Analisis Sentimen Pengguna Platform Belajar Online *Coursera* menggunakan *Random Forest* dengan Metode Ekstraksi Fitur *Word2vec*

Muhammad Jazaal Aufa[1], Anita Qoiriah[2]
[1,2] Jurusan Teknik Informatika, Fakultas Teknik, Universitas Negeri Surabaya
[1]muhammadjazaal.18051204026@mhs.unesa.ac.id
[2]anitaqoiriah@unesa.ac.id

***Abstrak*— Pembelajaran *daring* mengharuskan para pelajar dan mahasiswa menggunakan platform belajar online sebagai penunjang belajar di luar kelas, menyebabkan penggunanya meningkat, salah satunya *Coursera*. Berdasarkan grafik data *review* selama 2015-2020, pengguna *Coursera* meningkat drastis sebesar 256% saat awal pembelajaran *daring*. Namun menurun setelahnya sebesar 24%, sehingga diperlukan sistem analisis sentimen untuk mengetahui polaritas pengguna kursus saat itu secara otomatis. Algoritma klasifikasi yang efisien untuk analisis sentimen adalah *Random Forest*. Pada kasus *text classification* jumlah fitur adalah jumlah kata, kumpulan teks *review* yang mengandung banyak kata menghasilkan jumlah fitur yang banyak pula, maka waktu yang dibutuhkan untuk membangun model menggunakan *Random Forest* akan sangat lama, sehingga kurang efektif. Dibutuhkan ekstraksi yang mampu mereduksi jumlah fitur, seperti *Word2vec*. Menggunakan *Word2vec* sebagai ekstraksi fitur bertujuan agar sistem tidak hanya efisien tapi juga efektif untuk analisis sentimen, karena mampu mendeteksi semantik antar kata. Dari hasil pengujian menggunakan proporsi data berbeda dan parameter *Word2vec* acak, menunjukkan akurasi terbaik pada proporsi data 80:20 dengan ukuran fitur 100 dimensi dan *window* 5, dengan akurasi *train score* dan *test score* sebesar 92,71% dan 91,42%. Selain menghasilkan peforma paling baik, kombinasi *Random Forest* dengan *Word2vec* membutuhkan waktu *training model* yang 20 kali lebih cepat dibanding *Random Forest* dengan TF dan *Random Forest* dengan TF-IDF.**

***Kata Kunci*— Platfrom Belajar Online, *Coursera*, Analisis Sentimen, *Random Forest*, *Word2vec*.**

I. PENDAHULUAN

Penetapan kebijakan pembelajaran *daring* mengharuskan para pelajar menggunakan platform digital sebagai sarana komunikasi satu sama lain. Kewajiban menggunakan platform digital, salah satunya platform belajar online menciptakan sebuah kebiasaan baru dimasyarakat khususnya mahasiswa. Berdasarkan *Startup Report 2021*, dari hasil survei tingkat *awareness* masyarakat terhadap aplikasi startup, aplikasi *edutech* menjadi peringkat lima sebagai aplikasi paling popular di masyarakat, disebabkan oleh kebutuhan terhadap aplikasi sebagai penunjang pendidikan selama pandemi, menyebabkan jumlah penggunanya meningkat, salah satunya *Coursera*.

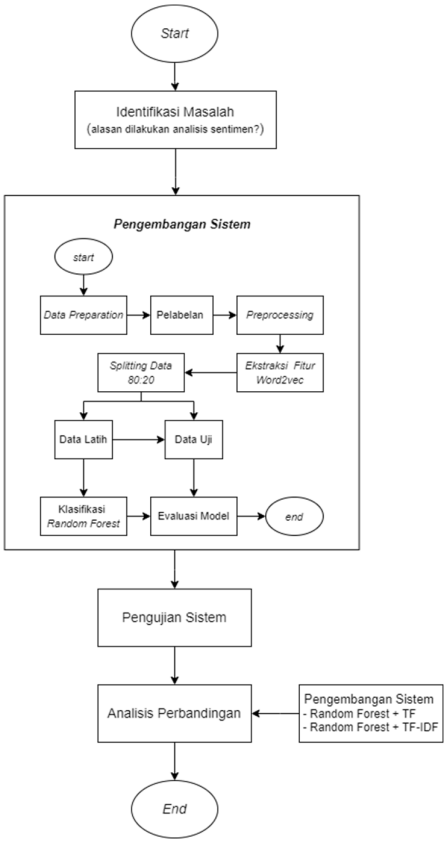
Berdasarkan data *review* pengguna selama 2015-2020 [1], *Coursera* sebagai salah satu platform belajar online yang menyediakan ribuan kursus dari universitas dan perusahaan ternama di dunia, mengalami peningkatan drastis sebesar 256% saat 03 Maret 2020 – 11 Mei 2020 atau saat awal pembelajaran *daring*. *Coursera* sebagai platform belajar online yang laris saat itu seharusnya memastikan agar pengguna tidak beralih (*churn*). Namun pada periode 06 Juni 2020- 26 Juli 2020 dan seterusnya pengguna *Coursera* menurun sebesar 24%. Sehingga perlu dilakukan analisis sentimen untuk mengetahui polaritas pengguna kursus *Coursera* saat itu berdasarkan *review*.

Analisis sentimen adalah proses menganalisis teks digital untuk menentukan apakah teks mengarah ke sentimen positif, negatif, atau netral [2]. Cara pengembangan sistem analisis sentimen umumnya diawali tahap pra-pemrosesan, yang meliputi; pembersihan teks dan *filtering* kata, kemudian pemberian skor pada setiap kata yang menggambarkan sentimen teks, dan terakhir adalah tahap pengklasifikasian teks sentimen. Algoritma yang efisien untuk pengklasifikasian adalah *Random Forest*. Berdasarkan penelitian dilakukan oleh Evita Fitri *et al* (2020), yang melakukan perbandingan antara algoritma klasifikasi *Naive Bayes*, *Random Forest* dan *Support Vector Machine* (SVM) untuk kasus analisis sentimen aplikasi Ruangguru berdasarkan *review* pengguna melalui *Google Play Store*, menunjukkan nilai akurasi tertinggi untuk analisis sentimen menggunakan *Random Forest* sebesar 97,16% dengan nilai *Area Under Curve* (AUC) sebesar 0,996 [3]. Namun waktu yang dibutuhkan *training model* menggunakan *Random Forest* membutuhkan waktu yang cukup lama. Berdasarkan penelitian yang dilakukan oleh Christanto & Setiabudi (2020), tentang penerapan *Random Forest* untuk mendeteksi spam pada *e-mail* menjelaskan bahwa waktu yang dibutuhkan untuk *training model* membutuhkan waktu 80 kali lebih lama dari *Naïve Bayes* [4], penyebabnya algoritma melatih model sebanyak fitur atau kolom. Pada kasus *text classification* setiap kata dalam teks menjadi fitur, jumlah teks *review* yang banyak tentu menghasilkan jumlah fitur yang banyak pula menyebabkan proses *training model* sangat lama.

Proses merubah teks menjadi fitur dinamakan ekstraksi fitur, dimana teks yang sifatnya *unstructured data* tidak dapat langsung diproses oleh algoritma klasifikasi, sehingga perlu dilakukan proses mengubah data tidak terstruktur menjadi data terstruktur atau tabular [5]. Proses ekstraksi fitur sama seperti proses pemberian skor kata berdasarkan jumlah frekuensi kata yang muncul dalam teks disebut *term frequency* (TF). Selain itu, terdapat metode yang menentukan nilai bobot kata berdasarkan frekuensi kemunculan dalam teks (*term frequency*), sekaligus menghitung frekuensinya terhadap keseluruhan dokumen (*inverse document frequency*) yang umum untuk kasus analisis sentimen yaitu TF-IDF. Metode TF-IDF dapat dikatakan lebih baik dari TF karena mampu menghasilkan fitur unik yang mengartikan pesan sebuah teks [6]. Namun dimensi fitur yang dihasilkan metode TF-IDF masih sangat besar bahkan mencapai ribuan, sehingga diperlukan proses ekstraksi fitur yang dapat mereduksi jumlah fitur yang ada.

