dilakukan dengan menggunakan kumpulan artikel pada Wikipedia Indonesia yang diambil pada tanggal bulan Juni 2020 dengan dari laman Wikipedia <https://dumps.wikimedia.org/idwiki/latest/idwiki-latest-pages-articles.xml.bz2>. Gambar arsitektur Word2vec dapat dilihat pada Gambar 6.



Gambar 6. Arsitektur Word2vec

Tahap pertama yang dilakukan adalah menggabungkan kumpulan artikel pada laman Wikipedia Indonesia kedalam satu file `wiki.id.case.txt`. Dalam pembuatan `word2vec` ini menggunakan Gensim. Gensim sendiri merupakan library pada python yang digunakan untuk *document indexing*. Setelah tahap pertama dilakukan kemudian tahap yang kedua adalah pembuatan kamus kata dan mengubah kamus vektor pada file `wiki.id.case.txt` tahap pertama. Pembuatan kamus kata dan kamus vektor dilakukan dengan algoritma *word embedding* yaitu `word2vec`. Pada `word2vec`, dimensi yang digunakan adalah 100 dimensi. Hasil `word2vec` disimpan dengan variabel `wordList` sebagai kamus kata, sedangkan `wordVectors` sebagai kamus vektor.

2.8. Vector Conversion

Konversi vektor dilakukan dengan menggunakan kamus kata yang telah dibuat dan direpresentasikan dalam model `word2vec` yang sebelumnya telah disiapkan. Model `word2vec` dibuat menggunakan teks-teks Wikipedia bahasa Indonesia. Jika kata itu ditemukan dalam model `word2vec` (memiliki representasi vektor untuk sebuah kata), maka vektor `word2vec` digunakan untuk mewakili kata tersebut. Tetapi jika kata itu tidak ditemukan dalam model, akan digantikan oleh nilai vektor acak dari semua nilai vektor pada model. Nilai vektor pada kamus kata kemudian diperoleh berdasarkan model *word2vec* [10]. Pada tahap selanjutnya hasil dari *vector conversion* dari seluruh kata pada *dictionary* data training akan disimpan sebagai variabel *embedding matrix*. *Embedding matrix* tersebut akan digunakan sebagai *embedding layer* pada seluruh model dalam penelitian ini. *Embedding matrix* memiliki vektor dimensi $n * m$ dimana n adalah jumlah maksimum text yang digunakan yaitu 20.000 kata dan menggunakan 100 dimensi dalam model `word2vec`. Oleh karena itu, input akan bernilai $20.000 \times 100 = 200.000$ dimensi. Salah satu hasil dari word embedding dengan `word2vec` dalam kamus dapat dilihat pada Tabel 3 dibawah ini.

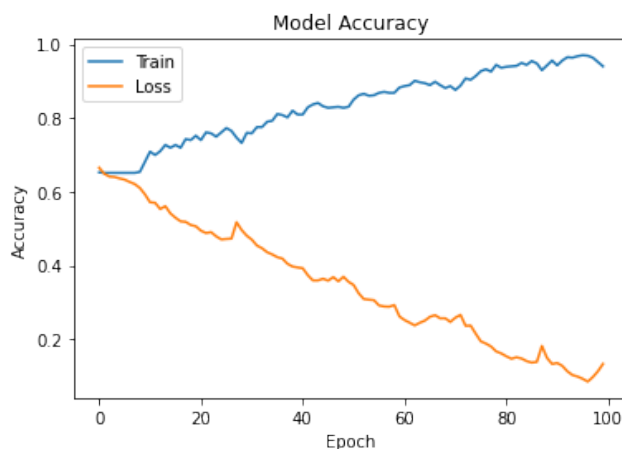
Tabel 3. Vector Conversion

| Input Kata | Tahun |
|--------------------------|--|
| wordList | Tahun |
| Vector Conversion | [0.15011457 0.12223593 -0.05914544 -0.01451576 -0.11412521 0.14369671 0.01565686 -0.15666975 0.09361115 0.13449727 0.09083663 0.15319431 0.06012673 -0.09612084 -0.00835942 -0.03225635 -0.09512474 0.05388279 -0.02717807 -0.00481287 0.0869659 -0.10497787 0.01098405 -0.00188777 -0.08383208 -0.10209516 -0.03121186 0.06353917 -0.2579768 0.07476923 -0.09776208 0.00369547 0.02606086 -0.09923253 0.05399366 -0.12148095 0.05931812 0.01462541 0.01381589 0.0343395 0.1095319 -0.02341822 0.05841604 -0.08655677 0.08428789 -0.07899253 -0.03720102 0.06529604 0.24003538 0.1387591 -0.01932097 0.23544015 -0.07708308 -0.03093985 -0.11219727 -0.16240142 0.0795774 -0.08938915 -0.03696011 -0.09556272 0.03182485 -0.0196524 -0.07325872 -0.07372283 0.05406671 0.06587923 0.0609282 0.06562376 -0.00124368 -0.03911165 0.15168665 0.0544526 -0.10217085 0.03558627 0.05059756 0.02183822 0.01956177 0.13398597 0.13100745 0.2230009 -0.0358769 -0.14602429 -0.10033105 -0.03560073 0.12221534 0.07602764 0.14897329 -0.01624453 -0.13605934 0.09563214 0.0080934 0.21982098 0.00167306 -0.120789 -0.03396938 -0.04474739 -0.07485911 -0.12608509 0.14621805 0.21062644] |

