

Klasifikasi Nama Paket Pengadaan Menggunakan Long Short-Term Memory (LSTM) Pada Data Pengadaan

Fathorazi Nur Fajri^{1,*}, Syaiful²

¹Fakultas Teknik, Sistem Informasi, Universitas Nurul Jadid, Probolinggo, Indonesia

²Fakultas Teknik, Informatika, Universitas Nurul Jadid, Probolinggo, Indonesia

Email: ^{1,*}fathorazi@unuja.ac.id, ²syaiful@unuja.ac.id

Email Penulis Korespondensi: fathorazi@unuja.ac.id

Submitted: 06/12/2022; Accepted: 28/12/2022; Published: 30/12/2022

Abstrak—Setiap tahun pemerintah selalu mengadakan pengadaan barang dan jasa (tender) yang diinformasikan melalui Layanan Pengadaan Secara Elektronik (LPSE) atau Sistem Informasi Rencana Umum Pengadaan (SIRUP). Proses pemilihan jenis pengadaan masih bersifat manual yaitu dengan memilih kategori paket sehingga memungkinkan untuk terjadi kesalahan seperti jenis pengadaan jasa masuk kedalam kategori jenis pengadaan barang atau sebaliknya. Oleh karena itu pada penelitian ini mengusulkan untuk menggunakan metode Natural Language Processing (NLP) yang dapat mengklasifikasikan paket tersebut berdasarkan kategori yang ada. Adapun metode yang digunakan adalah Long Short-Term Memory (LSTM) dengan melakukan perbandingan terhadap metode klasifikasi yang ada seperti Naïve Bayes, Logistic Regression, Decision Tree, XG Boost, Gradient Boost, Random Forest dan Support Vector Machine. Hasil yang diperoleh metode LSTM mempunyai akurasi lebih tinggi dari metode lainnya yaitu dengan akurasi 90.25%. Dengan konfigurasi parameter 100 unit di layer LSTM, epoch 10, batch size 64 dan validation step 5.

Kata Kunci: Pengadaan; Klasifikasi; NLP; LSTM

Abstract—Every year the government always holds procurement of goods and services (tenders) which are informed through the Electronic Procurement Service (LPSE) or the General Procurement Plan Information System (SIRUP). The process of selecting the type of procurement is still manual, namely by selecting the package category so that it is possible for mistakes to occur such as the type of service procurement into the category of goods procurement type or vice versa. Therefore, this research proposes to use the Natural Language Processing (NLP) method that can classify these packages based on existing categories. The method used is Long Short-Term Memory (LSTM) by comparing existing classification methods such as naïve bayes, logistic regression, decision tree, XG Boost, Gradient Boost, Random Forest and Support Vector Machine. The results obtained by the LSTM method have a higher accuracy than other methods, with an accuracy of 90.25%. With a parameter configuration of 100 units in the LSTM layer, epoch 10, batch size 64 and validation step 5.

Keywords: Procurement; Classification; NLP; LSTM

1. PENDAHULUAN

Setiap tahun pemerintah melakukan proses lelang melalui Lembaga Kebijakan Pengadaan Barang/Jasa Pemerintah (LKPP) untuk proses pengadaan barang/jasa. Saat ini proses lelang dilakukan menggunakan layanan pengadaan sistem elektronik sehingga lebih transparan dan semua orang mendapatkan informasi secara cepat. Layanan Pengadaan Secara Elektronik (LPSE) telah memudahkan sistem pengadaan yang dulunya dilakukan secara manual beralih menjadi sistem elektronik [1]. Selain itu terdapat Sistem Informasi Rencana Umum Pengadaan (SIRUP) yang berfungsi memudahkan pihak Pengguna Anggaran (PA) atau Kuasa Pengguna Anggaran (KPA) mengumumkan Rencana Umum Pengadaan (RUP) [2].

Saat ini proses pemilihan jenis pengadaan masih bersifat manual yaitu dengan memilih kategori paket sehingga memungkinkan untuk terjadi kesalahan seperti jenis pengadaan jasa masuk kedalam kategori jenis pengadaan barang atau sebaliknya. Selain itu pada website Layanan Pengadaan Secara Elektronik (LPSE) data yang disajikan berupa tabel tanpa adanya kategori spesifik terkait bidang project atau tender yang dikerjakan. Dengan adanya permasalahan tersebut maka penelitian ini mengusulkan proses klasifikasi dengan menggunakan Machine Learning Natural Language Processing (NLP). Serta melakukan komparasi metode klasifikasi terbaik untuk studi kasus jenis pengadaan pada data LPSE. Dimana kedepannya dapat mengurangi kesalahan dalam proses input jenis nama paket pengadaan. Hal ini jenis nama paket menjadi acuan dalam pengelompokan penggunaan dana anggaran.

Natural language processing (NLP) menjadi bidang menarik untuk dilakukan penelitian khususnya Klasifikasi teks (*text classification*) [3]. Penerapan klasifikasi menggunakan *Natural Language Processing* (NLP) semakin giat untuk dikembangkan karena mampu membuat sistem bekerja lebih mandiri. Melihat dari hal tersebut, penerapan NLP yang dirasa lebih efektif daripada menggunakan metode tradisional. Terdapat beberapa macam metode untuk *Natural Language Processing* (NLP) seperti *Convolutional Neural Network* (CNN) untuk mengklasifikasikan berita Indonesia [4], Deep learning dengan melakukan pendekatan *Recurrent Neural Network* (RNN) dan *Long Short-Term Memory* (LSTM) pada klasifikasi berita. Metode ini termasuk dalam kategori supervised learning yang berfokus pada pengenalan data dari label khusus yang telah diberikan sebelumnya. Penerapannya sendiri telah banyak diteliti seperti untuk; klasifikasi kategori berita [5], klasifikasi teks komentar pada media sosial [6], dan sentimen analysis ujaran kebencian [7].