Word Embedding merupakan metode *word representations* berupa numerik yang mampu mengurangi dimensi fitur [7], salah satu algoritmanya adalah *Word2vec*. Dimensi vektor dapat disesuaikan dengan kebutuhan, umumnya berukuran 300 dimensi, sehingga waktu yang dibutuhkan untuk melatih model dengan menggunakan algoritma *Random Forest* akan lebih cepat. Berbeda dengan metode ekstraksi lainnya yang kurang dapat memahami konteks, representasi vektor kata pada *Word2vec* juga mampu mengenali hubungan semantik antar kata berdasarkan nilai vektornya, apabila mesin dapat mempelajari hubungan semantik maka akan memudahkan untuk melakukan klasifikasi, sehingga sistem tidak hanya efisien tapi juga efektif untuk melakukan analisis sentimen.

Penelitian tentang analisis sentimen menggunakan *Random Forest* dan *Word2Vec* pernah dilakukan sebelumnya, pada penelitian yang dilakukan Siti Khomsah (2021), untuk kasus analisa sentimen komentar *Youtube* saat debat Capres menggunakan *Random Forest* dan *Word2vec*, sistem yang dibangun menghasilkan model terbaik dengan hasil akurasi antara 90,01% - 91% dengan hasil pengujian mencapai angka 89% [8]. Selain berbeda secara studi kasus, pada penelitian tersebut tidak menunjukkan bukti bahwa hasil akurasi analisis sentimen menggunakan *Random Forest* dengan *Word2vec* memang lebih baik dibanding *Random Forest* dengan TF dan *Random Forest* dengan TF-IDF, sehingga pada penelitian ini akan dilakukan perbandingan hasil akurasi dan *f1-score* untuk analisis sentimen antara kombinasi keduannya dengan hasil akurasi dan *f1-score Random Forest* dan *Word2vec*, untuk membuktikan efektivitas *Word2vec* sebagai ekstraksi fitur untuk sistem analisis sentimen menggunakan *Random Forest*. Tujuannya sebagai perbaikan untuk penelitian sebelumnya.



GBR 1. DIAGRAM ALUR PENELITIAN

II. METODOLOGI PENELITIAN

Metodologi penelitian adalah proses yang menjelaskan tahapan atau alur yang dilakukan selama penelitian agar output yang dihasilkan dapat memenuhi tujuan. Adapun alur metodologi dalam penelitian ini, seperti Gbr 1. berikut.

A. Identifikasi Masalah

Tahap awal pada penelitian adalah analisis masalah Berdasarkan beberapa studi literatur, menghasilkan beberapa statemen yang membahas kekurangan *Random Forest,* yaitu untuk memperoleh model yang efisien, diperlukan *resources* yang cukup banyak, sehingga waktu yang dibutuhkan proses pembelajaran akan lama. Suatu hal yang kurang tepat apabila menggunakan *Random Forest* untuk kasus klasifikasi teks yang umumnya menjadikan kata sebagai fitur atau kolom, karena apabila menggunakan *Random Forest* proses pembelajaran yang dibutuhkan akan semakin lama. Ekstraksi fitur merupakan tahapan wajib sebelum proses klasifikasi [5], sebagai tahapan untuk menjadikan teks sebagai fitur. Jumlah fitur yang rendah dapat mempercepat proses pembelajaran, istilahnya disebut *dimensionality reduction,* algoritma yang telah menerapkan *dimensionality reduction* adalah *Word2vec*.

Berdasarkan Gbr.1, pada penelitian ini memiliki beberapa tahap yang harus dipenuhi sebelum untuk mencapai tujuan penelitian yang diharapkan.

Adapun tahapan penelitian antara lain:

B. Pengembangan Sistem

Tapan kedua merupakan tahapan inti pada penelitian ini, yaitu pengembangan sistem. Alur pengembangan sistem diawali dengan persiapan data, pelabelan data, *preprocessing,* ekstraksi fitur *Word2vec,* pemodelan *Random Forest* dan evaluasi model klasifikasi.

1. Persiapan Data

   Tahap persiapan data memiliki dua bagian antara lain, pengumpulan dan pemahaman data

   a) Pengumpulan Data

   Penelitian ini menggunakan data sekunder atau data yang diperoleh oleh peneliti dari objek penelitian secara tidak langsung, data yang digunakan berasal dari website *data science* terkenal, yaitu *Kaggle* berjudul *Course Reviews on Coursera* yang memiliki 1,45 juta kolom, namun peneliti hanya menggunakan data *review* selama 08 Juni 2020 – 10 Oktober 2020 atau sampai akhir data yang berhasil di-*scraping.*

   b) Pemahaman Data

Tahap selanjutnya adalah pemahaman data. Pemahaman diperoleh dari proses mengamati atau eksplorasi data, biasanya disebut *Exploratory Data Analysis* (EDA). Proses EDA dilakukan untuk menyaring (*filter*) data *review* dengan rentang waktu tertentu, kemudian dieksplorasi untuk mengetahui persentase dan jumlah ulasan dari masing-masing rating sekaligus mengapus kolom yang tidak perlu.

2. Pelabelan Data

Tahap selanjutnya adalah pelabelan data. Tolok ukur pelabelan pada penelitian ini berdasarkan rating, yang ditunjukan pada Tabel 1 dengan pembagian kelas menjadi tiga kelas, diantaranya; untuk rating dengan nilai 1-3 memiliki makna pengguna kurang puas, diberikan label "negatif", untuk rating 4 memiliki makna pengguna cukup puas, diberikan label "netral", sedangkan rating 5 artinya pengguna puas, maka diberikan label "positif".

TABEL 1. TOLOK UKUR PELALBELAN DATA

Review	Rating	Kelas
Not as I expected. Sometimes there too much information which is not relevant	3	Negatif
Excellent course for beginners. However, the only con I found was a lot of week 3 content seem irrelevant,	4	Netral
The course is very well structured, with short videos that go straight to the point. The instructors are great, there is an offer that the platform gives for course introduction without having to pay.	5	Positif

Berdasarkan tolok ukur tersebut, apabila diperoleh jumlah data yang tidak seimbang maka perlu dilakukan proses penyeimbangan data. Data *imbalance* kurang baik untuk klasifikasi karena akan menciptakan model yang hanya mampu memprediksi kelas mayoritas, meskipun persentase akurasinya tinggi.

3. *Preprocessing*

Prepocessing merupakan proses menghilangkan informasi yang tidak diperlukan sekaligus bertujuan mengurangi dimensi input ke dalam model. Metode *preprocessing* yang digunakan pada penelitian ini, seperti: *case folding*, *revome punctuation (cleaning), stopword removal (filtering)*, *lemmatization* dan *tokenization*. Adapun penjelasan setiap metode pra-pemrosesan pada penelitian ini, sebegai berikut:

a) *Case Folding*

Proses pengubahan huruf yang tidak konsisten pada kata, fungsinya menyamaratakan huruf penggunaan huruf kapital menjadi huruf kecil (*lowercase*).

b) *Romove Punctuation*

Remove Punctuation bertujuan untuk, menghapus angka, tanda baca dan karakter aneh (*punctuatation*), seperti teks penuh emoji, pada data *tweets*. Dalam *cleaning* perlu diperhatikan bahwa tidak semua proses perlu diterapkan, seperti menghilangkan emoji karena itu sebenarnya dapat mewakili perasaan atau sentimen pengguna.

c) *Stopword Removal*

Stopword bertujuan memilah *(filtering*) kata-kata umum atau sering digunakan yang sebenarnya tidak bermakna kemudian menghapusnya, dalam bahasa Indonesia seperti: dan, atau, yang, dari, ke kemudian, untuk dan lain sebagainya.

