

Penerapan Metode *TF-IDF* dan *Deep Neural Network* untuk Analisa Sentimen pada Data Ulasan Hotel

Fachrizza Dian Adhiatma¹, Anita Qoiriah²

^{1,2} Jurusan Teknik Informatika, Prodi Teknik Informatika, Fakultas Teknik, Universitas Negeri Surabaya

¹fachrizza.18061@mhs.unesa.ac.id

²anitaqoiriah@unesa.ac.id

Abstrak—Data ulasan pelanggan memainkan peran yang penting dalam dunia bisnis karena dapat digunakan oleh para pelaku usaha untuk lebih memahami pelanggan serta mengetahui kelebihan dan kekurangan dari produk yang ditawarkan. Namun, data ulasan tersebut hadir dalam jumlah yang besar di internet dan terus bertambah setiap harinya sehingga proses analisis menjadi lebih susah jika dilakukan secara manual. Analisa sentimen merupakan teknik untuk mengklasifikasikan data-data teks kedalam berbagai kelas sentimen. Penelitian ini bertujuan untuk membuat sistem analisa sentimen untuk data perhotelan menggunakan metode *TF-IDF* dan *deep neural network*. Metode *TF-IDF* digunakan untuk mengubah data teks menjadi vektor numerik, sedangkan *deep neural network* digunakan untuk mengklasifikasikan data vektor tersebut kedalam tiga kelas sentimen, yaitu positif, negatif dan netral. Penelitian ini juga bertujuan untuk mengeksplorasi kombinasi parameter yang memberikan hasil terbaik atau biasa juga disebut dengan *hyperparameter tuning*. Beberapa parameter yang diuji adalah kombinasi *n-gram*, nilai *learning rate*, dan jumlah *hidden layers*. Dari hasil penelitian yang dilakukan, metode *TF-IDF* dan *deep neural network* terbukti mampu untuk menyelesaikan permasalahan analisa sentimen untuk data perhotelan. Selain itu, kombinasi parameter yang memberikan hasil terbaik yaitu *n-gram* dengan jumlah 3, nilai *learning rate* 0,001, dan *hidden layers* dengan jumlah 3.

Kata Kunci— analisa sentimen, *TF-IDF*, *deep neural network*, perhotelan, *hyperparameter tuning*

I. PENDAHULUAN

Industri pariwisata memainkan peran yang cukup penting dalam perkembangan ekonomi di Indonesia, baik sebagai sumber penghasil devisa negara, pencipta lapangan kerja, dan kesempatan berusaha, serta pemerataan pendapatan. Dengan semakin berkembangnya industri pariwisata, maka permintaan akan penyediaan kamar hotel dan akomodasi lainnya juga semakin meningkat. Berdasarkan data Badan Pusat Statistik (BPS) pada tahun 2020, tercatat ada sebanyak 30.823 penyedia akomodasi dengan jumlah kamar mencapai 870.783 kamar [1]. Selain mendorong perkembangan industri pariwisata, sektor perhotelan juga berkontribusi dalam penyediaan lapangan kerja melalui struktur padat karya dan meningkatkan belanja daerah dengan cepat [2].

Perkembangan internet yang semakin luas mendorong para pelaku usaha di bidang perhotelan untuk melakukan digitalisasi bisnis. Hal tersebut dapat dilihat dari munculnya industri-industri *Online Travel Agent* (OTA) yang memiliki fitur untuk melakukan *booking* kamar hotel. OTA juga dilengkapi dengan fasilitas pemberian ulasan terhadap hotel yang telah dipesan oleh pengguna. Data ulasan ini umumnya digunakan sebagai

tolak ukur kepuasan pelanggan dan sumber informasi mengenai layanan serta fitur yang harus ditingkatkan oleh penyedia layanan. Namun, data ulasan tersebut hadir dalam jumlah yang banyak sehingga menyebabkan kesulitan dalam menganalisis dan menyimpulkan seluruh ulasan secara manual [3]. Selain itu, industri hotel memiliki banyak cara dalam mengumpulkan *feedback* dari pelanggan seperti *guest comment cards*, kuisioner *in-room*, survey telepon, survey daring, survey email, *interview*, dan lain-lain [4]. Hal tersebut dilakukan karena perbedaan karakteristik demografi pelanggan seperti jenis kelamin, usia, dan tujuan kunjungan mempengaruhi cara pelanggan dan preferensi *channel* yang dipilih dalam menyampaikan opini [5].

Analisa sentimen atau juga bisa disebut *opinion mining* adalah salah satu teknik di bidang *Natural Language Processing* (NLP) untuk mengekstrak dan menganalisis opini terhadap entitas seperti topik, produk, dan layanan. Analisa sentimen dapat dikategorikan sebagai permasalahan klasifikasi teks karena tujuan akhirnya adalah mengklasifikasikan apakah suatu teks tergolong sentimen positif atau negatif [6]. Analisa sentimen dapat digunakan sebagai alat yang kuat bagi perusahaan, pemerintahan, dan peneliti untuk mengekstrak dan menganalisis pandangan publik, menambah pengetahuan bisnis, dan membuat keputusan yang lebih baik.

Salah satu metode yang sering digunakan untuk melakukan analisa sentimen adalah *supervised machine learning*. Beberapa penelitian tentang analisa sentimen menggunakan *supervised machine learning* antara lain : Studi mengenai perbandingan model *deep learning* dalam melakukan analisa sentimen pada data ulasan restoran di masa pandemi COVID-19 [7]; Eksperimen mengenai perbandingan model analisa sentimen untuk bahasa Turki menggunakan metode klasik *supervised machine learning* dengan *TF-IDF* sebagai metode ekstraksi fiturnya [8]; Eksperimen menggunakan dataset Twitter tentang Uber dan Ola untuk membuat analisa sentimen dengan model *deep neural network*. Selain itu, *hyperparameter tuning* juga dilakukan untuk menentukan jumlah *layer* yang memberikan akurasi terbaik [9]; Percobaan tentang penggunaan data ulasan hotel berbahasa Indonesia untuk membuat model analisa sentimen dengan LSTM (*Long Short-Term Memory*) dan Word2vec sebagai metode ekstraksi fiturnya. Eksperimen pada parameter juga dilakukan pada arsitektur Word2vec, metode evaluasi, dimensi vektor, nilai *dropout*, teknik *pooling*, dan *learning rate* untuk menentukan kombinasi parameter yang memberikan akurasi terbaik [10]; Percobaan mengenai variasi praproses data teks untuk analisa sentimen data Twitter menggunakan *TF-IDF* dan *backpropagation neural network*. Dari penelitian tersebut dapat

disimpulkan bahwa transformasi kata gaul, *cleaning*, dan normalisasi dapat meningkatkan akurasi model [11].

Dalam penelitian ini, sistem analisa sentimen untuk data ulasan hotel berbahasa indonesia dibuat menggunakan metode TF-IDF dan *deep neural network*.

