



## Optimasi *Hyperparameter* pada *Neural Network* (Studi Kasus: Identifikasi Komentar *Cyberbullying* Instagram)

Nur Fajriyani<sup>#1</sup>, Enda Esyudha Pratama<sup>#2</sup>, Rina Septiriana<sup>#3</sup>

<sup>#</sup>Jurusan Informatika Fakultas Teknik Universitas Tanjungpura  
Jl. Prof. Dr. H. Hadari Nawawi, Pontianak, 78124

<sup>1</sup>nurfajriyani16@student.untan.ac.id

<sup>2</sup>enda@informatika.untan.ac.id

<sup>3</sup>rinaseptiriana@informatika.untan.ac.id

Abstrak— Instagram adalah salah satu media sosial yang populer di Indonesia dan memiliki resiko terjadinya *cyberbullying*, sehingga klasifikasi komentar *cyberbullying* perlu dilakukan menggunakan *text mining*. Model klasifikasi yang dibangun pada penelitian ini menggunakan *Neural Network* dengan arsitektur *Multilayer Perceptron* dan dilatih dengan algoritma *Backpropagation*. *Bayesian Optimization* digunakan untuk melakukan optimasi *hyperparameter*, dengan *hyperparameter* yang dioptimasi adalah *hidden layer*, *learning rate*, dan *momentum*. Dilihat dari hasil evaluasi, performa dari model *Neural Network* dengan optimasi *hyperparameter* lebih baik daripada *Neural Network* yang hanya menggunakan default *hyperparameter*. Dimana model *Neural Network* dengan default *hyperparameter* mengalami *overfitting* dan menghasilkan nilai akurasi sebesar 81,38%, sedangkan model dengan optimasi *hyperparameter* tidak mengalami *overfitting* dan mengalami kenaikan akurasi dibandingkan dengan model *Neural Network* tanpa optimasi *hyperparameter* dengan model dengan tiga *hyperparameter* yang dioptimasi yaitu *hidden layer*, *learning rate*, dan *momentum* mengalami kenaikan akurasi sebesar 2,5%, model dengan optimasi *hyperparameter* pada *hidden layer* dan pada *learning rate* masing-masing mengalami kenaikan akurasi yang sama, yaitu sebesar 2,37%, dan model dengan optimasi *hyperparameter* pada *momentum* mengalami kenaikan akurasi sebesar 2%. Hasil penelitian ini menunjukkan bahwa *hidden layer*, *learning rate*, dan *momentum* ketika di optimasi secara bersamaan memiliki pengaruh besar dalam mencegah *overfitting*, menaikkan akurasi, dan memiliki waktu eksekusi yang baik daripada dioptimasi secara terpisah.

**Kata kunci**— Klasifikasi, *Neural Network*, Optimasi *Hyperparameter*, *Bayesian Optimization*

### I. PENDAHULUAN

Di Indonesia, terdapat 212,9 juta pengguna internet per Januari 2023. Media sosial digunakan oleh sebagian besar pengguna internet di Indonesia, dimana 167 juta orang atau sekitar 60,4% dari populasi saat ini memanfaatkan media sosial [1].

Instagram merupakan salah satu *platform* media sosial yang populer di kalangan masyarakat Indonesia, dengan rata-rata 215 juta kunjungan per bulan [2]. Instagram adalah *platform* berbagi gambar dan video gratis yang tersedia untuk berbagi perangkat seluler dan desktop. Instagram juga memiliki fitur yang memungkinkan seseorang untuk mengomentari foto dan video yang diunggah. Namun, dengan adanya fitur-fitur tersebut dapat memiliki efek negatif dan efek positif. Efek negatif yang muncul adalah terjadinya *cyberbullying* [3].

Mengingat Instagram adalah salah satu *platform* media sosial paling banyak diakses oleh masyarakat Indonesia, hal itu menyebabkan tingginya risiko masalah *cyberbullying* dan masalah *cyberbullying* pada komentar Instagram menjadi hal yang perlu untuk dikaji dalam pemrosesan teks. Karena permasalahan tersebut, maka diperlukan klasifikasi otomatis atau klasifikasi menggunakan *text mining*. *Text mining* saat ini menjadi topik yang banyak dibicarakan karena data yang dihasilkan seseorang bisa sangat besar. Metode dan teknik tertentu diperlukan untuk mengubah untuk mengubah kumpulan data yang dihasilkan seseorang menjadi informasi untuk pengambilan keputusan [4]. Algoritma *Support Vector Machine*, *Naive Bayes*, *K-Nearest Neighbors*, *Neural Network*, dan teknik lainnya merupakan algoritma untuk melakukan klasifikasi. Hasil kinerja masing-masing algoritma bergantung pada masalah yang dikaji, data yang digunakan, dan kriteria lainnya [5].