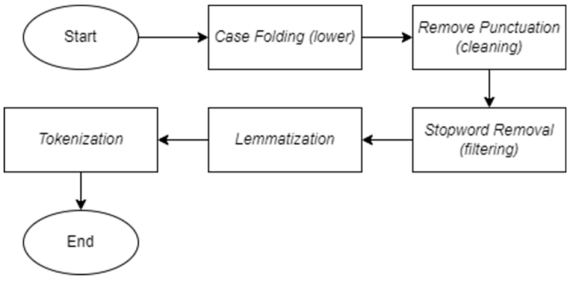
d) *Lemmatization*

Lemmatization adalah proses mereduksi variasi kata menjadi *lemma* atau akar kata. Dalam Bahasa Inggris kata *learning*, *learned*, semua akan diubah ke kata sebenarnya, yaitu *learn* dan lain sebagainya.

e) *Tokenization*

Tokenisasi adalah proses di mana teks akan dipecah menjadi susunan kata yang disebut token. Tokenisasi membantu dalam melakukan transformasi setiap kata menjadi angka.

Adapun alur tahap *preprocessing* pada penelitian ini ditujukkan pada Gbr 2. dibawah ini



GBR 2. ALUR TAHAPAN PREPROCESSING

Berikut ini merupakan gambaran setiap tahapan *peprocessing* yang dapat dilihat pada Tabel 2.

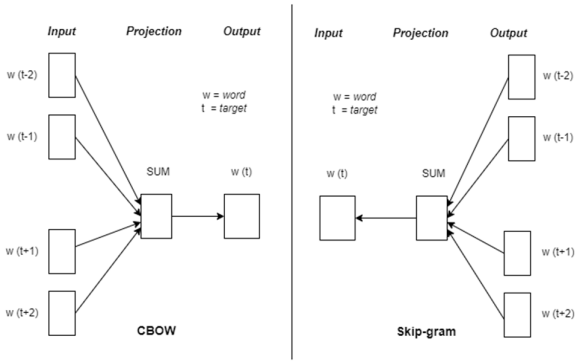
TABEL 2. GAMBARAN TAHAPAN PREPROCESSING

No	Tahapan	Sebelum	Sesudah
1	*Case Folding*	*I cant access the quiz 5 and beyond. Id really like to continue, and to complete the quizzes.*	*i cant access the quiz 5 and beyond. id really like to continue, and to complete the quizzes.*
2	*Remove Punctuation*	*i cant access the quiz 5 and beyond.*	*i cant access the quiz and beyond id*

		id really like to continue, and to complete the quizzes.	*really like to continue and to complete the quizzes*
3	*Stopword Removal*	*i cant access the quiz and beyond id really like to continue and to complete the quizzes*	*cant access quiz beyond like continue complete quizzes*
4	*Lemmatization*	*cant access quiz beyond like continue complete quizzes*	*cant access quiz beyond like continue complete quiz*
5	*Tokenization*	*cant access quiz beyond like continue complete quiz*	*can, access, quiz, beyond, like, continue, complete, quiz]*

4. Ekstraksi Fitur dengan *Word2Vec*

Ekstraksi fitur menggunakan teknik mengubah data berupa karakter (kata) menjadi sekumpulan angka (*vector*) atau vektor sebagai representasi dari kata. Teknik tersebut disebut *word embedding.* Salah satu metodenya adalah *Word2Vec*. Terdapat dua jenis arsitektur *Word2Vec*, pertama dengan memprediksi kata *target* menggunakan input atau kata-kata disekitar disebut *continuous bag of words* (CBOW), sebaliknya memprediksi kata-kata disekitar menggunakan input kata *target*, yang disebut Skip-gram. Adapun Gbr.3 di menunjukkan masing-masing arsitektur *Word2Vec*.



GBR 4. ARSITEKTUR *WORD2VEC*

Sumber: (Mikolov, 2013)

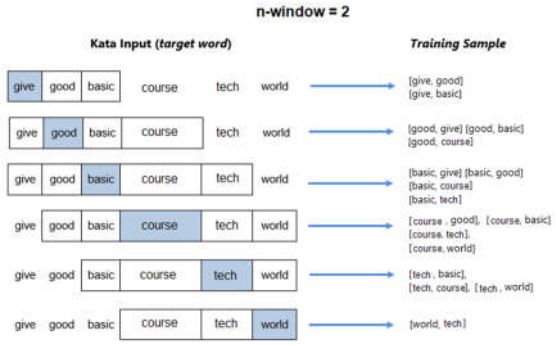
Penelitian kali ini menggunakan arsitektur Skip-gram, karena baik dalam memprediksi kata-kata yang jarang muncul daripada CBOW dalam kasus analisis sentimen [9]. Arsitektur *Word2Vec* mengadaptasi model arsitektur *neural* yang memiliki 3 lapisan antara lain: *input, projection* (*hidden layer)* dan *output* Pada lapisan input, menggunakan teknik *one-hot-encoder* dimana semua kolom akan bernilai 0, kecuali 1 yang artinya mewakili kata, sehingga membentuk vektor biner. Berikut merupakan ilustrasi input kata menggunakan *one-hot-encoder* yang ditunjukkan pada Tabel 3

Contoh Review: "give good basic course tech world"

TABEL 3
TEKNIK ONE-HOT ENCODING VEKTOR

Id	*give*	*good*	*basic*	*course*	*tech*	*world*
1	1	0	0	0	0	0
2	0	1	0	0	0	0
3	0	0	1	0	0	0
4	0	0	0	1	0	0
5	0	0	0	0	1	0
6	0	0	0	0	0	1

Vektor tersebut yang merepresentasikan sebuah kata, pada arsitektur skip-gram model diminta untuk memprediksi kata-kata sekitar ketika diberikan input satu kata agar dapat mempelajari semantik antar kata, kata-kata sekitar disebut *window,* yang banyaknya dapat ditentukan *n_window*. Adapun gambaran ketika jumlah *window=2*, yang ditunjukkan pada Gbr 4. berikut ini.