Adapun metode yang digunakan pada penelitian ini yaitu metode *Long Short-Term Memory* (LSTM). Dimana kelebihan dari LSTM yaitu mampu melakukan klasifikasi teks lebih baik karena dapat memproses interaksi antar teks

yang sebenarnya. Selain itu proses pada LSTM dapat menentukan nilai mana yang akan dipilih sebagai keluaran sehingga menjadi masukan yang relevan [8].

Terdapat beberapa penelitian yang berkaitan dengan *Natural Language Processing* (NLP) yang telah dilakukan. Pada penelitian [9] membahas tentang klasifikasi teks Berita Bahasa Indonesia menggunakan metode CNN dan LSTM (C-LSTM). Objek pada penelitian ini yaitu teks berita bahasa Indonesia dengan tujuan mengklasifikasi teks berita bahasa Indonesia menjadi 3 kategori yaitu “Internasional”, “Nasional”, dan “Regional”. Adapun tahapan text processing pada penelitian ini yaitu *topword removal*, *special character removal*, *case folding*, *stemming*, dan data *splitting*. Selanjutnya dataset dijadikan sebagai inputan untuk CNN, LSTM dan C-LSTM. Hasil pada penelitian ini yaitu Metode C-LSTM terlihat memiliki kinerja yang lebih baik dari pada kedua metode pendahulunya (CNN dan LSTM). Hal ini dapat dilihat dari nilai *F1-score* yang melebihi kedua metode lainnya. C-LSTM dapat memperoleh nilai *F1-score* sebesar 0,9327 (93,27%) pada dataset berita yang dipilih lebih baik 2,4% dari LSTM dan lebih baik 3,42% dari CNN.

Peneliti [10] melakukan penelitian pada objek hate speech atau ujaran kebencian. Pada penelitian ini menggunakan metode Long Short-Term Memory (LSTM) akan diimplementasikan untuk mendeteksi ujaran kebencian (hate speech) berkaitan dengan Pemilihan Presiden (Pilpres) 2019. Tahapan pengerjaan dalam penelitian ini dimulai dari studi kepustakaan, pengumpulan data, pra-pemrosesan data, pembuatan model word2vec, perancangan model pendeteksi ujaran kebencian, pelatihan model, dan pengujian model. Adapun dataset yang digunakan ialah 950 kalimat data training dan 190 kalimat data testing. Hasil pada penelitian ini 70.21% benar dideteksi sebagai ujaran kebencian, sedangkan sisanya 29.79% salah dideteksi sebagai bukan ujaran kebencian.

Penelitian selanjutnya yang berkaitan dengan *Natural Language Processing* yaitu pada penelitian Dedi Tri Hermanto [11] yang membahas tentang Algoritma LSTM-CNN untuk Sentimen Klasifikasi dengan Word2vec pada *Media Online*. 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%.

Adapun perbedaan penelitian ini dengan penelitian sebelumnya yaitu 1) Metode yang digunakan *Long Short-Term Memory* (LSTM) dikarenakan metode tersebut dinilai mampu melakukan klasifikasi teks lebih baik, 2) melakukan komparasi atau perbandingan metode klasifikasi untuk memberikan perbandingan hasil antara beberapa metode, 3) menggunakan data yang banyak dengan jumlah data 40000 ribu data text nama paket pengadaan yang didapatkan dari website Sistem Informasi Rencana Umum Pengadaan (SIRUP) untuk dijadikan sebagai bahan data training dan validasi.

Pada prinsipnya, tujuan penelitian adalah menghasilkan model klasifikasi yang akurat dengan algoritma Long Short-Term Memory (LSTM) dengan berbagai perbandingan metode klasifikasi *Naïve Bayes* [12], *Logistic Regression* [13], *Decision Tree* [14], *XG Boost*, *Gradient Boost*, *Random Forest* [15] dan *Support Vector Machine* [16].

2. METODOLOGI PENELITIAN

2.1 Tahapan Penelitian

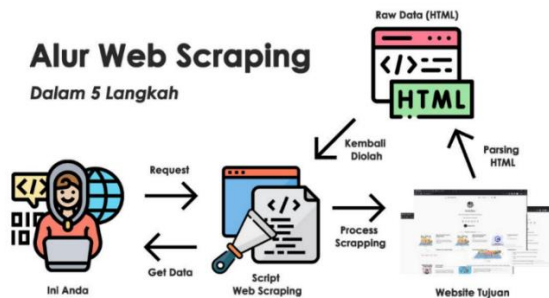


Gambar 1. Tahapan Penelitian

Tahapan penelitian seperti pada gambar 1 dimulai dari proses studi literatur berdasarkan buku, penelitian dan jurnal yang berkaitan dengan topik *text classification*, NLP dan LSTM. Kemudian merumuskan permasalahan yang berkaitan dengan *text classification* yaitu mengklasifikasi jenis pengadaan dengan menggunakan berdasarkan data dari Sistem Informasi Rencana Umum Pengadaan (SIRUP). Selain itu pada penelitian ini juga melakukan perbandingan metode untuk mengetahui metode yang efisien untuk melakukan *text classification*. Proses pengumpulan data dari website sirup dilakukan dengan teknik web scraping. Pada proses *text classification* diawali dengan melakukan proses filtering dan tokenizing [17]. Selanjutnya melakukan model eksperimen pada beberapa metode klasifikasi yaitu *naïve bayes*, *logistic regression*, *decision tree*, *XG Boost*, *Gradient Boost*, *Random Forest* dan *Support Vector Machine*. Evaluasi model dilakukan dengan menggunakan *Confusion Matrix* yaitu *accuracy*, *precision*, dan *recall*. Pada tahapan akhir yaitu melampirkan dokumentasi dari hasil komparasi dari metode klasifikasi dan LSTM.

2.2 Web Scraping

Metode pengumpulan data pada penelitian ini ialah menggunakan metode scraping pada website Sistem Informasi Rencana Umum Pengadaan (SIRUP). Metode scraping ialah teknik untuk mengekstrasi data dari halaman website. Terdapat 3 Metode yang bisanya digunakan untuk scraping website yaitu HTML Parsing, *Regular Expressions* dan *Document Object Model (DOM) parsing* [18]. Pada penelitian ini menggunakan metode scraping *DOM parsing* dengan tool *Web scraper*. Adapun tahapan seperti pada gambar 2 dan *graph tool web scraper* seperti pada gambar 3. Adapun hasil *web scraping* disimpan dalam bentuk file csv.