A. TF-IDF

Menurut [11], TF-IDF merupakan teknik pembobotan kata yang berbasis pada statistik kemunculan kata dan tingkat kepentingan dokumen yang mengandungnya. Pembobotan kata ini merupakan hasil dari perkalian *term frequency* dan *inverse document frequency* yang tiap nilainya didapatkan dari (1) dimana w_i adalah kata ke- i , d adalah dokumen, $TF(w_i, d)$ adalah jumlah kemunculan kata w_i pada dokumen d dan $IDF(w_i)$ adalah nilai *inverse document frequency* dari w_i .

$$TF * IDF = TF(w_i, d) * IDF(w_i) \quad (1)$$

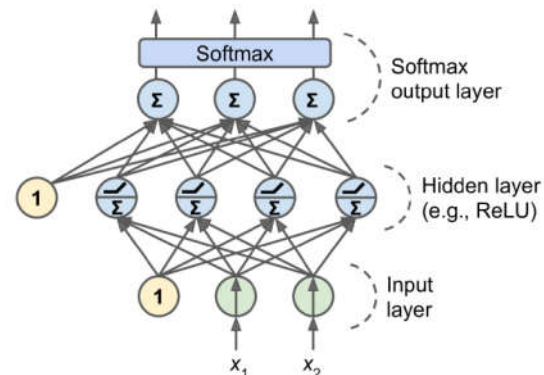
Sedangkan nilai IDF didapatkan dari (2) dimana N merupakan jumlah seluruh dokumen dan $DF(w_i)$ merupakan jumlah dokumen yang mengandung kata w_i .

$$IDF(w_i) = \frac{N}{DF(w_i)} \quad (2)$$

Word2Vec secara umum memiliki kemampuan yang lebih baik daripada TF-IDF dalam melakukan ekstraksi fitur. Namun, berdasarkan penelitian yang telah dilakukan mengenai perbandingan Word2Vec dan TF-IDF untuk *task* klasifikasi morbiditas dari catatan klinis pasien, menyatakan hasil yang sebaliknya. Hal tersebut dikarenakan *task* yang bersifat *domain specific* dan dataset yang digunakan tidak cukup besar sehingga menyebabkan Word2Vec kurang mampu menangkap dan mempelajari fitur-fitur spesifik yang ada dalam data [12]. TF-IDF dipilih sebagai metode ekstraksi fitur dalam penelitian ini dikarenakan dataset yang digunakan bersifat *domain specific*, yaitu data perhotelan dan juga ukuran dataset yang tidak cukup besar.

B. Deep Neural Network

Adalah sebuah metode yang bekerja dengan cara mengekstrak fitur dari kombinasi linear dalam data masukan dan menghasilkan fungsi nonlinear dari fitur tersebut. Umumnya arsitektur *deep neural network* terdiri dari *input layer*, *hidden layer*, dan *output layer* dimana di setiap *layer* terdapat banyak neuron yang terorganisir seperti yang ditunjukkan pada Gbr. 1.



Gbr. 1 Contoh arsitektur deep neural network

Koneksi antar *layer* dibentuk oleh hubungan antar neuron dari setiap *layer*. Setiap koneksi memiliki nilai bobot yang digunakan oleh *perceptron* untuk melakukan perkalian dengan data masukan. Hasil komputasi tersebut akan dibawa ke fungsi aktivasi untuk dilakukan transformasi nonlinear seperti ReLU pada *hidden layer* dan softmax pada *output layer* [6]. Fungsi softmax akan mengeluarkan nilai antara 0 dan 1 untuk masing-masing kelas, nilai-nilai ini merepresentasikan probabilitas hasil prediksi model kepada masing-masing kelas.

Proses pembelajaran atau *training* dalam *deep neural network* bertujuan untuk meminimalkan nilai *loss* atau *error* dengan cara mengestimasi nilai bobot di setiap koneksi. Proses ini terjadi secara iteratif dalam dua tahap yaitu *forward pass* dan *backward pass*. *Forward pass* bertujuan untuk menghitung nilai *error* antara hasil prediksi model dengan label asli menggunakan *loss function*, sedangkan *backward pass* bertujuan untuk melakukan *update* pada nilai bobot melalui sebuah metode optimasi. Metode optimasi tersebut akan menghitung nilai gradien dari *error* model sesuai dengan parameter model (bobot dan bias) dimana nilai gradien ini akan digunakan untuk menentukan besar kecilnya nilai bobot dan bias yang harus diubah untuk mengurangi *error* [13].

Eksperimen terhadap variasi parameter juga akan dilakukan untuk menemukan kombinasi parameter yang memberikan hasil prediksi paling akurat. Pada TF-IDF akan dilakukan eksperimen terhadap kombinasi n -gram yang digunakan. N -gram adalah sebuah proses pemisahan kata-kata berdasarkan aturan $|kata|n$, dimana n merupakan jumlah faktor pemisah kata yang diinginkan ($n=1,2,3,..$). Misalnya $|kaum Aad dan Tsamud|2$ akan menjadi (kaum Aad), (Aad dan), (dan Tsamud) [14]. Sedangkan pada *deep neural network* akan dilakukan eksperimen terhadap kombinasi jumlah *hidden layer* dan nilai *learning rate*. *Deep neural network* memiliki arsitektur yang fleksibel karena dapat memiliki jumlah neuron yang bervariasi pada setiap *layer* dan jumlah *layer*-nya pun juga bervariasi. Semakin banyak jumlah *layer* yang dimiliki, maka semakin kompleks pula pola yang dapat dipelajari oleh model tersebut [15]. Oleh karena itu, perlu dilakukan eksperimen untuk menentukan jumlah *layer* atau *hidden layer* yang tepat. Proses pembelajaran dalam *deep neural network* terjadi secara iteratif dan bertujuan untuk meminimalkan nilai *loss*. Kecepatan *deep*

neural network dalam menemukan titik *loss* terendah ini dipengaruhi oleh nilai *learning rate*. Jika nilai *learning rate* yang dipilih terlalu kecil, maka *deep neural network* akan membutuhkan waktu yang sangat lama untuk mencapai kondisi konvergen (*loss* terendah), meskipun ada jaminan bahwa tidak ada titik *local minimum* yang terlewatkan. Jika nilai *learning rate* yang dipilih terlalu besar, maka fluktuasi nilai *loss* selama proses pembelajaran akan menjadi relatif besar sehingga *deep neural network* akan kesulitan untuk mencapai kondisi konvergen [16]. Oleh karena itu, diperlukan eksperimen untuk menentukan nilai *learning rate* yang tepat sehingga proses pembelajaran tidak memakan waktu yang terlalu lama dan juga tidak terjadi fluktuasi nilai *loss*.

Untuk menentukan kombinasi parameter mana yang memiliki performa terbaik, diperlukan suatu metode evaluasi. Salah satu cara paling komprehensif dalam mengevaluasi model klasifikasi adalah menggunakan *confusion matrix*. Keluaran dari *confusion matrix* berupa matriks dimana barisnya merepresentasikan kelas yang sebenarnya dan kolomnya merepresentasikan kelas hasil prediksi. Setiap data merepresentasikan seberapa sering suatu sampel yang memiliki kelas yang sesuai dengan baris diprediksi sebagai kelas yang sesuai dengan kolom. Misalkan pada klasifikasi biner terdapat kelas “nine” dan “not nine”, maka *confusion matrix*nya dapat diilustrasikan seperti Gbr. 2.

true 'not nine'	401	2
true 'nine'	8	39
	predicted 'not nine'	predicted 'nine'

Gbr. 2 Ilustrasi confusion matrix pada klasifikasi biner dari kelas “nine” dan “not nine”

Data-data pada diagonal utama *confusion matrix* menyatakan jumlah klasifikasi yang benar, sedangkan data yang lain menyatakan jumlah klasifikasi yang salah. Jika kelas “nine” disebut sebagai kelas positif sedangkan kelas “not nine” disebut sebagai kelas negatif maka data yang diklasifikasikan sebagai kelas positif dengan benar dapat disebut sebagai *true positive* dan data yang diklasifikasikan dengan benar sebagai kelas negatif disebut dengan *true negative*, sedangkan klasifikasi yang salah disebut sebagai *false positive* dan *false negative* [17]. Istilah-istilah tersebut sering disingkat sebagai TP, TN, FP, dan FN dimana jika diilustrasikan sebagai *confusion matrix* akan tampak seperti Gbr. 3.

negative class	TN	FP
positive class	FN	TP
	predicted negative	predicted positive

Gbr. 3 Ilustrasi confusion matrix pada klasifikasi biner

Memeriksa *confusion matrix* secara keseluruhan untuk melakukan evaluasi cukup rumit dikarenakan proses dilakukan secara manual dan kualitatif. Oleh karena itu, diperlukan metrik-metrik yang dapat meringkas isi dari *confusion matrix*. Beberapa metrik yang tersebut yaitu *accuracy*, *precision*, *recall*, dan *f1 score*.