GBR 4. GAMBARAN WINDOW MODEL SKIP-GRAM

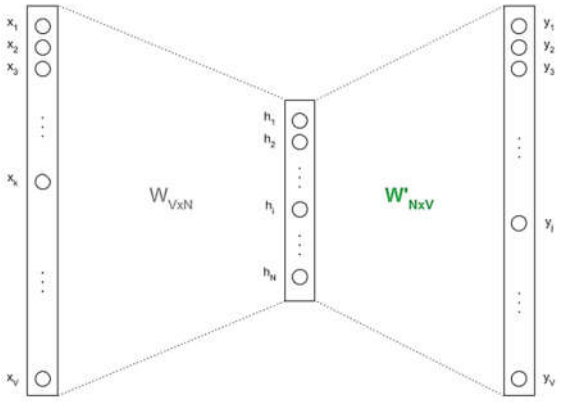
Jumlah *windows* akan menghasilkan beberapa pasangan kata yang akan dipelajari oleh model, disebut *training sample* pada *hidden layer* untuk kemudian diberikan pembobotan. Cara kerja pembobotan pada *hidden layer* dapat dilihat pada perhitungan matriks melalui fungsi (1) berikut.

$$\underset{1\,x\,V}{Input\ Vector} \quad \underset{W\ =\ V\,x\,N}{Weight} \quad \underset{1\,x\,h}{Hidden}$$

$$[0\ 0\ 1\ 0] \times \begin{bmatrix} a & b & c \\ d & e & f \\ g & h & i \\ j & k & l \end{bmatrix} = \ g\ h\ i] \ \ldots\ldots\ldots (1)$$

Pembototan (W) diperoleh dari hasil perkalian antara input vektor (V) dengan ukuran fitur sebanyak

(N) *neurons* yang digunakan. Bobot tersebut nantinya merepresentasikan hasil vektor setiap kata. Melihat cara kerja pembobotan, maka dalam *projection* (*hidden layer*) menerapkan *linear activation* yakni dengan menyalin baris yang sesuai dari matriks atau pembobotan yang diberikan. Cara yang sama, maka hubungan antara *hidden layer* dan lapisan output dapat digambarkan dengan pembobotan W' dengan perhitungan $N \times V$, yang artinya setiap kolom matriks W' mewakili vektor dari input vektor. Adapun cara kerja pembobotan kata pada *Word2Vec* ditunjukkan pada Gbr. 5.



GBR 5. CARA KERJA PEMBOBOTAN KATA WORD2VEC

Selanjutnya, agar model dapat mengenali hubungan antar kata, maka model harus menunjukkan probailitas tertinggi terhadapa kata-kata sekitar yang hendak diperdiksi. Pertama, pada *input* ke *hidden layer* akan dihitung mengggunakan pembobotan (*W)* melalui persamaan (2) berikut.

$$X \times W \text{ ..... (2)}$$

Keterangan:
X = input vektor
W = pembobotan (input-*hidden layer*)

Kemudian, untuk *hidden layer* ke *output* dilakukan perhitungan dengan pembobotan (W') persamaan (3).

$$h \times W' \text{ ..... (3)}$$

h = *hidden layer*
W = pembobotan (*hidden layer*-output)

Probabilitas tertinggi diperoleh menggunakan fungsi *softmax*. Namun banyaknya jumlah kata yang dihasilkan untuk setiap teks, maka menentukan probalilitas kata tidak dapat menggunakan fungsi *softmax* biasa. Pada algoritma *Word2vec* menggunakan fungsi *hierarchical softmax* sebagai alternatif, dimana probabilitas setiap kata dihitung melalui probabilitas sisi pada jalur *node* menggunakan *multi-layer binary tree.* Fungsi aktivasi *softmax* dapat dijelaskan pada pesamaan (4)

$$y_k = \mathrm{P}_r\,(kata_k|kata_{context}) = \frac{\exp(k)}{\sum_{n=1}^{v}\exp(n)} \text{ ..... (4)}$$

Keterangan:
y_k = nilai aktivasi *softmax*
$\mathrm{P_r}$ = mengitung probabilitas
$kata_k$ = kata target (*target word*)
$kata_{context}$ = kata konteks (*context word*)
$\exp(k)$ = eksponensial nilai vektor ke-k
$\exp(n)$ = eksponensial total nilai vektor
v = jumlah iterasi

Sebagai contoh, diambil dua *training sample* yaitu (*course, basic)* dan (*course, tech*). Model diminta memprediksi kata *basic* dan *tech* ketika diberikan *input* kata *course*, dengan asumsi ukuran fitur = 4, maka dimensi matriks untuk masing-masing bobot W dan W' adalah 6×4 dan 4×6 dengan nilai pembobotan sebagai berikut.

$$W = \begin{bmatrix} 0.4 & 0.1 & 0.6 & 0.3 \\ 0.2 & 0.8 & 0.5 & 0.7 \\ 0.3 & 0.5 & 0.8 & 0.6 \\ 0.2 & 0.1 & 0.3 & 0.5 \\ 0.1 & 0.7 & 0.6 & 0.4 \\ 0.1 & 0.2 & 0.7 & 0.8 \end{bmatrix}$$

$$W' = \begin{bmatrix} 0.3 & 0.4 & 0.6 & 0.1 & 0.8 & 0.2 \\ 0.7 & 0.3 & 0.2 & 0.1 & 0.6 & 0.5 \\ 0.1 & 0.3 & 0.5 & 0.4 & 0.6 & 0.2 \\ 0.5 & 0.2 & 0.7 & 0.9 & 0.1 & 0.1 \end{bmatrix}$$

Berdasarkan Tabel 3, input vector untuk kata *course* adalah $X = 0\ 0\ 0\ 1\ 0\ 0]^T$, sehingga masing-masing hasil pembobotan (W) untuk *input* ke *hidden layer* dan pembobotan (W') untuk *hidden layer* ke *output.*

$$\underset{(V \times 1)}{X} \quad \underset{(V \times N)}{W} \quad \underset{(N \times 1)}{h}$$

$$\begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \\ 0 \\ 0 \end{bmatrix} \times \begin{bmatrix} 0.4 & 0.1 & 0.6 & 0.3 \\ 0.2 & 0.8 & 0.5 & 0.7 \\ 0.3 & 0.5 & 0.8 & 0.6 \\ 0.2 & 0.1 & 0.3 & 0.5 \\ 0.1 & 0.7 & 0.6 & 0.4 \\ 0.1 & 0.2 & 0.7 & 0.8 \end{bmatrix}^T = \begin{bmatrix} 0.2 \\ 0.1 \\ 0.3 \\ 0.5 \end{bmatrix}$$

$$\underset{(N \times 1)}{h} \quad \underset{(V \times 1)}{W' \times h} \quad \underset{(V \times 1)}{y_{pred}}$$

$$\underset{(N \times V)}{W'}$$

$$\begin{bmatrix} 0.3 & 0.4 & 0.6 & 0.1 & 0.8 & 0.2 \\ 0.7 & 0.3 & 0.2 & 0.1 & 0.6 & 0.5 \\ 0.1 & 0.3 & 0.5 & 0.4 & 0.6 & 0.2 \\ 0.5 & 0.2 & 0.7 & 0.9 & 0.1 & 0.1 \end{bmatrix}^T \times \begin{bmatrix} 0.2 \\ 0.1 \\ 0.3 \\ 0.5 \end{bmatrix} = \begin{bmatrix} 0.414 \\ 0.3 \\ 0.6 \\ 0.64 \\ 0.45 \\ 0.2 \end{bmatrix} \rightarrow \begin{bmatrix} 0.1615 \\ 0.1440 \\ 0.1944 \\ \mathbf{0.2024} \\ 0.1674 \\ 0.1303 \end{bmatrix}$$

Kata dengan nilai probabilitas tertinggi diatas merupakan kata input atau *course* yang dimaksud. Pada contoh, kata-kata sekitar yang hendak diprediksi memiliki vektor $[0\ 0\ 1\ 0\ 0\ 0]^T$ dan $[0\ 0\ 0\ 0\ 1\ 0]^T$, vektor-vektor tersebut akan dikurangi dengan nilai probabilitas *softmax* untuk memperoleh nilai *loss/error*.