Gambar 2. Alur Web Scraping



Gambar 3. Graph Web Scraper Element SIRUP

2.3 Filtering

Tahapan *Filtering* merupakan proses dimana dilakukan penghapusan karakter dalam suku kata yang dianggap tidak digunakan atau menjadi karakter yang mengaburkan substansi dari kata tersebut. Adapun karakter yang dihapus yaitu menghapus angka, link, simbol seperti `!@#%&^&*` dst, paragraf, tanda baca dan x (kali). Dimana x (kali) biasanya digunakan pada pengadaan jasa per orang per hari dsb seperti pada gambar 4.

	jenisPengadaan	paket	text_clean
0	Barang	Belanja Modal Mebel	Belanja Modal Mebel
1	Barang	Belanja Alat/Bahan untuk Kegiatan Kantor-Benda...	Belanja AlatBahan untuk Kegiatan KantorBenda Pos
2	Barang	Belanja Jasa Tenaga Penanganan Prasarana dan S...	Belanja Jasa Tenaga Penanganan Prasarana dan S...
3	Barang	Belanja Alat/Bahan untuk Kegiatan Kantor-Alat ...	Belanja AlatBahan untuk Kegiatan KantorAlat Tu...
4	Barang	Belanja Bahan-Bahan Lainnya	Belanja BahanBahan Lainnya

Gambar 4. Hasil *Filtering* kolom *text_cleaning*

2.4 Tokenizing

Tokenizing yaitu proses pemisahan kata pada suatu kalimat yang bertujuan agar kata tersebut membentuk sebuah array sehingga memudahkan untuk proses analisa seperti pada gambar 5. Pada proses tokenizing menggunakan fungsi *countvectorizer*. *Countvectorize* merupakan fungsi untuk mengetahui frekuensi kata yang muncul pada sebuah data.

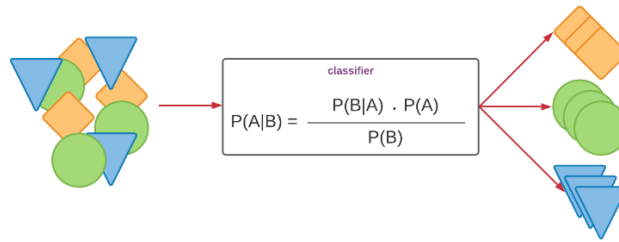
```
[ 'kompascom', 'barcelona', 'baru', 'saja', 'kalah', 'telak', 'dari', 'bayern', 'muenchen', 'pada', 'laga', 'semi', 'final', 'liga', 'champions', 'kemarin', 'pakar', 'sepak', 'bola', 'spanyol', 'bbc', 'guillem', 'balague', 'mengutarakan', 'bahwa', 'kekalahan', 'historis', 'barcelona', 'dari', 'fc', 'bayern', 'muenchen', 'sebagai', 'buah', 'dari', 'kebusukan', 'klub', 'yang', 'berlangsung', 'selama', 'bertahun-tahun' ]
```

Gambar 5. Hasil Tokenizing

2.5 Naïve Bayes

Metode *Naïve bayes* merupakan metode klasifikasi yang proses klasifikasi didasarkan pada probabilitas seperti gambar 6. Dimana proses pengklasifikasiannya berdasarkan teorema bayes. Keunggulan metode ialah mudah diterapkan pada

data biner atau multiclass. Akan tetapi kekurangan pada metode naïve bayes ialah jika perhitungan probabilitasnya menghasilkan 0 maka akan sulit untuk bisa diklasifikasikan.

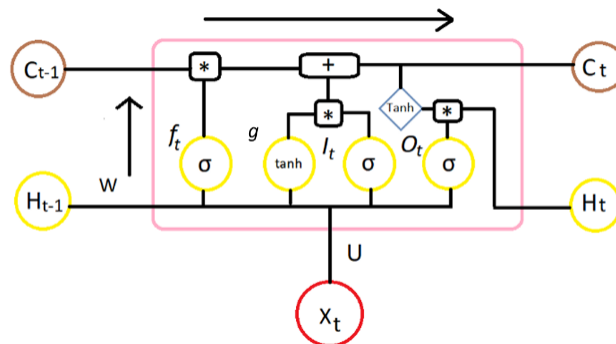


Gambar 6. Arsitektur Naïve Bayes

2.6 Long Short Term Memory

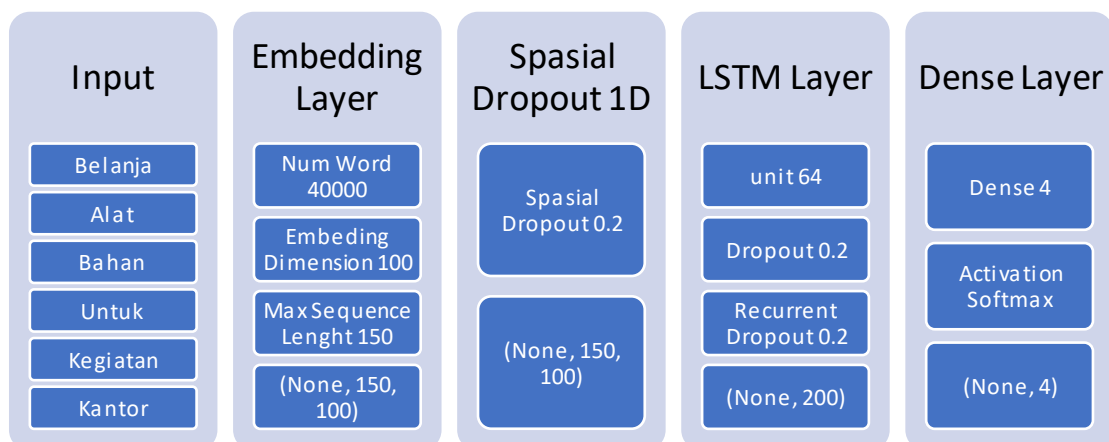
Long Short-Term Memory (LSTM) yaitu jenis model dari Recurrent Neural Network (RNN). Adanya LSTM karena dapat dapat mengingat informasi jangka panjang (*long term dependency*). Prinsip kerja LSTM yaitu menggantikan *node hidden layer* pada RNN dengan sel LSTM yang bertugas untuk menyimpan informasi sebelumnya. Dalam LSTM terdapat tiga gerbang yang mengendalikan penggunaan dan memperbarui informasi terks terdahulu yaitu *input gate*, *forget gate*, dan *ouput gate*. Adapun komponen pada LSTM seperti pada gambar 7 yaitu:

- f* = Forget Gate (NN dengan fungsi sigmoid).
- g* = Candidate Layer (NN dengan fungsi tanh).
- I* = Input Gate (NN dengan fungsi sigmoid).
- O* = Output Gate (NN dengan fungsi sigmoid).
- H* = Hidden State (vektor).
- C* = Memory State (vektor).



Gambar 7. Diagram Long Short-Term Memory (LSTM)

Sel memori dan tiga gerbang dirancang untuk dapat membaca, menyimpan, dan memperbarui informasi terdahulu. Adapun arsitektur LSTM pada penelitian ini seperti pada gambar 7.



Gambar 8. Arsitektur Long Short-Term Memory (LSTM)

2.7 Pengukuran Model

Model yang telah dibuat dilakukan proses pengukuran dengan melihat accuracy, precision dan recall pada model tersebut. Selain itu dilakukan pengukuran menggunakan confusion matrix. Confusion Matrix adalah suatu pengukuran

performa untuk masalah klasifikasi machine learning [19], yaitu sebuah tabel dengan 4 kombinasi yang berbeda dari hasil nilai prediksi dan nilai aktual.

Tabel 1. Confusion Matrix

Predicted Values	Actual Values	
	Positif	Negative
Positif	TP	FP
Negatif	FN	TN

Adapun rumus untuk menghitung recall, precision, dan accuracy ialah sebagai berikut:

$$Accuracy = \frac{TP+TN}{TP+FP+TN+FN} \tag{1}$$

$$Precision = \frac{TP}{TP+FP} \tag{2}$$

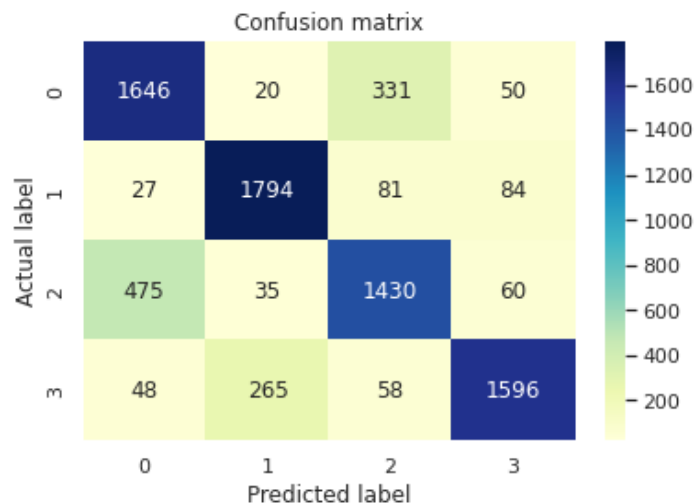
$$Recall = \frac{TP}{TP+FN} \tag{3}$$

3. HASIL DAN PEMBAHASAN

Berdasarkan dari hasil pengumpulan data dengan menggunakan teknik web scraping maka didapat data sebanyak 40000 data pengadaan dalam bentuk 4 kategori yaitu Barang, Jasa Konsultan, Jasa Lainnya, Pekerjaan Konstruksi. Dataset tersebut dibagi dalam bentuk data training dan data testing dengan perbandingan data training sebanyak 80% dan data testing sebanyak 20%. Data yang digunakan ialah data yang telah di tokenizing terlebih dahulu.

3.1 Hasil Eksperimen Model Naïve Bayes

Pada eksperimen model *naïve bayes* menggunakan metode probabilitas *multinomial* menghasilkan *accuracy* 80.825%. Dimana pada masing-masing class nya untuk *class* barang menghasilkan nilai *precision* 75%, *recall* 80%. Untuk class jasa konstruksi menghasilkan nilai *precision* 85%, *recall* 90%, untuk class jasa lainnya menghasilkan nilai *precision* 75%, *recall* 71% dan untuk *class* pekerjaan konstruksi menghasilkan nilai *precision* 89%, *recall* 81%. Dengan *confusion matrix* seperti pada gambar 9.