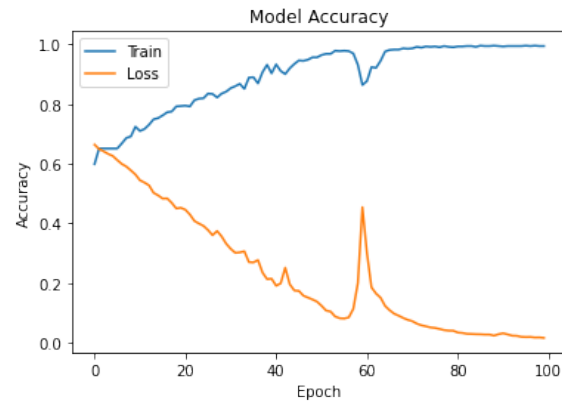
3. HASIL DAN PEMBAHASAN

3.1. Pengujian Model

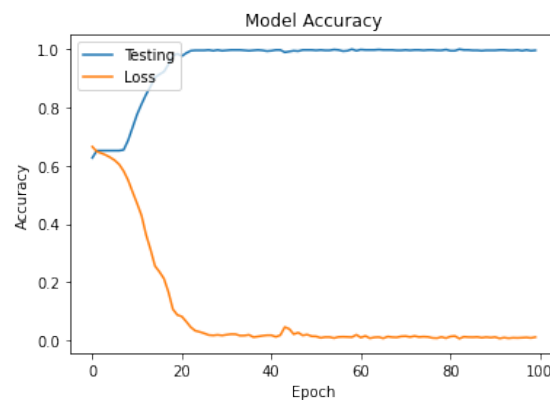
Pada tahapan ini dilakukan training model yang bertujuan untuk menguji model pada data training dengan parameter pada setiap model yang telah dibuat. Berdasarkan pengujian parameter yang telah dicoba, didapatkan model dengan nilai parameter terbaik adalah model CNN-LSTM. Pengujian model menghasilkan nilai akurasi dan loss pada data training dan validasi. Proses training menggunakan jumlah *epoch* sebanyak 100 dan *batch size* sebanyak 128. Dengan parameter yang sudah ditentukan akan dilihat berapa akurasi dari data training dan melihat nilai *loss* terendah. Model akan menyimpan epoch yang optimal pada nilai *loss* yang terendah selama proses epoch berlangsung. Berikut adalah hasil pengujian model dapat dilihat pada gambar dibawah ini. Model LSTM ditunjukkan pada Gambar 7, Model LSTM-CNN pada Gambar 8 dan Model CNN-LSTM pada Gambar 9.



Gambar 7. Model LSTM



Gambar 8. Model LSTM-CNN



Gambar 9. Model CNN-LSTM

Bedasarkan gambar pengujian diatas hasil yang paling baik ditunjukkan pada Gambar 4.6. Model CNN-LSTM menghasilkan nilai akurasi terendah pada data validasi diepoch pertama dengan nilai sebesar 0.6261 dan nilai loss tertinggi pada data training menghasilkan 0.6644. Pada epoch ke 20 model CNN-LSTM menghasilkan akurasi yang baik dibanding dengan model yang lainnya. Perbandingan terlihat bahwa model sudah cukup baik dengan menunjukkan akurasi data training dengan validasi tidak berbeda jauh sehingga tidak mengalami *over fitting*. Selanjutnya model akan diuji kepercayaannya pada tahap pengujian dengan data testing.