$$\underset{(V \times 1)}{y_{pred}}\ \underset{(V \times 1)}{y_{target\ w+1}}\ \underset{(V \times 1)}{loss_{w+1}}\ \underset{(V \times 1)}{y_{pred}}\ \underset{(V \times 1)}{y_{target\ w-1}}\ \underset{(V \times 1)}{loss_{w-1}}$$

$$\begin{bmatrix} 0 \\ 0 \\ 1 \\ 0 \\ 0 \\ 0 \end{bmatrix} - \begin{bmatrix} 0.1615 \\ 0.1440 \\ 0.1944 \\ 0.2024 \\ 0.1674 \\ 0.1303 \end{bmatrix} = \begin{bmatrix} -0.1615 \\ -0.1440 \\ 0.8056 \\ -0.2024 \\ -0.1674 \\ -0.1303 \end{bmatrix} \quad \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 1 \\ 0 \end{bmatrix} - \begin{bmatrix} 0.1615 \\ 0.1440 \\ 0.1944 \\ 0.2024 \\ 0.1674 \\ 0.1303 \end{bmatrix} = \begin{bmatrix} -0.1615 \\ -0.1440 \\ -0.1944 \\ -0.2024 \\ 0.8326 \\ -0.1303 \end{bmatrix}$$

$$\underset{(V \times 1)}{loss_{w+1}}\quad \underset{(V \times 1)}{loss_{w-1}}\quad \underset{(V \times 1)}{loss_{w-1}}$$

$$\begin{bmatrix} -0.1615 \\ -0.1440 \\ 0.8056 \\ -0.2024 \\ -0.1674 \\ -0.1303 \end{bmatrix} + \begin{bmatrix} -0.1615 \\ -0.1440 \\ -0.1944 \\ -0.2024 \\ 0.8326 \\ -0.1303 \end{bmatrix} = \begin{bmatrix} -0.323 \\ -0.288 \\ 0.611 \\ -0.405 \\ 0.662 \\ -0.26 \end{bmatrix}$$

Nilai *loss* berfungsi untuk memperbaruhi nilai *weight matrix* pada *input* dan *output*. Diperoleh hasil ekstraksi fitur *Word2Vec* dari teks *review* diatas pada Tabel 4, sebagai berikut.

TABEL 4
HASIL EKSTRAKSI FITUR WORD2VEC

Word	fitur_1	fitur_2	fitur_3	fitur_4
give	0.44158584	-0.15249345	-0.22609861	0.7774802
good	-0.01496343	0.31543693	0.03357472	0.56858397
basic	-0.39373228	-0.6296626	0.64077973	0.05164771
course	-0.76311743	-0.7635295	0.8247233	0.10079303
tech	-0.15249345	0.64077973	-0.01496343	-0.2260986
world	0.5004419	0.03357472	-0.47035068	-0.2098551

Hasil esktraksi fitur kemudian akan digunakan saat *training model* klasifikasi. Namun, perlu diketahui bahwa jumlah kata dalam setiap teks berbeda-beda, sehingga diperlukan tahap lanjutan dengan melakukan rata-rata vektor seluruh kata dalam teks yang mewakili vektor dari teks atau *average word vectors*. Vektor yang dihasilkan telah merepresentasikan vektor dari kalimat (*sentence vector*), secara fungsi matematis cara merubah vektor kata-kata menjadi vektor kalimat ditunjukkan pada persamaan (5).

$$\underset{}{W_1}\begin{bmatrix} W_{11} \\ W_{12} \\ \\ \\ W_{1n} \end{bmatrix} + \underset{}{W_2}\begin{bmatrix} W_{21} \\ W_{22} \\ \\ \\ W_{2n} \end{bmatrix} + \cdots + \underset{}{W_n}\begin{bmatrix} W_{n1} \\ W_{n2} \\ \\ \\ W_{nn} \end{bmatrix} = \underset{}{S}\begin{bmatrix} \frac{W_{11}+W_{21}+\ldots+W_{n1}}{n} \\ \\ \\ \frac{W_{1n}+W_{2n}+\ldots+W_{nn}}{n} \end{bmatrix} \quad \ldots\ldots (5)$$

Keterangan :

W_n = *word vectors* kata ke -n
W_{nn} = *word vectors* kata ke -n, index ke-n
S = *sentence vector*

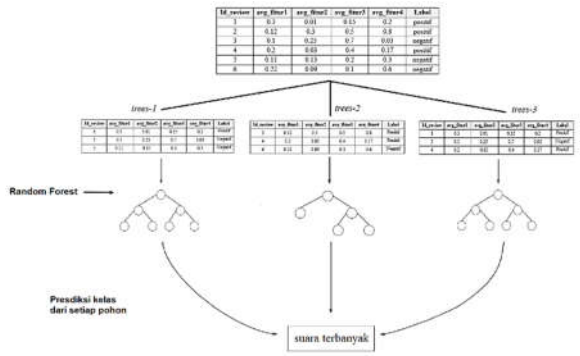
Berdasarkan Tabel 4, maka *sentence vector* untuk teks *review "give good basic course tech world"* ditunjukkan pada Tabel 5, sebagai berikut.

TABEL 5
SENTENCE VECTOR

Word	fitur_1	fitur_2	fitur_3	fitur_4
give	0.44158584	-0.15249345	-0.22609861	0.7774802
good	-0.01496343	0.31543693	0.03357472	0.56858397
basic	-0.39373228	-0.6296626	0.64077973	0.05164771
course	-0.76311743	-0.76352953	0.8247233	0.10079303
tech	-0.15249345	0.64077973	-0.01496343	-0.2260986
world	0.5004419	0.03357472	-0.47035068	-0.209855
average	**-0,17654415**	**-0,092649033**	**0,131277505**	**0,17709186**

5. Klasifikasi *Random Forest*

Pengklasifikasian data review dalam kelas sentimen merupakan tujuan dari penelitian ini, algoritma yang digunakan adalah *Random Forest.* Algoritma adalah salah satu implementasi metode *ansambel*. Metode *ensemble* merupakan teknik pembelajaran mesin yang menggabungkan beberapa model untuk menghasilkan satu model prediksi yang optimal [10]. Cara kerja algoritma *random forest* adalah membagi data ke beberapa pohon individu, yang diasumsikan terdapat pohon (*trees)* sebanyak *n* yang diambil secara *random* dari dataset, dengan nilai *n* dipastikan lebih kecil dataset menggunakan teknik *bootstrap aggregating*[11].



GBR 6. ILUSTRASI ALGORITMA RANDOM-FOREST

Sumber: (Khomsah, 2021)

Sebagai ilustrasi, terdapat teks *review* dalam bentuk vektor teks, dengan jumlah fitur sebanyak 4 (*avg_fitur1, avg_fitur2, avg_fitur3, avg_fitur4*) memiliki kelas atau label sebanyak dua kelas (positif dan negatif), dan jumlah *n_trees* sebanyak 3. Dari ketiga pohon, *trees-1* dan *trees-1* memperdiksi teks *review* pertama sebagai kelas positif, maka berdasarkan jumlah suara terbanyak, *review* pertama akan diidentifikasi sebagai kelas positif.

6. Evaluasi Model Klasifikasi

Evaluasi hasil dilakukan untuk mengetahui kinerja model klasifikasi. Cara untuk mengevaluasi dapat dilakukan dengan evaluasi matriks, *Confusion Matrix*. dengan empat representasi hasil proses klasifikasi yaitu; *True Positive* (TP), *True Negative* (TN), *False Positive* (FP), dan *False Negative* (FN), yang menunjukkan keakuratan, presisi dan *recall* serta *f1-score* model klasfikasi saat menghasilkan informasi yang diminta dengan informasi yang diperoleh, yang dalam kasus ini adalah memprediksi sentiment teks *review*.