Gambar 9. Confusion Matrix Naïve Bayes

3.2 Hasil Eksperimen Model Long Short-Term Memory

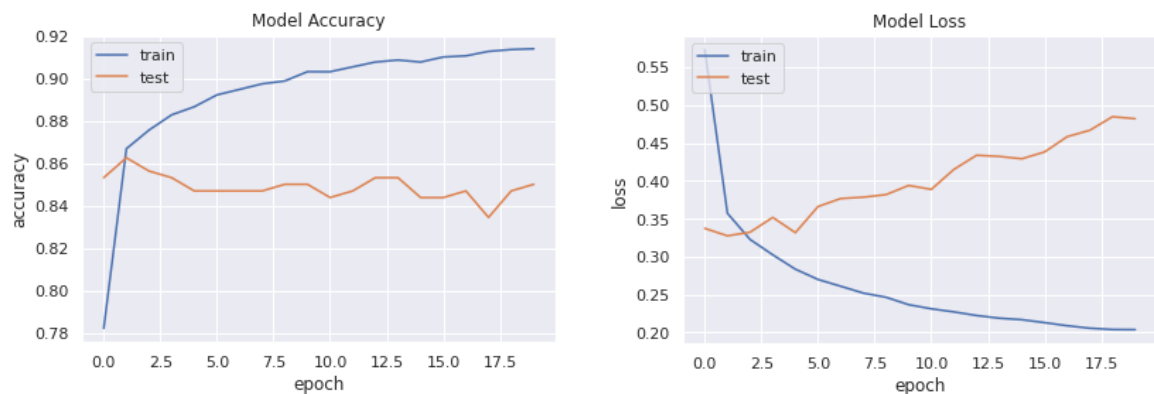
Pada eksperimen model *long short-term memory* terdapat beberapa parameter yang di *tunning* seperti parameter unit pada layer LSTM dan *epoch*. Adapun unit pada layer LSTM yang kita uji coba yaitu 32, 64, dan 100. Kemudian untuk *epoch* nya ialah 10 *epoch* dan 20 *epoch*. Hal ini untuk mengetahui model dengan konfigurasi mana yang terbaik. Pada percobaan 32 unit pada layer LSTM, epoch 20, batch size 64 dan *validation step* 5 maka menghasilkan nilai *accuracy* 91.39% dan *loss* 20.39% untuk *data training* dan untuk *data testing* menghasilkan *accuracy* 85% dan *loss* 48.22% seperti pada gambar 10. untuk *graph* per *epoch* nya seperti pada gambar 11 dimana pada gambar grafik tersebut pada model *loss* menunjukkan adanya peningkatan *loss* pada data testing di setiap peningkatan *epoch* yang artinya *overfitting*.



```

Epoch 1/20
500/500 [=====] - 97s 190ms/step - loss: 0.5732 - accuracy: 0.7821 - val_loss: 0.3375 - val_accuracy: 0.8531
Epoch 2/20
500/500 [=====] - 93s 187ms/step - loss: 0.3574 - accuracy: 0.8668 - val_loss: 0.3276 - val_accuracy: 0.8625
Epoch 3/20
500/500 [=====] - 94s 188ms/step - loss: 0.3228 - accuracy: 0.8756 - val_loss: 0.3324 - val_accuracy: 0.8562
Epoch 4/20
500/500 [=====] - 95s 190ms/step - loss: 0.3026 - accuracy: 0.8827 - val_loss: 0.3520 - val_accuracy: 0.8531
Epoch 5/20
500/500 [=====] - 93s 187ms/step - loss: 0.2836 - accuracy: 0.8866 - val_loss: 0.3318 - val_accuracy: 0.8469
Epoch 6/20
500/500 [=====] - 94s 189ms/step - loss: 0.2700 - accuracy: 0.8922 - val_loss: 0.3663 - val_accuracy: 0.8469
Epoch 7/20
500/500 [=====] - 93s 187ms/step - loss: 0.2611 - accuracy: 0.8947 - val_loss: 0.3769 - val_accuracy: 0.8469
Epoch 8/20
500/500 [=====] - 94s 189ms/step - loss: 0.2521 - accuracy: 0.8974 - val_loss: 0.3786 - val_accuracy: 0.8469
Epoch 9/20
500/500 [=====] - 93s 187ms/step - loss: 0.2466 - accuracy: 0.8987 - val_loss: 0.3821 - val_accuracy: 0.8500
Epoch 10/20
500/500 [=====] - 94s 188ms/step - loss: 0.2368 - accuracy: 0.9031 - val_loss: 0.3941 - val_accuracy: 0.8500
Epoch 11/20
500/500 [=====] - 93s 186ms/step - loss: 0.2313 - accuracy: 0.9031 - val_loss: 0.3890 - val_accuracy: 0.8438
Epoch 12/20
500/500 [=====] - 94s 187ms/step - loss: 0.2273 - accuracy: 0.9054 - val_loss: 0.4154 - val_accuracy: 0.8469
Epoch 13/20
500/500 [=====] - 95s 190ms/step - loss: 0.2225 - accuracy: 0.9077 - val_loss: 0.4340 - val_accuracy: 0.8531
Epoch 14/20
500/500 [=====] - 94s 188ms/step - loss: 0.2190 - accuracy: 0.9086 - val_loss: 0.4324 - val_accuracy: 0.8531
Epoch 15/20
500/500 [=====] - 95s 190ms/step - loss: 0.2170 - accuracy: 0.9077 - val_loss: 0.4293 - val_accuracy: 0.8438
Epoch 16/20
500/500 [=====] - 94s 188ms/step - loss: 0.2131 - accuracy: 0.9100 - val_loss: 0.4384 - val_accuracy: 0.8438
Epoch 17/20
500/500 [=====] - 94s 189ms/step - loss: 0.2090 - accuracy: 0.9106 - val_loss: 0.4585 - val_accuracy: 0.8469
Epoch 18/20
500/500 [=====] - 94s 187ms/step - loss: 0.2057 - accuracy: 0.9127 - val_loss: 0.4670 - val_accuracy: 0.8344
Epoch 19/20
500/500 [=====] - 93s 187ms/step - loss: 0.2040 - accuracy: 0.9136 - val_loss: 0.4849 - val_accuracy: 0.8469
Epoch 20/20
500/500 [=====] - 94s 188ms/step - loss: 0.2039 - accuracy: 0.9139 - val_loss: 0.4822 - val_accuracy: 0.8500
    
```

Gambar 10. Nilai accuracy dan loss 32 unit 20 epoch



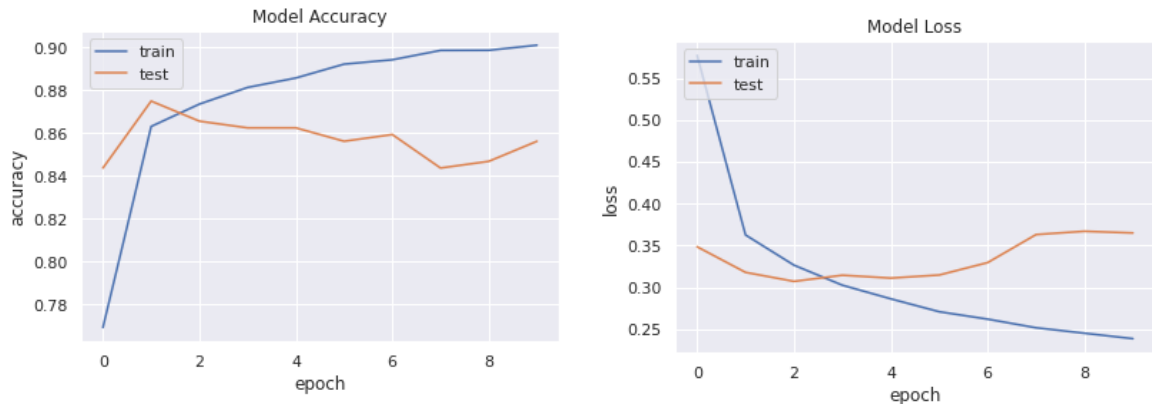
Gambar 11. Graph 32 unit 20 epoch.