Accuracy atau akurasi adalah metrik yang paling umum digunakan dalam permasalahan klasifikasi, nilai akurasi merepresentasikan rasio antara prediksi yang benar dengan total jumlah data. Nilai akurasi dapat dihitung dari (3). *Precision* adalah metrik yang merepresentasikan proporsi dari sampel yang terprediksi dengan benar sebagai kelas positif dengan jumlah sampel yang diprediksi sebagai kelas positif. Nilai dari *precision* dapat dihitung dari (4). *Recall* adalah metrik yang merepresentasikan proporsi dari sampel yang diprediksi dengan benar sebagai kelas positif dengan semua sampel yang memiliki kelas positif. Nilai *recall* dapat dihitung menggunakan (5). *F1 score* adalah metrik dengan range nilai 0 dan 1 yang digunakan untuk menghitung nilai *precision* dan *recall* dengan cara menghitung rata-rata harmonik dari kedua metrik ini. Nilai *f1 score* dapat dihitung menggunakan (6).

Setelah didapatkan model dengan kombinasi *hyperparameter* yang menghasilkan performa terbaik, maka model tersebut akan *dideploy* menjadi API. API atau REST API adalah salah satu tipe *web server* yang memungkinkan *client* untuk mengakses sumber daya yang dimiliki oleh suatu sistem. Sederhananya, API dapat mengekspos sejumlah data dan fungsi untuk memfasilitasi interaksi antar program komputer dan memungkinkan mereka untuk saling bertukar informasi. REST API memiliki semua aspek dari HTTP (HyperText Transfer Protocol), termasuk semua metode *request*, kode *response*, dan *header* pesan. *Client* dapat menentukan metode yang diinginkan untuk berinteraksi dengan API melalui *request* HTTP seperti GET, PUT dan POST [18].

II. METODE PENELITIAN

Penelitian ini dilakukan sesuai dengan langkah-langkah berikut.

A. Pengumpulan Dataset

Data yang digunakan dalam penelitian ini adalah data-data ulasan hotel yang berbentuk teks. Proses pengambilan data dilakukan dengan menggunakan teknik *web scrapping* pada situs tiket.com.

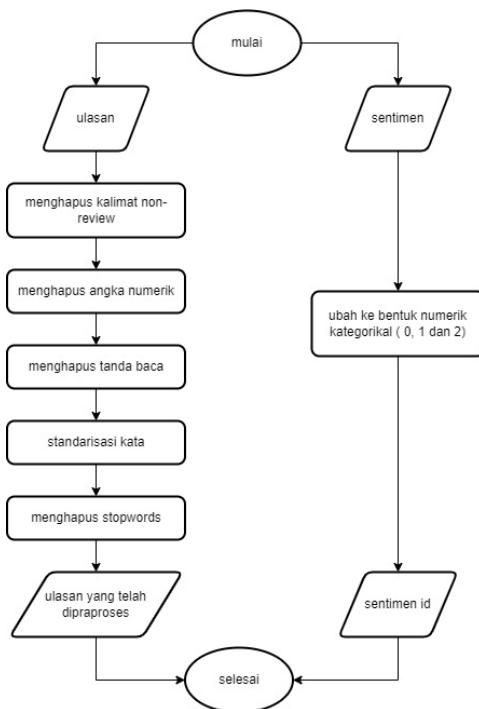
B. Analisis Data dan Pelabelan

Data-data ulasan yang telah dikumpulkan akan dianalisis dan diberikan label sentimen secara manual. Sentimen dibagi kedalam tiga kelas, yaitu positif, negatif, dan netral. data sentimen ini akan menjadi label, sedangkan data teks ulasan digunakan sebagai fitur.

C. Prapemrosesan Data

Prapemrosesan data dilakukan secara terpisah untuk data teks ulasan dan data sentimen. Pada data teks ulasan, prapemrosesan data yang dilakukan yaitu menghapus kalimat *non-review*, menghapus angka numerik, dan tanda baca seperti yang dilakukan pada [7], standarisasi kata, serta menghapus *stopwords*. Penghapusan *stopwords* dilakukan karena *stopwords* tidak memiliki makna yang signifikan sehingga proses bisa terfokus pada kata-kata yang penting [10]. Standarisasi kata atau transformasi kata gaul dapat meningkatkan performa model analisa sentimen [11].

Sedangkan data sentimen diubah kedalam bentuk numerik kategorikal. Tahap-tahap prapemrosesan data untuk data ulasan dan sentimen ditunjukkan oleh Gbr. 4.



Gbr. 4 Alur prapemrosesan data untuk data ulasan dan sentimen

D. Split Data

Dataset yang terdiri atas teks ulasan dan sentimen dipisah menjadi dua bagian, yaitu data *training* dan data *testing* dengan persentase masing-masing 80% dan 20%.

Sesuai dengan namanya, model dilatih menggunakan data *training* dan di tes menggunakan data *testing*. Nilai *error* yang didapatkan pada data baru disebut dengan *generalization error*. Dengan mengevaluasi model menggunakan data *testing*, maka

nilai *error* ini dapat diestimasi dimana nilai ini memberitahu tentang bagaimana performa model terhadap data baru yang belum pernah dilihat.

E. Vektorisasi Teks

Vektorisasi dilakukan dengan tujuan untuk mengubah data ulasan yang semula dalam bentuk teks menjadi bentuk vektor numerik agar dapat diproses oleh *deep neural network*. Selain itu, vektorisasi juga digunakan mengekstraksi fitur dalam teks. Metode vektorisasi teks yang digunakan adalah TF-IDF. Masukan TF-IDF adalah data teks ulasan yang telah dipraproses dan keluarannya berupa matriks yang jumlah barisnya sama dengan jumlah data dalam dataset ulasan, sedangkan jumlah kolomnya sama dengan jumlah token dalam dataset. Elemen-elemen dalam matriks tersebut berisi nilai bobot yang menyatakan besarnya bobot kata dalam suatu teks. Nilai bobot tersebut didapatkan dari (1) dan (2). Model TF-IDF dilatih menggunakan data *training*. Contoh dari hasil TF-IDF ditunjukkan pada Gbr. 5.

	dan	enak	pelayanan	ramah	sarapan
pelayanan ramah	0.000000	0.000000	0.707107	0.707107	0.000000
pelayanan ramah dan sarapan enak	0.499221	0.499221	0.355200	0.355200	0.499221

Gbr. 5 Hasil vektorisasi teks menggunakan TF-IDF

F. Training Model

Masukan dari model *deep neural network* adalah data vektor TF-IDF yang telah dibuat pada tahap vektorisasi teks. Jika mengacu pada Gbr. 16, maka jumlah kolom matriks yang diharapkan adalah 1819 yang artinya jumlah token dari TF-IDF yang diharapkan adalah sebanyak 1819 token. Jumlah ini dapat bervariasi, tergantung dari besar kecilnya dataset dan kombinasi n-gram yang digunakan. Keluaran dari model *deep neural network* adalah nilai probabilitas sejumlah banyaknya kelas yang ada dan jika nilai probabilitas tersebut dijumlahkan akan bernilai 1. Dalam penelitian ini, kelas yang digunakan ada tiga yaitu positif, negatif, dan netral. Nilai probabilitas ini menyatakan tingkat kepercayaan model terhadap hasil prediksinya.