Adapun masing-masing formula untuk mengitung akurasi, presisi, *recall* dan *f1-score*, ditunjukkan pada persamaan (6), (7), (8), (9) dan (10)

$$Accuracy_{2\times2} = \frac{TP+TN}{TP+TN+FP+FN} \quad \text{..... (6)}$$

$$Accuracy_{3\times3} = \frac{AA+BB+CC}{AA+AB+AC+BA+BB+BC+CA+CB+CC} \quad \text{..... (7)}$$

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

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

C. Pengujian Sistem dan Analisis Perbandingan

harapan penelitian ini adalah memperoleh pengetahuan terkait efektivitas *Word2vec* sebagai ekstraksi fitur dalam melakukan analisis sentimen menggunakan *Random Forest*, dilakukan pengujian untuk membandingkan performa model antara *Random Forest + TF, Random Forest* + TF-IDF dengan *Random Forest + Word2vec* berdasarkan akurasi dan nilai rata-rata *f1-score* untuk masing-masing kelasnya.

III. HASIL DAN PEMBAHASAN

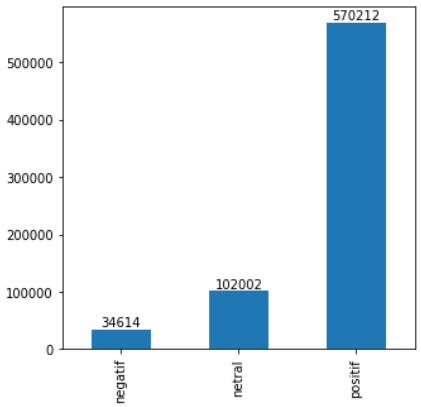
A. Data

Data yang diolah merupakan data *review* platform belajar online Coursera saat terjadi penurunan akes pengguna. Hasil *filtering* diperoleh sebanyak 706.828 data dengan 5 kolom, dengan jumlah dan presentase setiap rating dintunjukkan pada Tabel 6 berikut ini.

TABEL 5
JUMLAH DAN PRESESNTASE DATA

No	Rating	Jumlah	Persentase
1	5	570.212	80,67%
2	4	102.002	14,44%
3	3	19.782	2,79%
4	2	6.792	0,96%
5	1	8.040	1,13%

Selanjutnya dilakukan pelabelan berdasarkan tolok ukur nilai rating seperti pada Tabel 1, sehingga jumlah label untuk masing-masing kelas adalah, 570.212 data untuk label positif, 102.002 untuk label netral dan 34.614 untuk label negatif yang divisualiasasikan pada Gbr 7, pada proses pelabelan sekaligus menghapus kolom yang tidak diperlukan.



GBR 6. HASIL PELABELAN DATA

Hasil pelabelan setiap kelas dapat divisualisasikan juga menggunakan *Wordcloud*.

Wordcloud merupakan gambar yang menampilkan daftar kata-kata dalam sebuah teks, semakin banyak jumlah sebuah

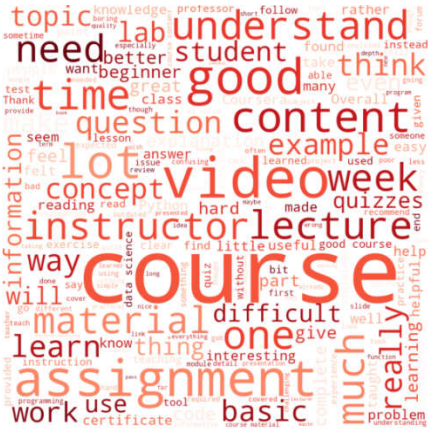
kata dalam teks maka semakin besar ukuran kata tersebut dalam gambar. Adapun *Wordcloud* kelas positif, netral dan negatif ditunjukkan pada beberapa gambar dibawah ini.

GBR 7. WORDCLOUD SENTIMEN POSITIF

Gbr 7, menggambarkan *wordcloud* kelas sentimen positif, gambar tersebut memuat banyak kata yang bernuansa positif seperti *'great', 'excellent', 'easy', 'good'* dan lain sebagainya.

GBR 8. WORDCLOUD SENTIMEN NETRAL

Wordcloud diatas Gbr 8, adalah *wordcloud* kelas sentimen netral memuat daftar kata yang merujuk ke saran, karena menampilkan beberapa kata positif sekaligus bernuansa negatif seperti; *'need', 'better', 'help', difficult*" dan lain-lain.



GBR 9. WORDCLOUD SENTIMEN NEGATIF

Wordcloud kelas sentimen negatif ditunjukkan Gbr 9, yang memuat beberapa kata yang bernuansa negatif seputar keluhan pengguna. Kata tersebut seperti ; *'difficult', 'poor', 'boring', 'problem',* dan lain sebagainya.

Data yang sudah diberi label kemudian dilanjutkan ke tahapan *preprocessing* seperti alur tahapan yang ditunjukkan Gbr 2. Berikut hasil tahap *preprocessing* menggunakan salah satu data *review* yang dilakukan secara manual, pada Tabel 6.

TABEL 5
PREPROCESSING DATA MANUAL

ID	*Reviews*	Label
000001	would better experience video screen shot she side text instructor going thru user go way beginning text able view slide instructor showing	Netral
000004	excellent course train provide detail easy follow	Netral
000006	solid presentation way really appreciate intermittent question popped check learn well regular needle quiz visual chart ppt visual incline well transcript video follow along presentation	Positif
000008	proctoru com system took time amount time spent course day complete worse production user system use year career switch another vendor	Positif
000019	excellent training enough past test	Positif

Hasil *preprocessing* kemudian dibagi menjadi potongan-potongan kata disebut token untuk dilakukan ekstraksi fitur dengan merubah kata menjadi vektor atau *Word2vec*.

B. *Word2vec Model*

Pembuatan model *Word2vec* dapat memanfaatkan *library Gensim*, dengan hanya mengidentifikasikan nilai parameter yang ada. Parameter *Word2vec* pada *library Gensim* beserta fungsinya antara lain;

vector_size : ukuran fitur *n* dimensi
window : rentang pembacaan *token*s sebelum dan sesudah
min_count : frekuensi kata yang muncul dalam teks untuk diekstraksi
sg : aktivasi model *Skip-gram* dengan *binary value, default* nya bernilai 0 artinya artinya tidak menjalankan *Skip-gram*, namun model sebaliknya yaitu CBOW.
worker : Jumlah *thread* saat melatih model disesuaikan dengan *core* CPU
epochs : Jumlah perulangan (*epoch*) yang diterapkan dalam proses *training*.

Percobaan parameter *Word2vec* diasumsikan, memiliki *vector_size = 300*, *window* = 5, *min_count* = 3, *sg* = 1 dan *epoch* bernilai *default*. Kemampuan model dalam mengenali hubungan semantik antar kata dapat dilihat ketika diminta memeriksa *cosine similarity* antara kata "*course"* dan "*class*". Hasilnya, berdasarkan jarak dalam ruang vektor, kedekatan antara dua kata tersebut sebesar **0,00255** atau mendekati 0, artinya model mengidetifikasi dua kata tersebut sebagai kata yang mirip. Selain itu, dapat dilakukan dengan cara model diminta menampilkan beberapa kata yang mirip dengan sebuah kata, sebagai contoh kata *"code"* dan *"instructor"*.

Adapun hasil kemiripan kata dari dua kata tersebut ditunjukkan Tabel 6, sebagai berikut.