Kemudian Pada percobaan 64 unit di layer LSTM, epoch 10, batch size 64 dan validation step 5 maka menghasilkan nilai accuracy 90.11% dan loss 23.88% untuk data training dan untuk data testing menghasilkan accuracy 85.62% dan loss 36.51% seperti pada gambar 12. untuk graph per epoch nya seperti pada gambar 13 dimana pada gambar grafik tersebut pada model loss menunjukkan mulai stabil antara loss data train dan data testing. Ini menunjukkan bahwasanya model dengan parameter berikut merupakan model yang baik.

```

Epoch 1/10
500/500 [=====] - 139s 274ms/step - loss: 0.5774 - accuracy: 0.7693 - val_loss: 0.3486 - val_accuracy: 0.8438
Epoch 2/10
500/500 [=====] - 137s 274ms/step - loss: 0.3627 - accuracy: 0.8632 - val_loss: 0.3180 - val_accuracy: 0.8750
Epoch 3/10
500/500 [=====] - 136s 272ms/step - loss: 0.3266 - accuracy: 0.8736 - val_loss: 0.3073 - val_accuracy: 0.8656
Epoch 4/10
500/500 [=====] - 136s 273ms/step - loss: 0.3027 - accuracy: 0.8814 - val_loss: 0.3146 - val_accuracy: 0.8625
Epoch 5/10
500/500 [=====] - 138s 276ms/step - loss: 0.2864 - accuracy: 0.8858 - val_loss: 0.3112 - val_accuracy: 0.8625
Epoch 6/10
500/500 [=====] - 137s 273ms/step - loss: 0.2710 - accuracy: 0.8923 - val_loss: 0.3149 - val_accuracy: 0.8562
Epoch 7/10
500/500 [=====] - 139s 277ms/step - loss: 0.2621 - accuracy: 0.8943 - val_loss: 0.3297 - val_accuracy: 0.8594
Epoch 8/10
500/500 [=====] - 138s 275ms/step - loss: 0.2518 - accuracy: 0.8987 - val_loss: 0.3632 - val_accuracy: 0.8438
Epoch 9/10
500/500 [=====] - 137s 273ms/step - loss: 0.2453 - accuracy: 0.8987 - val_loss: 0.3671 - val_accuracy: 0.8469
Epoch 10/10
500/500 [=====] - 137s 274ms/step - loss: 0.2388 - accuracy: 0.9011 - val_loss: 0.3651 - val_accuracy: 0.8562
    
```

Gambar 12. Nilai accuracy dan Loss 64 unit 10 epoch



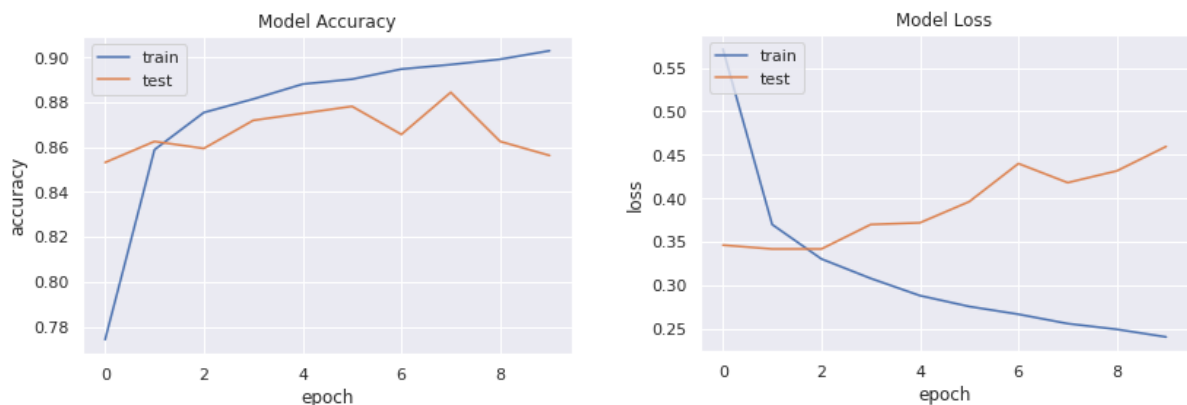
Gambar 13. Graph 64 unit 10 epoch.