Model dilatih menggunakan data *training* dan sistem *early stopping* diterapkan selama proses pelatihan, sistem ini akan menghentikan proses pelatihan jika nilai akurasi dari data *testing* tidak mengalami kenaikan selama beberapa *epoch* (iterasi). Dalam penelitian ini, toleransi *epoch* pada *early stopping* yang diberikan adalah sebanyak dua *epoch*.

Karena sifat nondeterministik pada *deep neural network*, performa model cenderung berubah-ubah meski dilatih dengan parameter yang sama [19]. Metode *ensemble learning* dengan strategi *unweighted model averaging* diterapkan untuk mengatasi permasalahan tersebut. Jumlah model dalam *ensemble* yang digunakan berjumlah 10 buah dan perhitungan rata-rata dilakukan pada nilai probabilitas yang dihasilkan dari fungsi *softmax*.

G. Evaluasi

Evaluasi dilakukan pada data *testing* dengan mengambil nilai rata-rata *precision*, *recall*, *f1 score*, dan *accuracy* yang

secara umum, masing-masing nilainya dapat dihitung menggunakan (3), (4), (5), dan (6).

$$accuracy = \frac{TP+TN}{TP+TN+FP+FN} \quad (3)$$

$$precision = \frac{TP}{TP+FP} \quad (4)$$

$$recall = \frac{TP}{TP+FN} \quad (5)$$

$$F1 \text{ score} = 2 * \frac{precision*recall}{precision+recall} \quad (6)$$

Dalam penelitian ini, terdapat tiga kelas sentimen yaitu positif, negatif, dan netral, maka *confusion matrix*-nya dapat diilustrasikan seperti Gbr. 6 dimana TPos, TNeg, dan TNet merupakan *true* positif, *true* negatif dan *true* netral. Sedangkan, FPos, FNeg, dan FNet merupakan *false* positif, *false* negatif, dan *false* netral.

		Predicted class		
		Positif	Negatif	Netral
True class	Positif	TPos	FNeg ₁	FNet ₁
	Negatif	FPos ₁	TNeg	FNet ₂
	Netral	FPos ₂	FNeg ₂	TNet

Gbr. 6 Ilustrasi confusion matrix untuk tiga label

Nilai *precision* untuk masing-masing kelas dapat dihitung menggunakan (7), (8), dan (9) yang kemudian dijumlahkan dan dibagi dengan tiga (jumlah label) untuk mendapatkan nilai rata-rata *precision*.

$$positif \text{ precision} = \frac{TPos}{TPos+FPos_1+FPos_2} \quad (7)$$

$$negatif \text{ precision} = \frac{TNeg}{TNeg+FNeg_1+FNeg_2} \quad (8)$$

$$netral \text{ precision} = \frac{TNet}{TNet+FNeg_1+FNeg_2} \quad (9)$$

Nilai *accuracy*, *recall*, dan *F1 score* juga didapatkan dengan cara yang serupa, yaitu dengan menyesuaikan (3), (5), dan (6) ke format tiga label, kemudian dihitung nilai rata-ratanya.

H. Hyperparameter Tuning

Hyperparameter yang diuji pertama adalah jumlah n-gram pada TF-IDF, nilai-nilai n-gram yang diuji yaitu 1, 2, dan 3. Kemudian dilakukan juga pengujian pada jumlah *hidden layer*

model *deep neural network*. Jumlah *hidden layer* yang diuji yaitu 1, 2, dan 3. *Hyperparameter* terakhir yang diuji yaitu *learning rate* dengan nilai-nilai 0,001, 0,01, dan 0,1. Pengujian mengenai jumlah *hidden layers* dan nilai *learning rate* dilakukan pada nilai n-gram yang tetap, yaitu nilai n-gram yang memberikan performa terbaik pada pengujian kombinasi n-gram. Kombinasi *hyperparameter* terbaik dipilih berdasarkan nilai *f1 score* dan akurasi.

I. Deployment

Model dengan kombinasi parameter yang menghasilkan performa terbaik akan di *deploy* dalam bentuk REST API menggunakan pustaka FastAPI¹. Percobaan API ini dilakukan pada komputer lokal dengan bantuan aplikasi Postman untuk *testing* API. URL yang harus dituju yaitu *localhost* pada *port* 8000. Prediksi sentimen dapat dilakukan dengan cara mengirimkan teks ulasan dalam bentuk JSON dengan nama *text* pada *endpoint* “/sentiment” menggunakan HTTP *method* POST. *Response* dari API berupa JSON dengan dua variabel yaitu “*class*” dan “*confidence*” dimana *class* menyatakan hasil prediksi sentiment, sedangkan *confidence* menyatakan nilai probabilitas atau *confidence score*.

III. HASIL DAN PEMBAHASAN

Proses inferensi dari sistem analisa sentimen ini dimulai dari prapemrosesan data masukan yang berupa teks ulasan. Tahap-tahap prapemrosesan data yaitu menghapus kalimat *non-review*, menghapus angka numerik, menghapus tanda baca, standarisasi kata, dan menghapus *stopwords*. Gbr. 7 menunjukkan kode program yang digunakan dalam prapemrosesan data dimana setiap fungsi-fungsi prapemrosesan data dibungkus dalam modul *TextPreprocessor*.

```
text_preprocessing = TextPreprocessor()

text = "bgs hotelnya, anak2 pd suka...tp sayang mkanan kurang bervariasi"

text_preprocessed = text_preprocessing.remove_non_review_sentences(text)
text_preprocessed = text_preprocessing.remove_number(text_preprocessed)
text_preprocessed = text_preprocessing.remove_punct(text_preprocessed)
text_preprocessed = text_preprocessing.standardize(text_preprocessed)
text_preprocessed = text_preprocessing.remove_stopwords(text_preprocessed)
```

Gbr. 7 Kode program prapemrosesan data

Gbr. 8 hingga Gbr. 12 menunjukkan fungsi-fungsi prapemrosesan data yang dibungkus dalam modul *TextPreprocessor*. Fungsi penghapusan kalimat non review pada Gbr. 8 menggunakan *regular expression* untuk mendeteksi pola-pola kalimat *non-review* lalu mengubahnya kedalam bentuk string kosong.

```
def remove_non_review_sentences(self, text: str) -> str:
    result = re.sub("\d+ people found this helpful. Report", '', text).rstrip()
    result = result.replace('Apa review ini membantu? Laporkan', '')
    result = result.replace('Is this review helpful? Report', '')
    result = result.replace(u'\xa0', u'')

    return result
```

¹ <https://fastapi.tiangolo.com/>

Gbr. 8 Fungsi untuk menghapus kalimat non-review

Fungsi untuk menghapus tanda baca ditunjukkan oleh Gbr. 9 dimana *regular expression* digunakan untuk mendeteksi karakter-karakter non alfabet dan numerik lalu menghapusnya.

```
def remove_punct(self, text: str) -> str:
    result = re.sub(r'^\w\s', ' ', text)
    result = self.__remove_extra_whitespace(result)

    return result
```

Gbr. 9 Fungsi untuk menghapus tanda baca

Fungsi untuk menghapus angka numerik ditunjukkan oleh Gbr. 10. Fungsi ini menggunakan *regular expression* untuk mendeteksi karakter-karakter numerik lalu menghapusnya.

```
def remove_number(self, text: str) -> str:
    result = re.sub(r'\d+', ' ', text)
    result = self.__remove_extra_whitespace(result)

    return result
```

Gbr. 10 Fungsi untuk menghapus angka numerik

Fungsi untuk melakukan standarisasi kata ditunjukkan oleh Gbr. 11. Dalam fungsi ini, kalimat ulasan akan dipisah per kata dan kemudian dicocokkan dengan kamus yang berisi pemetaan kata baku dan non baku yang telah dibuat secara manual. Pencocokan kata ini dilakukan dengan pustaka Pandas². Kamus tersebut dalam fungsi pada Gbr. 11 bernama *kamus_alay* dan isinya ditunjukkan oleh Gbr. 23.

```
def standardize(self, text: str) -> str:
    text = text.lower().split(' ')
    text_normal = []

    for word in text:
        result = self.__kamus_alay[self.__kamus_alay['vocab'] == word]

        if len(result) > 0:
            normal_word = result['standard'].values[0]
            text_normal.append(normal_word)
        else:
            text_normal.append(word)

    return ' '.join(text_normal)
```