TABEL 6
KEMAMPUAN SEMANTIK ANTAR KATA WORD2VEC

Kata	**1**	**2**	**3**	**4**	**5**
Code	*program*	*coding*	*syntax*	*python*	*algorithm*
	74,4%	**74,2%**	**74%**	**72,4%**	**72%**
Kata	**1**	**2**	**3**	**4**	**5**
Instructor	*lecturer*	*tutor*	*teacher*	*professor*	*trainer*
	84,4%	**83%**	**81,4%**	**80,3%**	**74,7%**

Berdasarkan Tabel 6, model mampu menampilkan kata-kata yang mirip dengan kata masukan yang diminta, seperti kata *instructor* dengan *lecturer* yang memiliki kemiripan 84,4%, dimana secara makna dua kata tersebut sama-sama merujuk pada profesi seorang tenaga pendidik.

C. Pengujian Sistem

Pengujian dilakukan dengan tiga skenario pembagian data latih dan data uji yang berbeda, yaitu 90:10, 80:20 dan 70:30, sekaligus kombinasi parameter *window* dan ukuran vektor yang digunakan sejumlah 3, 5 dan 8 *window* dan ukuran vektor sebesar 100, 300 dan 500 dimensi. Dari tiga skenario pembagian data dan jumlah parameter *Word2vec* akan menghasilkan 27 model. Hasil pengujian model diperoleh kombinasi proporsi data dan parameter *Word2vec* terbaik untuk sistem analisis sentimen data *review* berdasarkan akurasi model (*train score*) dan akurasi pengujian (*test score*). Kombinasi terbaik akan dilakukan evaluasi untuk mengetahui peformanya saat memprediksi sentimen.

Adapun hasil pengujian parameter untuk masing- masing skenario ditunjukkan sebagai berikut.

1. Skenario 90:10

Menghasilkan akurasi model dan pengujian ditunjukkan pada Tabel 7 berikut.

TABEL 7
PENGUJIAN MODEL DENGAN PEMBAGIAN DATA 90:10

Model	Matriks	*Window*	*Train Score*	*Test Score*
1	100	3	0,9268	0,9138
2	300	3	0,9181	0,8957
3	500	3	0,9232	0,9133
4	100	5	0,9277	0,9114
5	300	5	0,9253	0,9105
6	500	5	0,9238	0,9015
7	100	8	0,9265	0,9046
8	300	8	0,9255	0,9119
9	500	8	0,9214	0,9053

2. Skenario 80:20

Menghasilkan akurasi model dan pengujian ditunjukkan pada Tabel 8.

TABEL 8
PENGUJIAN MODEL DENGAN PEMBAGIAN DATA 80:20

Model	Matriks	*Window*	*Train Score*	*Test Score*
1	100	3	0,9265	0,9125
2	300	3	0,9188	0,9058
3	500	3	0,9253	0,9112
4	100	5	0,9271	0,9142
5	300	5	0,9264	0,9133
6	500	5	0,9244	0,9125
7	100	8	0,9245	0,9046
8	300	8	0,9265	0,9023
9	500	8	0,9262	0,9057

3. Skenario 70:30

Menghasilkan akurasi model dan pengujian ditunjukkan pada Tabel 9.

TABEL 8
PENGUJIAN MODEL DENGAN PEMBAGIAN DATA 70:30

Model	Matriks	*Window*	*Train Score*	*Test Score*
1	100	3	0,9257	0,9153
2	300	3	0,9282	0,9124
3	500	3	0,9238	0,9134
4	100	5	0,9255	0,9049
5	300	5	0,9261	0,9075
6	500	5	0,9273	0,9058
7	100	8	0,9184	0,9066
8	300	8	0,9254	0,9055
9	500	8	0,9265	0,9041

Tabel diatas, menggambarkan bahwa model cukup stabil dan memiliki kemampuan **generalisasi yang baik**, artinya sistem mampu memprediksi sentimen sebuah teks dengan baik pula yang terlihat dari hampir samanya antara hasil *training* dan *testing*. Hasil dari tiga skenario diatas, diperoleh rata-rata untuk masing-masing akurasi model (*train score*) dan akurasi pengujian (*test score*) sebesar;

a. Skenario 1
   Train Score : 0,9243
   Test Score : 0,9076
b. Skenario 2
   Train Score : 0,9251
   Test Score **: 0,9091**
c. Skenario 3
   Train Score : 0,9252
   Test Score : 0,9084

Berdasarkan rata-rata akurasi diatas, menunjukkan bahwa pembagian data dengan proporsi 80:20 memiliki nilai *test score* paling tinggi dibandingkan proporsi data lainnya, dengan kombinasi parameter *Word2vec* terbaik *vector_size=100* dan *window=5*. Model tersebut selanjutnya akan dievaluasi untuk mengetahui performnya menggunakan matriks evaluasi. Hasilnya, *confusion matriks* untuk masing-masing kelas baik, positif, netral dan negatif ditunjukkan pada Tabel 9, Tabel 10, dan Tabel 11 berikut ini.

TABEL 9
CONFUSION MATRIKS KELAS POSITIF

Kelas	**Positif**	**Bukan**
Positif	105153	8548
Bukan	2975	23748

$$Precision = \frac{TP}{TP+FN}$$
$$= \frac{105153}{105153+8548} = \mathbf{0,9724}$$

$$Recall = \frac{TP}{TP+FP}$$
$$= \frac{105153}{105153+8548} = \mathbf{0,9248}$$

$$F1-Score = 2 \times \frac{precision\ .\ recall}{precision+recall}$$
$$= 2 \times \frac{0,9724\ .\ 0,9248}{0,9724+0,9248} = \mathbf{0,9480}$$

TABEL 10
CONFUSION MATRIKS KELAS NETRAL

Kelas	**Netral**	**Bukan**
Netral	17599	2801
Bukan	7702	111644

$$Precision = \frac{TP}{TP+FN}$$
$$= \frac{17599}{17599+7702} = \mathbf{0,6955}$$

$$Recall = \frac{TP}{TP+FN}$$
$$= \frac{17599}{17599+2801} = \mathbf{0,8626}$$

$$F1-Score = 2 \times \frac{precision\ .\ recall}{precision+recall}$$
$$= 2 \times \frac{0,9724\ .\ 0,9248}{0,9724+0,9248} = \mathbf{0,7701}$$

TABEL 11
CONFUSION MATRIKS KELAS NEGATIF

Kelas	**Negatif**	**Bukan**
Negatif	6149	811
Bukan	1446	123094

$$Precision = \frac{TP}{TP+FP}$$
$$= \frac{6149}{6149+1446} = \mathbf{0,8096}$$

$$Recall = \frac{TP}{TP+FN}$$
$$= \frac{6149}{6149+881} = \mathbf{0,8834}$$

$$F1-Score = 2 \times \frac{precision\ .\ recall}{precision+recall}$$
$$= 2 \times \frac{0,8429\ .\ 0,8746}{0,8429+0,8746} = \mathbf{0,8449}$$

Dari evaluasi model dapat dikatakan tingkat ketepatan (*precision*) sistem dalam memprediksi sentimen negatif sebesar 81%, sentimen netral sebesar 70%, dan sentimen positif sebesar 97%. Sedangkan tingkat keberhasilan sistem menemukan kembali sebuah informasi (*recall*) untuk sentimen negatif sebesar 89%, sentimen netral 86% dan sentimen positif 93%.

Nilai *f1-score* untuk sentimen negatif sebesar 84,5%, sentimen netral 77% dan sentimen positif 95%. Maknanya keberhasilan sistem dalam memprediksi atau menganalisis sentimen sebuah teks secara otomatis untuk kasus data *review* pengguna *Coursera* dengan **cukup baik**, dengan kesimpulan sentimen sebanyak 114043 pengguna menyatakan puas (sentimen positif), 20400 pengguna merasa cukup puas (sentimen netral) dan sebanyak 6923 pengguna merasa tidak puas (sentimen negatif).