Selanjutnya Pada percobaan 100 unit di *layer* LSTM, *epoch* 10, *batch size* 64 dan *validation step* 5 maka menghasilkan nilai *accuracy* 90.29% dan *loss* 24.05% untuk *data training* dan untuk *data testing* menghasilkan *accuracy* 85.62% dan *loss* 45.96% seperti pada gambar 14. untuk *graph* per *epoch* nya seperti pada gambar 15 dimana pada gambar grafik tersebut pada model *loss* menunjukkan adanya peningkatan *loss* pada data testing di setiap peningkatan *epoch* yang artinya *overfitting*.

```

Epoch 1/10
500/500 [=====] - 203s 400ms/step - loss: 0.5716 - accuracy: 0.7742 - val_loss: 0.3462 - val_accuracy: 0.8531
Epoch 2/10
500/500 [=====] - 201s 402ms/step - loss: 0.3699 - accuracy: 0.8588 - val_loss: 0.3417 - val_accuracy: 0.8625
Epoch 3/10
500/500 [=====] - 199s 398ms/step - loss: 0.3303 - accuracy: 0.8754 - val_loss: 0.3417 - val_accuracy: 0.8594
Epoch 4/10
500/500 [=====] - 199s 397ms/step - loss: 0.3079 - accuracy: 0.8814 - val_loss: 0.3699 - val_accuracy: 0.8719
Epoch 5/10
500/500 [=====] - 199s 397ms/step - loss: 0.2881 - accuracy: 0.8880 - val_loss: 0.3719 - val_accuracy: 0.8750
Epoch 6/10
500/500 [=====] - 199s 397ms/step - loss: 0.2756 - accuracy: 0.8902 - val_loss: 0.3961 - val_accuracy: 0.8781
Epoch 7/10
500/500 [=====] - 200s 401ms/step - loss: 0.2666 - accuracy: 0.8947 - val_loss: 0.4400 - val_accuracy: 0.8656
Epoch 8/10
500/500 [=====] - 199s 399ms/step - loss: 0.2559 - accuracy: 0.8967 - val_loss: 0.4180 - val_accuracy: 0.8844
Epoch 9/10
500/500 [=====] - 198s 397ms/step - loss: 0.2493 - accuracy: 0.8991 - val_loss: 0.4316 - val_accuracy: 0.8625
Epoch 10/10
500/500 [=====] - 199s 398ms/step - loss: 0.2405 - accuracy: 0.9029 - val_loss: 0.4596 - val_accuracy: 0.8562
    
```

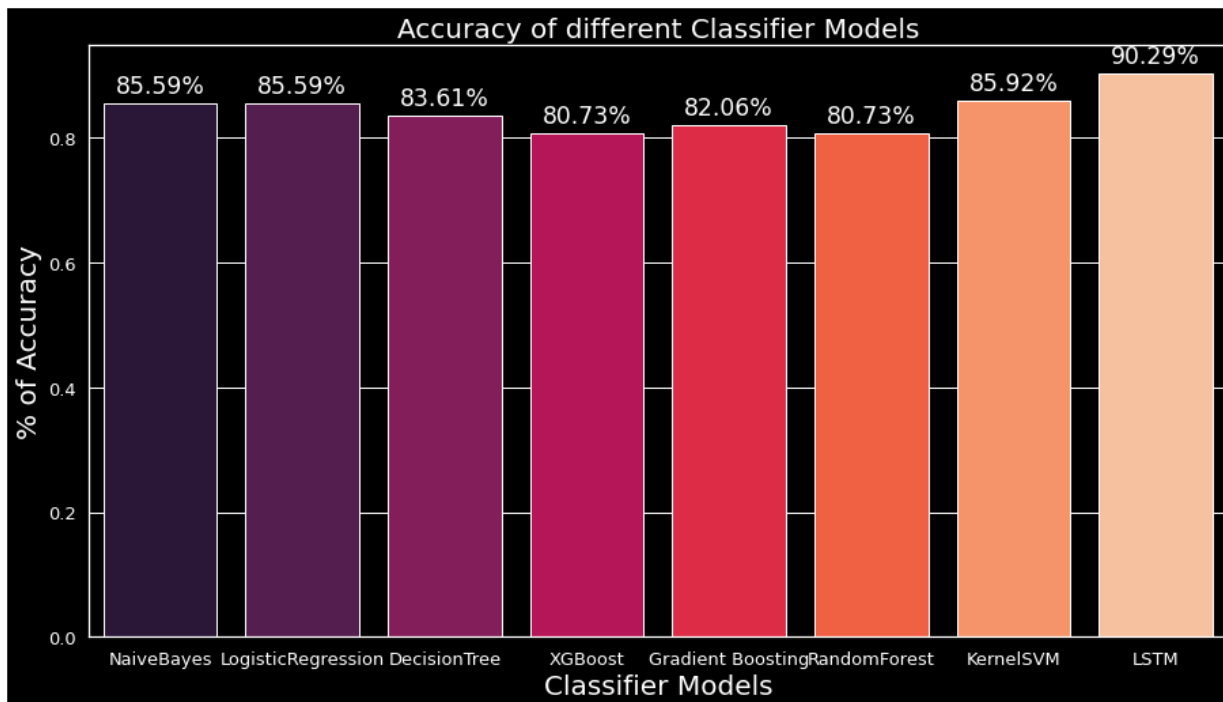
Gambar 14. Graph 100 unit 10 epoch.



Gambar 15. Graph 100 unit 10 epoch.

3.3 Perbandingan Metode Klasifikasi

Selain itu dataset juga dilakukan pengujian pada beberapa metode klasifikasi seperti metode *Naive Bayes*, *Logistic*, *Decision Tree*, *XGBoost*, *Gradient Boosting*, *Random Forest*, *SVM* untuk dilakukan perbandingan *accuracy* yang mana hasilnya seperti pada gambar 15. Dimana akurasi pada masing-masing metode yaitu *Naive Bayes* sebesar 85.59%, *Logistic Regression* sebesar 85.59%, *Decision Tree* sebesar 83.61%, *XG Boost* sebesar 80.73%, *Gradient Boost* sebesar 82.06%, *Random Forest* sebesar 80.73% dan *Support Vector Machine* sebesar 85.92% sedangkan metode *Long Short-Term Memory* (LSTM) mempunyai akurasi tertinggi dibandingkan dengan akurasi metode klasifikasi lainnya yaitu sebesar 90.29%



Gambar 15. Perbandingan akurasi pada setiap metode

4. KESIMPULAN

Berdasarkan penelitian ini dataset yang digunakan ialah data nama paket pengadaan yang diperoleh dari website Sistem Informasi Rencana Umum Pengadaan (SIRUP) sebanyak 40000 data *text*. Dengan perbandingan *data training* dan *data testing* sebesar 80% *data training* dan 20% *data testing*. Dilakukan beberapa eksperimen dari berbagai metode dimana akurasi pada masing-masing metode klasifikasi lainnya yaitu *Naïve Bayes* sebesar 85.59%, *Logistic Regression* sebesar 85.59%, *Decision Tree* sebesar 83.61%, *XG Boost* sebesar 80.73%, *Gradient Boost* sebesar 82.06%, *Random Forest* sebesar 80.73% dan *Support Vector Machine* sebesar 85.92%. Metode dengan akurasi terbaik yaitu metode *Long Short-Term Memory (LSTM)* pada akurasi 90.25%. Adapun arsitektur model LSTM yang digunakan ialah 100 unit di *layer LSTM*, *epoch 10*, *batch size 64* dan *validation step 5*.