Gbr. 11 Fungsi untuk melakukan standarisasi kata

Gbr. 12 menunjukkan fungsi untuk menghapus *stopwords* yang mana bekerja dengan cara mencocokkan setiap kata dalam data masukan dengan daftar *stopwords* yang ada.

```
def remove_stopwords(self, text: str) -> str:
    tokens = text.split(' ')
    tokens = [token for token in tokens if token not in self.stopwords]
    result = ' '.join(tokens)

    return result
```

² <https://pandas.pydata.org/>

³ <https://scikit-learn.org/stable/index.html>

Gbr. 12 Fungsi untuk menghapus stopwords

Perbandingan data masukan sebelum dipraproses dan setelah dipraproses ditunjukkan oleh Gbr. 13.

Sebelum di praproses : bgs hotelnya, anak2 pd suka...tp sayang mkanan kurang bervariasi
Setelah di praproses : bagus hotelnya anak suka tapi sayang makanan kurang bervariasi

Gbr. 13 Perbandingan data teks masukan sebelum di praproses dan setelah di praproses

Data yang telah dipraproses tersebut kemudian diubah kedalam bentuk vektor bobot TF-IDF. Perhitungan TF-IDF dilakukan menggunakan modul *TfidfVectorizer* pada pustaka *Scikit-learn*³. *Vectorizer* ini sebelumnya telah dilatih menggunakan data *training* sehingga memiliki informasi mengenai *vocabulary* yang ada dalam data *training*. Gbr. 14 menunjukkan kode program untuk perhitungan TF-IDF pada data masukan yang telah di praproses.

```
tfidf_weight = vectorizer.transform([text_preprocessed])
```

Gbr. 14 Perbandingan data teks masukan sebelum di praproses dan setelah di praproses

Gbr. 15 menunjukkan hasil vektor TF-IDF dari data masukan yang telah dipraproses.

	about	ac	ada	ac	dingin	ac	kamar	terlalu	ac	tidak	acceptable	access	ada	...	yang	yes	yg	kurang	yogyakarta	you	you	are	can
bagus																							
hotelnya																							
anak																							
suka																							
tapi																							
sayang																							
makanan																							
kurang																							
bervariasi																							

Gbr. 15 Vektor TF-IDF dari data masukan yang telah dipraproses

Vektor TF-IDF digunakan sebagai masukan dalam *method* “predict” pada model *deep neural network* dimana hasil prediksinya berupa nilai probabilitas dari ketiga kelas sentimen seperti yang ditunjukkan pada Gbr. 17.

```
predictions = np.array([model.predict(tfidf_weight) for model in models])
```

Gbr. 16 Kode program untuk melakukan prediksi model deep neural network

```
array([[9.3439305e-01, 6.7912083e-04, 6.4927816e-02]], dtype=float32)
```

Gbr. 17 Hasil prediksi model deep neural network

Agar data hasil prediksi dapat dibaca dengan jelas, maka perlu dilakukan *postprocessing* dengan cara mengambil indeks *array* dengan nilai prediksi terbesar menggunakan fungsi *argmax* dari pustaka *Numpy*⁴, kemudian melakukan pemetaan dengan label asli untuk mendapatkan kelas sentimen. Nilai probabilitas tersebut juga dapat diambil sebagai *confidence score*.

⁴ <https://numpy.org/>

```
id2sentiment = {
    0: 'netral',
    1: 'negatif',
    2: 'positif'
}

predictions = np.array([model.predict(tfidf_weight) for model in models])
prediction_proba_averaged = np.mean(predictions, axis=0)
prediction_class_id = np.argmax(prediction_proba_averaged, axis=-1)

prediction_class = id2sentiment[prediction_class_id[0]]
confidence = prediction_proba_averaged[0][prediction_class_id]

print(f"prediction class id : {prediction_class_id[0]}")
print(f"prediction class : {prediction_class}")
print(f"confidence : {confidence[0]}")
```

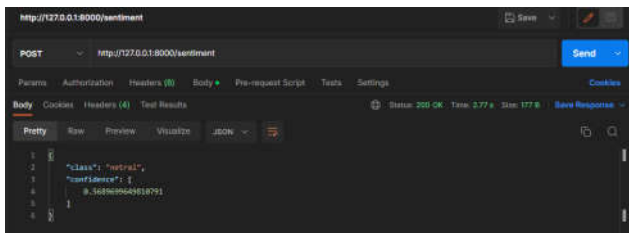
Gbr. 18 Kode program untuk melakukan postprocessing hasil prediksi deep neural network

Gbr. 19 menunjukkan hasil prediksi *deep neural network* setelah dilakukan *postprocessing* dimana hasil prediksi sentimennya adalah netral dengan probabilitas sebesar 56,8 %.

```
prediction class id : 0
prediction class : netral
confidence : 0.5689699649810791
```

Gbr. 19 Hasil prediksi model deep neural network setelah dilakukan postprocessing

Gbr. 20 menunjukkan hasil percobaan API menggunakan aplikasi Postman. Pembuatan API dilakukan menggunakan pustaka FastAPI. Masukan dari API ini berupa data teks yang dibungkus kedalam JSON sedangkan nilai keluarannya terdiri dari dua buah yaitu "class" yang berisi kelas sentimen yang diprediksi dan "confidence" yang berisi nilai probabilitas hasil prediksi.



Gbr. 20 bandingan performa model berdasarkan jumlah n-gram yang digunakan

Berikut ini adalah pembahasan setiap proses yang dilakukan dalam penelitian ini.

A. Dataset

Pengumpulan data dilakukan dengan teknik *web scrapping* menggunakan pustaka Beautiful Soup⁵ dari situs tiket.com. data ulasan hotel yang diambil adalah hotel-hotel dari daerah Surabaya, DKI Jakarta, Malang, Bali, Lombok, Yogyakarta, dan Medan. Total data yang berhasil diperoleh yaitu 1072 kalimat ulasan dan bentuk data dapat dilihat pada Gbr. 21.

⁵ <https://www.crummy.com/software/BeautifulSoup/>

	review
0	Ok ok lah ya
1	Secara keseluruhan baik,hanya saja sarapan kur...
2	kamar luas banget, AC dingin, bathroom nya Oke...
3	nyaman.... next pasti nginep disini lagi 2 pe...
4	waktu masih dijalan pas keliatan gedungnya dar...

Gbr. 21 Dataset ulasan hotel hasil web scrapping

Sentimen dianggap positif bila semua aspek yang disebutkan dalam ulasan dinilai baik, sentimen dianggap negatif jika semua aspek yang disebutkan dalam ulasan dinilai buruk serta sentimen dianggap netral jika sebagian aspek yang disebutkan dalam ulasan dinilai baik dan sebagiannya lagi dinilai buruk. Bentuk dataset setelah dilakukan proses pemberian label akan tampak seperti yang ditunjukkan Gbr. 22.

	review	sentiment
0	Ok ok lah ya	positif
1	Secara keseluruhan baik,hanya saja sarapan kur...	netral
2	kamar luas banget, AC dingin, bathroom nya Oke...	positif
3	nyaman.... next pasti nginep disini lagi 2 pe...	positif
4	waktu masih dijalan pas keliatan gedungnya dar...	positif

Gbr. 22 Bentuk dataset setelah dilakukan pelabelan

Adapun distribusi data untuk masing-masing sentiment dapat dilihat pada Tabel I.