Penelitian tentang mendeteksi *cyberbullying* pada komentar Instagram pernah dilakukan sebelumnya oleh [6] menggunakan metode *Support Vector Machine (SVM)* dan menghasilkan nilai akurasi tertinggi yaitu 90%, dan penelitian lainnya oleh [3], penelitian ini membangun sebuah aplikasi untuk mendeteksi *cyberbullying* dengan metode *Naive Bayes Classifier* dan menghasilkan akurasi sekitar 98,5%.

Adapun penelitian terkait klasifikasi oleh [7], penelitian ini membandingkan performa pada metode klasifikasi

SVM, Neural Network, dan Naive Bayes untuk klasifikasi kualitas pengajuan kredit di koperasi simpan pinjam. Hasil Neural Network memiliki performa yang lebih baik daripada SVM dan Naive Bayes.

Penelitian Neural Network juga pernah dilakukan oleh [8], dimana penelitian ini dilakukan optimasi parameter pada Neural Network dengan menggunakan Algoritma Genetika, hasil dari penelitian ini menunjukkan kombinasi Neural Network-Algoritma Genetika lebih baik daripada hanya menggunakan Neural Network saja. Menemukan konfigurasi hyperparameter yang menghasilkan kinerja terbaik dikenal sebagai optimasi hyperparameter. Melakukan pengaturan hyperparameter dapat mempengaruhi seberapa baik kinerja algoritma klasifikasi menjadikan langkah ini penting [9]. Selama beberapa tahun terakhir, berbagai algoritma dikembangkan untuk mengoptimalkan nilai hyperparameter, diantaranya adalah menggunakan Random Search, pendekatan berbasis model menggunakan Random Forest, dan optimasi berbasis model sekuensial [10]. Optimasi berbasis model sekuensial menjadi strategi yang paling baik untuk mengoptimalkan hyperparameter, karena melibatkan model probabilistik data untuk menentukan titik yang paling menjanjikan untuk dievaluasi. Salah satu algoritma optimasi berbasis model sekuensial adalah Bayesian Optimization [10].

Penelitian terkait optimasi hyperparameter menggunakan Bayesian Optimization pernah dilakukan pada algoritma Gradient Boosted Trees oleh [4], hasil dari penelitian ini menunjukkan optimasi hyperparameter menggunakan Bayesian Optimization terhadap model *machine learning* mengalami kenaikan akurasi dibanding tidak menggunakan optimasi hyperparameter.

Oleh karena itu, model klasifikasi yang dibangun pada penelitian ini akan menggunakan Neural Network. Klasifikasi yang dilakukan merupakan *supervised learning* dalam *machine learning*, dan model yang termasuk *supervised learning* pada Neural Network adalah dengan arsitektur Multilayer Perceptron yang dilatih dengan algoritma Backpropagation.

Menurut [11] dan [12] terdapat faktor-faktor yang mempengaruhi kinerja algoritma Backpropagation, yaitu hidden layer, learning rate, dan momentum, dimana faktor tersebut merupakan hyperparameter dalam neural Network. Karena hyperparameter tersebut merupakan faktor penting pelatihan algoritma backpropagation maka dalam membangun model klasifikasi pada Neural Network dalam penelitian ini dilakukan optimasi hyperparameter tersebut.

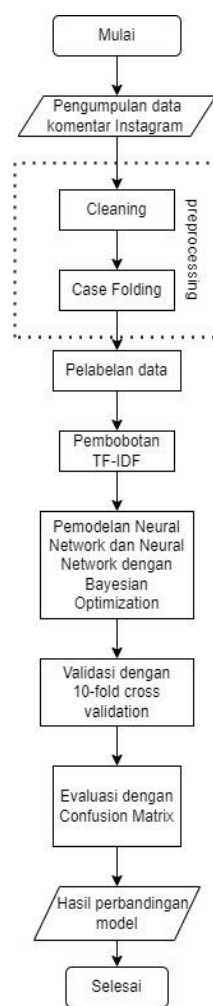
Berdasarkan latar belakang yang telah dipaparkan maka dalam penelitian ini akan dilakukan optimasi hyperparameter Neural Network menggunakan Bayesian Optimization pada kasus identifikasi komentar *cyberbullying* Instagram, dengan hyperparameter yang akan dioptimasi adalah hidden layer, learning rate, dan momentum.