REFERENCES

- [1] V. Mayasari, "Perbandingan Pelelangan Berbasis Sistem Manual Dengan Sistem Layanan Pengadaan Secara Elektronik (Lpse)," *Konstruksia*, pp. 79-88, 2020.
- [2] A. Adriansyah, M. B. Alexandri dan M. Halimah, "IMPLEMENTASI KEBIJAKAN PERENCANAAN PENGADAAN BARANG DAN JASA PEMERINTAH DI DINAS KESEHATAN KABUPATEN BANDUNG.," *esponsive: Jurnal Pemikiran Dan Penelitian Administrasi, Sosial, Humaniora Dan Kebijakan Publik*, vol. 4, no. 1, pp. 13-22, 2021.
- [3] K. Kowsari, K. J. Meimandi, M. Heidarysafa, S. Mendu, L. Barnes dan D. Brown, "Text classification algorithms: A survey," *Information*, vol. 10, no. 4, p. 150, 2019.
- [4] M. A. Ramdhani, D. S. Maylawati dan T. Mantoro, "Indonesian news classification using convolutional neural network," *Indonesian Journal of Electrical Engineering and Computer Science*, vol. 19, no. 2, pp. 1000-1009, 2020.
- [5] A. Kulkarni dan A. Shivananda, "Deep learning for NLP," *Natural language processing recipes*, pp. 213-262, 2021.
- [6] Y. yuli Astari, A. Afiyati dan S. W. Rozaqi, "Analisis Sentimen Multi-Class pada Sosial Media menggunakan metode Long Short-Term Memory (LSTM)," *Jurnal Linguistik Komputasional*, vol. 4, no. 1, pp. 8-12, 2021.
- [7] M. Ridwan dan A. Muzakir, "Model Klasifikasi Ujaran Kebencian pada Data Twitter dengan Menggunakan CNN-LSTM," *Teknomatika*, vol. 12, no. 02, pp. 209-218, 2022.
- [8] W. Hastomo dan A. Satyo, "Long Short Term Memory Machine Learning Untuk Memprediksi Akurasi Nilai Tukar IDR Terhadap USD," *Prosiding SeNTIK*, vol. 3, no. 1, pp. 115-124, 2019.
- [9] Y. Widhiyasana, T. Semiawan, I. Gibran, A. Mudzakir dan M. R. Noor, "Penerapan Convolutional Long Short-Term Memory untuk Klasifikasi Teks Berita Bahasa Indonesia," *Jurnal Nasional Teknik Elektro dan Teknologi Informasi*, vol. 10, no. 4, pp. 354-361, 2021.
- [10] A. S. Talita dan A. Wiguna, "Implementasi Algoritma Long Short-Term Memory (LSTM) Untuk Mendeteksi Ujaran Kebencian (Hate Speech) Pada Kasus Pilpres 2019," *MATRIK: Jurnal Manajemen, Teknik Informatika dan Rekayasa Komputer*, vol. 19, no. 1, pp. 37-44, 2019.
- [11] D. T. Hermanto, A. Setyanto dan E. T. Luthfi, "Algoritma LSTM-CNN untuk Binary Klasifikasi dengan Word2vec pada Media Online," *Creative Information Technology Journal*, vol. 8, no. 1, pp. 64-77, 2021.



- [12] D. F. Zhafira, B. Rahayudi dan I. Indriati , “Analisis Sentimen Kebijakan Kampus Merdeka Menggunakan Naive Bayes dan Pembobotan TF-IDF Berdasarkan Komentar pada Youtube,” *Jurnal Sistem Informasi, Teknologi Informasi, dan Edukasi Sistem Informasi*, vol. 2, no. 1, 2021.
- [13] A. B. P. Negara, H. Muhandi dan F. Sajid, “Perbandingan Algoritma Klasifikasi terhadap Emosi Tweet Berbahasa Indonesia,” *JEPIN (Jurnal Edukasi dan Penelitian Informatika)*, vol. 7, no. 2, pp. 242-249, 2021.
- [14] W. T. Handoko, E. Supriyanto, D. I. Purwadi, Z. Budiarto dan H. Listiyono, “Klasifikasi Opini Pengguna Media Sosial Twitter Terhadap JNT Di Indonesia dengan Algoritma Decision Tree,” *J-SAKTI (Jurnal Sains Komputer dan Informatika)*, vol. 6, no. 2, pp. 790-799, 2022.
- [15] W. Willy, D. P. Rini dan S. Samsuryadi, “Perbandingan Algoritma Random Forest Classifier, Support Vector Machine dan Logistic Regression Clasifier Pada Masalah High Dimension (Studi Kasus: Klasifikasi Fake News),” *JURNAL MEDIA INFORMATIKA BUDIDARMA*, vol. 5, no. 4, pp. 1720-1728, 2021.
- [16] I. A. Ropikoh, R. Abdulhakim, U. Enri dan N. Sulistiyowati, “Penerapan Algoritma Support Vector Machine (SVM) untuk Klasifikasi Berita Hoax Covid-19,” *Journal of Applied Informatics and Computing*, vol. 5, no. 1, pp. 64-73, 2021.
- [17] M. Kubat, *An introduction to machine learning*, Cham, Switzerland: Springer International Publishing, 2017.
- [18] M. J. U. H. Bahrudin dan H. Gutama, “Scraping Web Marketplace Menggunakan Metode DOM Parsing Untuk Pengumpulan Data Produk,” *JURNAL ILMIAH INFORMATIKA*, vol. 8, no. 01, pp. 77-80, 2020.
- [19] J. W. G. Putra, *Pengenalan Konsep Pembelajaran Mesin dan Deep Learning*, Tokyo, Jepang, 2020.