3.2. Pengujian Data Testing

Tahap ini merupakan tahap pengujian model dengan data testing. Data yang akan diuji adalah berjumlah 300 judul artikel dengan jumlah label di kelas negatif 78 judul artikel dan 222 judul pada kelas positif. Selanjutnya setelah model menghasilkan prediksinya di masing-masing kelas dilakukan perhitungan tingkat kepercayaan model dengan melihat akurasi, presisi dan recall. Tujuannya untuk mengetahui seberapa dapat dipercaya model dalam memprediksi kelas. Hasil pengujian data testing ditunjukkan dengan confusion matrix seperti table dibawah ini. Tabel 4 – Tabel 6 menunjukkan hasil Confusion Matrix masing – masing model.

Tabel 4. Hasil Confusion Matrix LSTM

| Aktual | Prediksi | |
|---------|----------|---------|
| | Negatif | Positif |
| Negatif | 33 | 102 |
| Positif | 45 | 120 |

Tabel 5. Hasil Confusion Matrix LSTM-CNN

| Aktual | Prediksi | |
|---------|----------|---------|
| | Negatif | Positif |
| Negatif | 25 | 91 |
| Positif | 53 | 131 |

Tabel 6. Hasil Confusion Matrix CNN-LSTM

| Aktual | Prediksi | |
|---------|----------|---------|
| | Negatif | Positif |
| Negatif | 32 | 79 |
| Positif | 46 | 143 |

Bedasarkan model yang telah dibuat dimana data test yang berlabel negatif benar diprediksi oleh model LSTM sebesar 33, sedangkan pada model LSTM-CNN sebesar 25 dan pada model CNN-LSTM sebesar 32 dan salah memprediksi pada model LSTM sebesar 45, sedangkan pada model LSTM-CNN sebesar 53 dan pada model CNN-LSTM sebesar 46. Begitupula dengan ulasan yang berlabel positif dapat benar diprediksi oleh model LSTM sebesar 120, sedangkan pada model LSTM-CNN sebesar 131 dan pada model CNN-LSTM sebesar 143 dan salah memprediksi pada model LSTM sebesar 52. Sedangkan pada model LSTM-CNN sebesar 41 dan pada model CNN-LSTM sebesar 29. Dari *Confusion Matrix* pada tabel diatas model yang menghasilkan nilai prediksi positif benar paling banyak adalah model CNN-LSTM sebesar 90 sedangkan model yang menghasilkan nilai negatif benar paling banyak adalah model LSTM sebesar 33. Namun apabila jumlah model yang dapat memprediksi positif dan negative benar dijumlahkan maka diperoleh urutan model yang pertama adalah CNN-LSTM sebesar 175 data diprediksi benar positif dan negatif, model yang kedua adalah LSTM-CNN sebesar 156 data diprediksi benar positif dan negative dan model yang terakhir adalah LSTM sebesar 153 data diprediksi benar positif dan negatif.

Selanjutnya setelah model menghasilkan prediksinya di masing-masing kelas dilakukan perhitungan tingkat kepercayaan model dengan melihat akurasi, presisi dan recall. Tujuannya untuk mengetahui seberapa besar persentase akurasi untuk dapat dipercaya sebagai model dalam memprediksi kelas. Hasil Perbandingan Hasil Model dapat dilihat pada Tabel 7. Dari penelitian ini didapati bahwa model CNN-LSTM lebih unggul dalam penilaian akurasi, presisi, dan recall.

Tabel 7. Hasil Perbandingan Hasil Model

| Evaluasi | LSTM | LSTM-CNN | CNN-LSTM |
|----------|------|----------|----------|
| Akurasi | 51% | 53% | 62% |
| Presisi | 60% | 60% | 66% |
| Recall | 57% | 66% | 76% |

4. KESIMPULAN

Setelah melalui tahap pengujian tingkat peforma dengan parameter akurasi, presisi, recall pada model LSTM, LSTM-CNN dan CNN-LSTM dalam menentukan sentimen analisis pada judul artikel diDetik Finance, maka dapat diambil kesimpulan antara lain:

1. Nilai akurasi pada model CNN-LSTM lebih tinggi dibandingkan dengan model LSTM dan model LSTM-CNN yaitu sebesar 62%.

2. Nilai recall model CNN-LSTM lebih tinggi dibandingkan dengan model LSTM dan LSTM-CNN yaitu sebesar 76%.
3. Nilai presisi pada model CNN-LSTM yang lebih tinggi dibandingkan dengan model LSTM dan model LSTM-CNN yaitu sebesar 66%.
4. Model CNN-LSTM merupakan model terbaik dibandingkan dengan model LSTM dan LSTM-CNN pada penelitian ini.