TABEL I
DISTRIBUSI DATA PADA SETIAP LABEL

Sentiment	Jumlah Data
Positif	625
Negatif	225
Netral	222

B. Prapemrosesan Data

Tahap prapemrosesan pada data teks ulasan adalah sebagai berikut:

1) *Menghapus kalimat non review*: Beberapa kalimat *non review* yang ditemukan pada data-data ulasan yaitu "people found this helpful. Report", "Apa review ini membantu? Laporkan", dan "Is this review helpful? Report". Kalimat-kalimat ini bukanlah bagian dari ulasan, melainkan adalah fungsi-fungsi untuk menyediakan interaksi antara sistem dengan pengguna.

2) *Menghapus angka numerik*: angka-angka numerik tidak terlalu memiliki makna dalam data ulasan karena hanya digunakan untuk menyatakan jam, nominal harga, jumlah hari, jumlah lantai, dan lain-lain. Beberapa contoh

kalimat yang mengandung angka numerik dalam dataset yaitu “nginap 2 malam”, “jam 9 keatas”, dan “sewa 75 rb”.

3) *Menghapus tanda baca*: tanda baca tidak terlalu memiliki makna dalam data ulasan karena hanya digunakan selayaknya penggunaan tanda baca pada umumnya dan beberapa ulasan terdapat tanda baca yang digunakan secara berlebihan, seperti pada kalimat “pelayanan ramah...rekomened lah.” dan “dikurangi yaa standarnya??”.

4) *Standarisasi kata*: beberapa kata dalam ulasan ditulis dalam bentuk kata singkatan dan kata gaul seperti “yg” untuk “yang”, “byk” untuk “banyak”, “gede” untuk “besar”, “nyampe”, dan “sampe” untuk “sampai”. Standarisasi kata dalam penelitian ini dilakukan dengan membuat terlebih dahulu daftar pemetaan kata non-baku dan baku, kemudian kata non-baku diubah kedalam bentuk bakunya sesuai dengan pemetaan yang telah dibuat. Gbr. 23 menunjukkan tabel pemetaan antara kata non-baku dan baku atau bentuk standar.

vocab	standard
abis	setelah
ad	ada
aja	saja
ampek	sampai
baguss	bagus
bbrp	beberapa
bfast	breakfast
bget	banget
bgs	bagus
bgt	banget
blm	belum
bs	bisa
bnyak	banyak

Gbr. 23 Tabel pemetaan kata non-baku dan kata bentuk bakunya

5) *Menghapus stopwords*: *stopwords* adalah kata-kata yang sering muncul dalam dokumen sehingga membuat dokumen-dokumen menjadi susah untuk dibedakan berdasarkan kontennya. Kata-kata yang digolongkan sebagai *stopwords* dalam penelitian ini yaitu “dan”, “yang”, “di”, “saya”, “untuk”, “dengan”, “nya”, “ke”, “karena”, “itu”, “ya”, “sih”, “kita”, “pada”, “atau”, “kali”, “pun”, “k”, “rb”, dan “ribu”.

Perbandingan antara data ulasan sebelum dipraproses dan setelah dipraproses ditunjukkan oleh Gbr. 24.

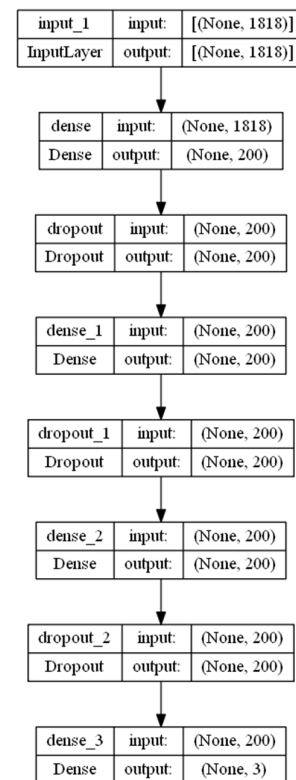
reviews	reviews preprocessed
<p>Ok ok lah ya</p> <p>Secara keseluruhan baik hanya saja sarapan kurang bervariasi dan rasanya kurang enak (setelah), air tidak jernih a.k.a keruh/kekuningan</p> <p>kamar luas banget, AC dingin, bathroom nya Oke ada bathtub nya, makanan Oke standart, kalau ada promo lagi worth it nginap sini</p> <p>nyaman... next pasti nginap disini lagi</p> <p>waktu masih dijalan pas keliatan gedungnya dari kejauhan tuh langsung kaget, kok dekil banget gedungnya? langsung lemes karna bawa anak2 liburan kesana, masa hotelnya gitu? kayak pas sudah masuk ke lobby, langsung berasa mewahnya... sft nya aja bagus hgt... kamarnya apalgi... kamar luasasss, kamar mandinya lg ada bathtub, pelayanan sangat baik, hotelnya bersihh, lokasi juga ditengah kota, dekat dgn turunan plaza... puasss banget pokoknya, krn km pandemi hotelnya gak kemana ternyata tetap luar biasa... akan menginap disini lagi klo ke surabaya</p> <p>Resepsionis kurang ramah dan kata kata nya tidak tepat (bikin tersinggung)</p> <p>Contoh : saya minta late check out, dan diberikan jam 13.00 karena saya juga cek in jam 22.00. Resepsionis nya blg "maksimal jam 13.00 out kalau kamar karena besok kita full booked, atau saya kemana charge yaa?"</p>	<p>oke oke lah</p> <p>secara keseluruhan baik hanya saja sarapan kurang bervariasi rasanya kurang enak setelah air tidak jernih a.k.a keruh/kekuningan</p> <p>kamar luas banget ac dingin bathroom oke ada bathtub makanan oke standart kalau ada promo lagi worth it nginap sini</p> <p>nyaman next pasti nginap disini lagi</p> <p>waktu masih dijalan pas keliatan gedungnya dari kejauhan tuh langsung kaget kok dekil banget gedungnya langsung lemes bawa anak2 liburan kesana masa hotelnya gitu kayak pas sudah masuk lobby langsung berasa mewahnya sft saja bagus banget kamarnya apalgi kamar luasasss kamar mandinya juga ada bathtub pelayanan sangat baik hotelnya bersihh lokasi juga ditengah kota dekat turunan plaza puasss banget pokoknya krn pandemi hotelnya tidak kemana ternyata tetap luar biasa akan menginap disini lagi kalau surabaya</p> <p>resepionis kurang ramah kata kata tidak tepat bikin tersinggung contoh minta late check out diberikan jam 13.00 karena saya cek in jam 22.00 kamar karena besok full booked karena charge</p>

Gbr. 24 Perbandingan data ulasan sebelum dipraproses dan setelah dipraproses

Sedangkan pada data label dilakukan pemetaan antara nama kelas sentimen dengan id yang berupa integer. Sentimen netral akan diubah ke angka 0, sentimen positif akan diubah ke angka 1, dan sentimen negatif akan diubah ke angka 2.

C. Training Model dan Evaluasi

Model yang digunakan dalam penelitian ini adalah *deep neural network* yang dibuat menggunakan pustaka Keras⁶ dan Tensorflow⁷. Arsitektur *deep neural network* yang menjadi *baseline* dari penelitian ini terdiri dari tiga buah *hidden layers* dengan dengan jumlah 200 neuron pada setiap *layer*nya. Pada setiap *hidden layer*, terdapat *dropout layer* dengan *dropout rate* sebesar 0.2. Arsitektur *baseline* dari model *deep neural network* dalam penelitian ini ditunjukkan pada Gbr. 25.