D. Analisis Perbandingan

Ditunjukkan akurasi dan nilai *f1-score* keseluruhan kelas untuk analisis yang menggunakan *Random Forest* dengan TF *Random Forest* dengan pembobotan TF-IDF, kemudian dibandingkan antara dua kombinasi tersebut dengan hasil akurasi dan nilai *f1-score* keseluruhan kelas dari analisis sentimen menggunakann *Random Forest* dengan ekstraksi fitur *Word2vec*, yang ditunjukkan pada Tabel 12 berikut.

TABEL 12
PERBANDINGAN AKURASI DAN F1-SCORE DARI 3 KOMBINASI

Classifier	**Akurasi (%)**	***F1-Score* (%)***
Random Forest + TF	0,9043	0,8309
Random Forest + TF-IDF	0,9125 (+0,0082)	0,8490 (+0,0181)
Random Forest +Word2vec	**0,9142 (+0,0017)**	**0,8595 (+0,0105)**

* : rata-rata *F1-Score* keseluruhan kelas sentimen

Berdasarkan hasil perbandingan secara keseluruhan, nilai akurasi dan nilai *f1-score* tertinggi terjadi pada kombinasi *Random Forest* dengan *Word2vec*, walaupun *Random Forest* dengan TF-IDF sudah cukup baik. Penyebabnya, ukuran dimensi fitur yang rendah, 100 dimensi, *Word2vec* sudah mampu mengenali hubungan semantik antar kata. Selain itu, dengan ukuran dimensi yang ringkas, waktu yang dibutuhkan saat *training model* menggunakan *Randomm Forest* hanya selama 33 menit 19 detik, dimana waktu tersebut 20 kali lebih cepat dibandingkan *Random Forest* + TF yang membutuhkan waktu 9 jam 16 menit 32 detik dan *Random Forest* + TF-IDF, yang membutuhkan waktu selama 6 jam 28 menit 44 detik.

Sehingga terbukti analisis sentimen *Random Forest* akan efisien sekaligus efektif apabila menggunakan *Word2vec* sebagai esktraksi fiturnya, khususnya pada kasus teks *review* dalam jumlah banyak.

IV. KESIMPULAN

Berdasarkan hasil penelitian dan pembahasan sebelumnya diperoleh kesimpulan sebagai berikut:

1. Alur pembangunan sistem untuk menganalisis sentimen pengguna platform belajar *Coursera* diawali dengan tahap persiapan data, *labelling*, pra-pemrosesan, proses ekstraksi teks dengan *Word2vec*, serta pembuatan model klasifikasi menggunakan *Random Forest*, dan evaluasi model.
2. Hasil pengujian sistem pada proporsi data berbeda dan parameter *Word2vec* acak, menunjukkan akurasi terbaik pada proporsi data 80:20 dengan ukuran fitur 100 dimensi dan *window* 5 dengan dengan masing-masing *train score* dan *test score* sebesar 92,7% dan 91,4%.
3. Hasil perbandingan, diperoleh akurasi dan nilai *f1-score* tertinggi pada kombinasi Random Forest dan Word2vec dengan waktu yang dibutuhkan *training model* 20 kali lebih cepat dibanding menggunakan *Random Forest*+TF dan *Random Forest*+TF-IDF.

V. SARAN

Berdasarkan hasil dan kesimpulan penelitian ini, beberapa saran penulis yang mungkin dapat diterapkan untuk penelitian selanjutnya khususnya pada kasus analisis sentimen adalah;

1. Identifikasi parameter *epochs* dapat dilakukan pada saat pembuatan model Wor2vec untuk meningkatkan akurasi.
2. Walaupun akurasi yang dihasilkan cukup baik, untuk kasus *text classification* seperti analisis sentimen, penggunaan *tree-based model* (seperti; *Random Forest*, *XG Boost* dan lain-lain) tidak disarankan karena akan semakin memperlambat proses pembelajaran, karena model memproses fitur sebanyak fitur atau *n-vocab*.
3. Disarankan mengkomparasikan hasil akurasi dengan algoritma *classifier* lainnya, guna mengetahui efektivitas *Word2vec* sebagai ekstraksi fitur.untuk kasus analisis sentimen.

UCAPAN TERIMA KASIH

Ungkapan terimakasih sebanyak-banyaknya kepada Allah SWT, Tuhan Yang Maha Esa, atas rahmat-Nya penulis dapat menyelesaikan penelitian dengan lancar dan sukses. Tidak lupa ucapan terimakasih ditujukkan kepada kedua orang tua, dosen pembimbing, rekan-rekan dan semua pihak yang telah terlibat dalam proses penelitian ini hingga akhir.

REFERENSI

[[1] M. Nakhaee, "Course Reviews on Coursera Datasets," *Kaggle.com*, 2020. https://www.kaggle.com/datasets/imuhammad/course-reviews-on-coursera (accessed May 24, 2022).

[2] AWS, "What is Sentiment Analysis?," *Amazon Web Services*, 2022. https://aws.amazon.com/id/what-is/sentiment-analysis/ (accessed Dec. 12, 2022).

[3] W. G. Evita Fitri, Yuri Yuliani, Susy Rosyida, "Analisis Sentimen Terhadap Aplikasi Ruangguru Menggunakan Algoritma Naive Bayes, Random Forest Dan Support Vector Machine," *J. Transform.*, vol. 18, no. 1, p. 71, 2020, doi: 10.26623/transformatika.v18i1.2317.

[4] B. Christanto and D. H. Setiabudi, "Penerapan Random Forest dalam Email Filtering untuk Mendeteksi Spam," *J. Infra*, vol. 8, no. 2, 2020.

[5] irwan budiman, M. R. Faisal, and D. T. Nugrahadi, "Studi Ekstraksi Fitur Berbasis Vektor Word2Vec pada Pembentukan Fitur Berdimensi Rendah," *J. Komputasi*, vol. 8, no. 1, pp. 62–69, 2020, doi: 10.23960/komputasi.v8i1.2517.

[6] A. N. Assidyk, E. B. Setiawan, and I. Kurniawan, "Analisis Perbandingan Pembobotan TF-IDF dan TF-RF pada Trending Topic di Twitter dengan Menggunakan Klasifikasi K-Nearest Neighbor," *e-Proceeding Eng.*, vol. 7, no. 2, pp. 7773–7781, 2020.

[7] X. Jin, S. Zhang, and J. Liu, "Word Semantic Similarity Calculation Based on Word2vec," *ICCAIS 2018 - 7th Int. Conf. Control. Autom. Inf. Sci.*, pp. 12–16, Dec. 2018, doi: 10.1109/ICCAIS.2018.8570612.

[8] S. Khomsah, "Sentiment Analysis On YouTube Comments Using Word2Vec and Random Forest," *Telematika*, vol. 18, no. 1, p. 61, 2021, doi: 10.31315/telematika.v18i1.4493.

[9] R. P. Nawangsari, R. Kusumaningrum, and A. Wibowo, "Word2Vec for Indonesian Sentiment Analysis towards Hotel Reviews: An Evaluation Study," *Procedia Comput. Sci.*, vol. 157, pp. 360–366, Jan. 2019, doi: 10.1016/J.PROCS.2019.08.178.

[10] Z.-H. Zhou, *Ensemble methods - Zhou*, 13th ed., vol. 2, no. Schapire 1990. Cambridge: Taylor & Francis Group, 2007.

[11] M. L. Suliztia, "PENERAPAN ANALISIS RANDOM FOREST PADA PROTOTYPE SISTEM PREDIKSI HARGA KAMERA BEKAS MENGGUNAKAN FLASK," Universitas Islam Indonesia, 2020.