5. SARAN

Ada beberapa kelemahan dalam penelitian ini. Saran untuk penelitian selanjutnya adalah:

1. Pada penelitian selanjutnya dapat menambahkan jumlah dataset yang digunakan dan melakukan perbandingan akurasi berdasarkan jumlah dataset yang lebih banyak.
2. Pada penelitian ini hanya menggunakan binari klasifikasi pada pelabelan yang dibuat, selanjutnya dapat menggunakan tiga klasifikasi pada pelabelan.
3. Pada penelitian selanjutnya dapat mengkombinasikan model-model algoritma yang lain untuk mendapatkan hasil akurasi yang lebih baik.

DAFTAR PUSTAKA

- [1] Detiknet.com, 2020, Riset: Ada 175,2 Juta Pengguna Internet di Indonesia, <https://inet.detik.com/cyberlife/d-4907674/riset-ada-1752-juta-pengguna-internet-di-indonesia>.
 - [2] Meng, X., Liu, M., Wu, Q., 2020, Prediction of Rice Yield via Stacked LSTM, *International Journal of Agricultural and Environmental Information Systems (IJAEIS)*, No. 1, Vol. 11, Hal. 86–95
 - [3] Reis, J., Benevenuto, F., Olmo, P., Prates, R., Kwak, H., An, J., 2015, Breaking the news: First impressions matter on online news, *The 9th International Conference on Web and Social Media*, oxford, 26-29 Mei
 - [4] Zamahsyari., Nurwidyanoro, A., 2017, Sentiment analysis of economic news in Bahasa Indonesia using majority vote classifier, *2016 International Conference on Data and Software Engineering (ICoDSE 2016)*, Denpasar, 26 - 27 Oktober
 - [5] Joshi, K., N Bharathi, H., Rao, J., 2016, Stock Trend Prediction Using News Sentiment Analysis, *International Journal of Computer Science & Information Technology*, No. 3, Vol. 8, Hal. 1-8
 - [6] Zubiaga, A., Kochkina, E., Liakata, M., Procter, R., Lukasik, M., 2016, Structure of Social Media Conversations, *arXiv preprint*, arXiv:1609.09028
 - [7] Nugraheni, E., 2020, Indonesian Twitter Data Pre-processing for the Emotion Recognition, *2019 International Seminar on Research of Information Technology and Intelligent Systems (ISRITI)*, Yogyakarta, 5 -6 Desember
 - [8] Kochkina, E., Liakata, M., Augenstein, I., 2017, Turing at SemEval-2017 Task 8: Sequential Approach to Rumour Stance Classification with Branch-LSTM, *the 11th International Workshop on Semantic Evaluation (SemEval-2017)*, Vancouver, 3-4 Agustus
 - [9] Augenstein, I., Rockt, T., Vlachos, A., Bontcheva, K., 2016, Stance Detection with Bidirectional Conditional Encoding, *arXiv preprint*, arXiv:1606.05464
-

-
- [10] Jelodar, H., Wang, Y., Orji, R., Huang, H., 2020, Deep Sentiment Classification and Topic Discovery on Novel Coronavirus or COVID-19 Online Discussions: NLP Using LSTM Recurrent Neural Network Approach, IEEE Journal of Biomedical and Health Informatics, doi: 10.1109/JBHI.2020.3001216
- [11] Murthy, G. S. N., Allu, S. R., Andhavarapu, B., Bagadi, M., Belusont, M., 2020, Text based Sentiment Analysis using LSTM, IJERT, No. 5, Vol. 9, Hal. 299–303
- [12] Jain, D., Kumar, A., Garg, G., 2020, Sarcasm detection in mash-up language using soft-attention based bi-directional LSTM and feature-rich CNN, Applied Soft Computing Journal, Vol. 91, Hal. 106198, 2020, doi: 10.1016/j.asoc.2020.106198.
- [13] Kane, S. N., Mishra, A., Dutta, A. K., 2016, Preface: International Conference on Recent Trends in Physics (ICRTP 2016), International Conference on Recent Trends in Physics 2016 (ICRTP2016), Inore - India, 13-14 Februari
- [14] Nurrohmat, M. A., A. Sn, 2019, Sentiment Analysis of Novel Review Using Long Short-Term Memory Method, IJCCS (Indonesian Journal of Computing and Cybernetics Systems, No. 3, Vol. 13, Hal. 209–218
- [15] Kurniasari, L., Setyanto, A., 2020, Sentiment Analysis using Recurrent Neural Network, Journal of Physics: Conference Series, Vol. 1471, Bukit Tinggi, 18 Oktober, doi: 10.1088/1742-6596/1471/1/012018.
- [16] Manaswi, N. K., 2018, Deep Learning with Applications Using Python, Apress Media, New York
- [17] Sosa, P. M., 2017, Twitter Sentiment Analysis using combined LSTM-CNN Models, Academia.edu, Hal. 1–9
-