Gbr. 25 Model baseline deep neural network

Model ini dilatih menggunakan 857 data dan dievaluasi menggunakan 215 data. Adam digunakan sebagai metode optimasi dengan nilai beta 1 dan 2 masing-masing 0,001 dan 0,999 serta nilai *epsilon* 1e-8. Model dilatih menggunakan laptop dengan spesifikasi Intel Core i5 7200U (4 CPUs) 2.50 GHz dan GPU NVIDIA GeForce 930MX 2 GB. Proses pelatihan model dilakukan menggunakan GPU. Kemudian dilakukan evaluasi menggunakan pustaka Scikit-learn. Hasil

⁶ <https://keras.io/>

⁷ <https://www.tensorflow.org/>

evaluasi performa model *baseline* dengan kombinasi n-gram 1 dan nilai *learning rate* 0,001 ditunjukkan pada Tabel II.

TABEL II
PERFORMA BASELINE MODEL

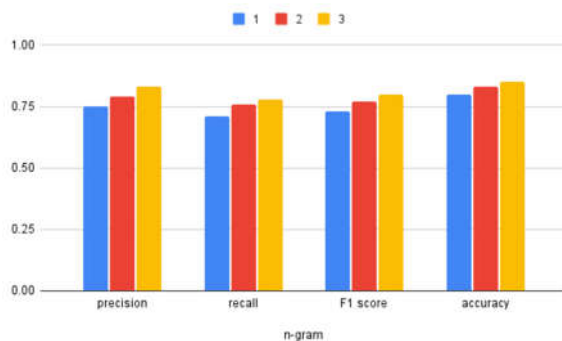
Precision	Recall	F1 Score	Accuracy
0,75	0,71	0,73	0,80

D. Hyperparameter Tuning

Pada kombinasi n-gram, nilai n-gram 1 memiliki performa terendah sedangkan nilai n-gram 3 memiliki performa terbaik diantara semuanya.

TABEL III
PERBANDINGAN PERFORMA MODEL

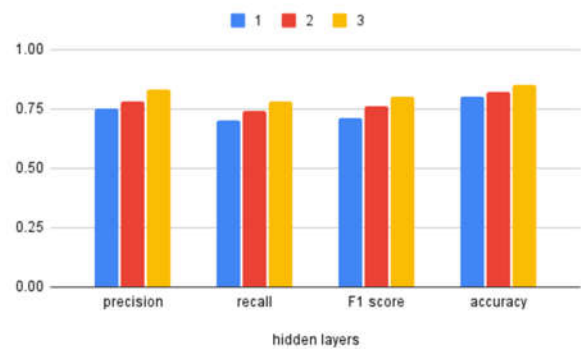
N-gram	Precision	Recall	F1 Score	Accuracy
1	0,75	0,71	0,73	0,80
2	0,79	0,76	0,77	0,83
3	0,83	0,78	0,80	0,85



Gbr. 26 Perbandingan performa model berdasarkan jumlah n-gram yang digunakan

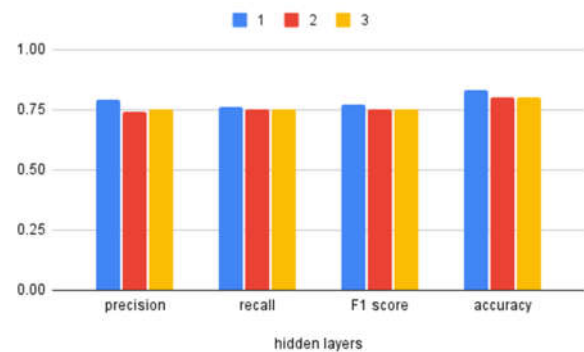
Oleh karena kombinasi n-gram 3 menghasilkan performa yang terbaik, maka pada percobaan kombinasi *hidden layers* dan *learning rate* dilakukan menggunakan nilai n-gram 3.

Pada kombinasi *hidden layers* dan *learning rate*, jumlah *hidden layers* 3 memiliki performa terbaik jika model dilatih dengan *learning rate* 0,001, seperti yang ditunjukkan pada Gbr. 27.



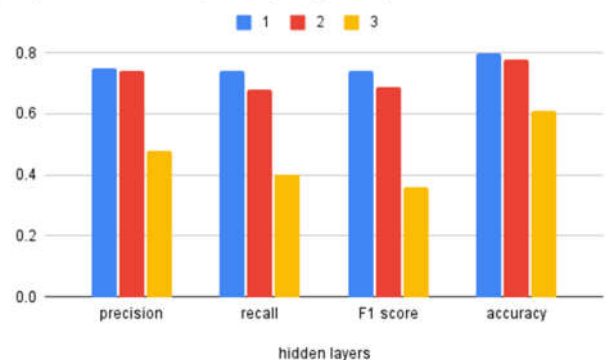
Gbr. 27 Perbandingan performa model berdasarkan jumlah hidden layers dengan learning rate 0,001

Sedangkan dari Gbr. 28 dapat dilihat bahwa pada nilai *learning rate* 0,01, model dengan jumlah *hidden layers* 1 memiliki performa terbaik.



Gbr. 28 Perbandingan performa model berdasarkan jumlah hidden layers dengan learning rate 0,01

Pada *learning rate* 0,1, performa model terbaik diraih oleh model dengan 1 *hidden layer*. Performa model cenderung turun jika jumlah *hidden layers*-nya diperbanyak.



Gbr. 29 Perbandingan performa model berdasarkan jumlah hidden layers dengan learning rate 0,1

Jika semua kombinasi jumlah *hidden layer* dan nilai *learning rate* dibandingkan, maka kombinasi yang menghasilkan performa terbaik yaitu 3 *hidden layer* dan *learning rate* 0,001 seperti yang ditunjukkan pada Tabel IV.

TABEL IV
PERBANDINGAN PERFORMA MODEL

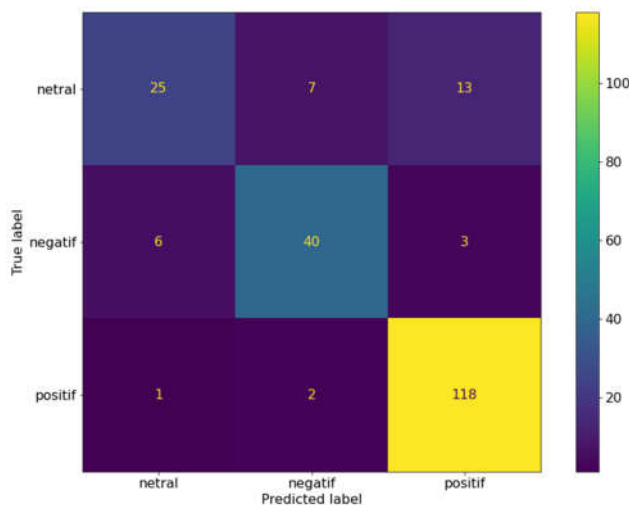
Hidden Layers	Learning Rate	Precision	Recall	F1 Score	Accuracy
1	0,001	0,75	0,70	0,71	0,80
2		0,78	0,74	0,76	0,82
3		0,83	0,78	0,80	0,85
1	0,01	0,79	0,76	0,77	0,83
2		0,74	0,75	0,75	0,80
3		0,75	0,75	0,75	0,80
1	0,1	0,75	0,74	0,74	0,80
2		0,74	0,68	0,69	0,78
3		0,48	0,40	0,36	0,61

Analisa lebih lanjut terhadap model terbaik dilakukan untuk melihat kemampuan model dalam memprediksi masing-masing sentimen. Dari tabel V dapat dilihat bahwa sentimen netral memiliki performa yang terburuk daripada sentimen positif dan negatif meskipun data latih untuk sentimen netral dan sentimen negatif memiliki jumlah yang hampir sama.

TABEL V
PERBANDINGAN PERFORMA ANTAR LABEL

	Precision	Recall	F1 Score	Total data
Positif	0,88	0,98	0,93	121
Negatif	0,82	0,82	0,82	49
Netral	0,78	0,56	0,65	45

Pada Gbr. 30 dapat dilihat bahwa sentimen netral banyak yang mengalami kesalahan klasifikasi sebagai sentimen positif. Sedangkan untuk sentimen negatif, kesalahan klasifikasi terbanyak terletak pada sentimen netral. Hal ini menunjukkan bahwa model masih mengalami kesulitan dalam mengenali sentimen netral.



Gbr. 30 Confusion matrix pada data testing

IV. KESIMPULAN

Dalam penelitian ini, dilakukan percobaan mengenai penggunaan metode *supervised machine learning* untuk menyelesaikan permasalahan analisa sentimen pada data ulasan hotel. Metode *deep neural network* dengan TF-IDF sebagai teknik ekstraksi fitur terbukti cukup mampu dalam mengatasi permasalahan analisa sentimen pada data ulasan hotel, meskipun masih mengalami kesulitan dalam mengenali sentimen netral. Selain itu, penelitian ini juga menemukan bahwa menentukan jumlah n-gram, jumlah *hidden layers*, dan nilai *learning rate* yang tepat dapat meningkatkan performa model *deep neural network*. Dalam penelitian ini, kombinasi *hyperparameter* terbaik yaitu n-gram 3, 3 *hidden layers*, dan nilai *learning rate* 0,001 dimana kombinasi tersebut dapat meningkatkan nilai f1 dan akurasi dari yang semula 0,73 dan 0,80 menjadi 0,80 dan 0,85.

V. SARAN

Kekurangan utama dari penelitian ini adalah performa model yang kurang begitu baik dalam memprediksi sentimen netral. Hal ini dikarenakan struktur kalimat ulasan netral memiliki kombinasi aspek positif dan negatif sehingga membuatnya lebih kompleks daripada kalimat ulasan positif dan negatif. Metode TF-IDF tidak mampu menangkap makna dalam kalimat dikarenakan sistem pembobotan yang dilakukan hanya berdasarkan frekuensi kemunculan suatu *term*. Sedangkan *deep neural network* memproses data secara terpisah sehingga hubungan antar kata atau token tidak dapat ditangkap dengan baik. Oleh karena itu, penelitian ini dapat dikembangkan lebih lanjut dengan menggunakan metode *word embedding* sebagai ekstraksi fitur. Sedangkan, agar hubungan antar kata dapat ditangkap dengan cukup baik, bisa digunakan model dari jenis RNN seperti LSTM atau *Bidirectional LSTM*, hal ini dikarenakan arsitektur RNN memproses data secara sekuensial.

UCAPAN TERIMA KASIH

Rasa terima kasih penulis sampaikan kepada Allah SWT atas berkat dan nikmat yang diberikan sehingga penelitian ini dapat diselesaikan dengan baik. Kemudian, kepada bapak Suyatno dan ibu Sri Kustiarini selaku orang tua penulis atas doa dan dukungannya, serta ibu Anita Qoiriah, S.Kom., M.T. selaku dosen pembimbing atas arahan dan bimbingannya. Rasa terima kasih tak lupa penulis ucapkan juga pada teman-teman Jurusan Teknik Informatika angkatan 2018 atas bantuan dan dukungannya.

REFERENSI

- [1] Fadhlullah, Maskurdin, Sarip Utoyo, Andi Setiawan, Desy Apresiyanti, Ahmad Saifullah, Vera Citra Wulandari, dan Khairul Amri. 2020. Statistik Hotel dan Akomodasi Lainnya di Indonesia 2020. Jakarta: Badan Pusat Statistik.
- [2] M. Geetha, Pratap Singha, Sumeda Sinha. 2017. "Relationship between customer sentiment and online customer ratings for hotels – An empirical analysis". *Tourism Management*. Hal. 43 – 54.
- [3] Rizka Putri Nawangsari, Retno Kusumaningrum, Adi Wibowo. 2019. "Word2Vec for Indonesian Sentiment Analysis towards Hotel Reviews: An Evaluation Study". 4th International Conference on Computer Science and Computational Intelligence. Hal. 360 – 366.

- [4] Vincent C.S Heung, Deniz Kucukusta, Erdogan Ekiz. 2010. "Evaluation of Guest Feedback Channels in China Hotels". Journal of China Tourism Research, 6. Hal. 296 – 309.
- [5] Helen Mavis Dah, Arnold Dumenya. 2016. "Investigating Customer Feedback Channel's in the Hotel Industry: the Case of Ho - Ghana". European Scientific Journal, Vol. 12, No. 26. Hal. 353 – 369.
- [6] Marouane Birjali, Mohammad Kasri, Abderarrahim Beni-Hssane. 2021. "A Comprehensive Survey on Sentiment Analysis: Approaches, Challenges, and Trends". Knowledge-Based System.
- [7] Yi Luo, Xiaowei Xu. 2021. "Comparative study of deep learning models for analyzing online restaurant reviews in the era of the COVID-19 pandemic". International Journal of Hospitality Management
- [8] Murat Demircan, Adem Seller, Fatih Abut, Mehmet Fatih Akay. 2021. "Developing Turkish sentiment analysis models using machine learning and ecommerce data". Interational Journal of Cognitive Computing in Engineering. Hal. 202 – 207.
- [9] Yash Indulkar, Abhijit Patil. 2020. "Sentiment Analysis of Uber & Ola using Deep Learning". Proceedings of the International Conference on Smart Electronics and Communication. Hal. 24 – 27.
- [10] Putra Fissabil Muhammad, Retno Kusumaningrum, Adi Wibowo. 2021. "Sentiment Analysis Using Word2vec And Long Short-Term Memory (LSTM) For Indonesian Hotel Reviews". 5th International Conference on Computer Science and Computational Intelligence 2020. Hal. 728 – 735.
- [11] Maulana Aziz Assuja, Saniati. 2016. "Analisis Sentimen Tweet Menggunakan Backpropagation Neural Network". Jurnal TEKNOINFO, Vol. 10. Hal. 23 – 28.
- [12] Danilo Dess, Rim Helaoui, Vivek Kumar, Diego Reforgiato Recupero, Daniele Riboni. 2020. "TF-IDF vs Word Embeddings for Morbidity Identification in Clinical Notes: An Initial Study". CEUR Proceedings of the First Workshop on Smart Personal Health Interfaces co-located with 25th International Conference on Intelligent User Interfaces. Hal. 1 - 12.
- [13] Aurélien Géron, *Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow, 2nd Edition*, Sebastopol, United States of America: O'Reilly Media, 2019.
- [14] Syopiansyah Jaya Putra, Muhamad Nur Gunawan, Agung Suryatno. 2016. "Tokenization and N-gram for Indexing Indonesian Translation of the Quran". 6th International Conference on Information and Communication Technology (ICoICT).
- [15] Peerapon Vateekul, Thanabhat Koomsubha. 2016. "A Study of Sentiment Analysis Using Deep Learning Techniques on Thai Twitter Data". 13th International Joint Conference on Computer Science and Software Engineering (JCSSE).
- [16] Changyong Yu, Xin Qi, Haiato Ma, Cuirong Wang, Yuhai Zhao. 2020. "LLR: Learning learning rates by LSTM for training neural networks". Neurocomputing. Hal. 41 – 50.
- [17] Andreas C. Müller, Sarah Guido, *Introduction to Machine Learning with Python: A Guide for Data Scientists*, Sebastopol, United States of America: O'Reilly Media, 2017.
- [18] Mark Masse, *REST API Design Rulebook*, Sebastopol, United States of America: O'Reilly Media, 2012.
- [19] M.A. Ganaia, Minghui Hu, A.K. Malik, M. Tanveer, P.N. Suganthan. 2022. "Ensemble deep learning: A review". Engineering Applications of Artificial Intelligence. Vol. 115.