## II. METODOLOGI

Terdapat tujuh tahap metode penelitian yang dilakukan pada penelitian ini yang dapat dilihat pada Gambar 1 untuk membangun model *machine learning*.

### A. Pengumpulan Data

Untuk membuat suatu *machine learning* dibutuhkan suatu data pembelajaran. Data pembelajaran ini berupa teks. Data yang digunakan berasal dari komentar di lima akun Instagram *public figure* di Indonesia, diantara @najwashihab, @smidrawati, @zinidinzidan\_real, @amandamanopo, dan @cimoyunuraini\_wjy. Data tersebut dikumpulkan menggunakan *web scraping* <https://phantombuster.com/>



Gambar. 1 Langkah penelitian

### B. Text Preprocessing

Penelitian ini menggunakan *tools* Google Colab dengan menggunakan bahasa pemrograman Python. Dataset yang digunakan berupa komentar Instagram yang dikumpulkan dari lima akun *public figure* di Indonesia. Label kelas data terdiri dari *cyberbullying* dan *non-cyberbullying*. Data tersebut kemudian di inialisasi ke dalam Google Colab

untuk pra-pemrosesan teks. Hasil inisialisasi data dapat dilihat pada Gambar 2.

*Text preprocessing* merupakan suatu proses mengubah teks asli sebagai masukan dan menerapkan beberapa rutinitas dasar yang mengubah atau menghilangkan unsur tekstual yang tidak berguna dalam pengolahan lebih lanjut dalam proses klasifikasi [13]. *Text preprocessing* diperlukan untuk memaksimalkan performa dari algoritma klasifikasi [14].

*Text preprocessing* pada penelitian ini menggunakan bahasa pemrograman Python yang memanfaatkan *library* seperti NLTK, Pandas, dan lainnya. Adapun tahapan-tahapan *Text Preprocessing* yang dilakukan sebagai berikut:

1) *Cleaning*: Pada penelitian ini *Text Preprocessing* dimulai dengan proses *cleaning*. Pada tahapan ini membersihkan text dari link, tag html, script dan komponen lainnya yang tidak memiliki kaitan informasi pada kalimat, seperti (0-9,!@#\$\$%&\*+\_={}[;:;''/?<>). Contoh kode program dan hasil *cleaning* dapat dilihat pada Gambar 2 dan Gambar 3.

```

1 #Proses cleaning
2 def filtering_text(text):
3     # menghilangkan url
4     text = re.sub('https?://\S+', '', text)
5     # menghilangkan mention, link, hashtag
6     text = ''.join(re.sub('@|#|_|[A-Za-z0-9]+|(\w+://\S+)', '', text).split())
7     # menghilangkan yang bukan huruf
8     text = re.sub('[^a-zA-Z]', '', text)
9     # menghilangkan digit angka
10    text = re.sub('[0-9]', '', text)
11    # menghilangkan tanda baca
12    text = text.translate(str.maketrans("", "", string.punctuation))
    
```

Gambar. 2 Kode program *cleaning*



Gambar. 3 *Cleaning*

2) *Case Folding*: yaitu proses mengubah semua huruf dalam komentar menjadi huruf kecil (*lowercase*). Contoh kode program dan hasil *case folding* dapat dilihat pada Gambar 4 dan Gambar 5.

```

14 #Proses case folding
15 # mengubah teks menjadi huruf kecil
16 text = text.lower()
17
18 return text
19
20 data['filtered'] = data['text'].apply(filtering_text)
21 print(data['filtered'].head())
    
```

Gambar. 4 Kode program *case folding*



Gambar. 5 *Case folding*

C. Pelabelan Data

Data yang telah didapatkan dari proses *Text Preprocessing* kemudian dipilih dan diberi label mengandung *cyberbullying* dan tidak mengandung *cyberbullying* oleh 4 orang partisipan secara manual. Pemberian label pada dataset mengikuti kriteria komentar yang mengandung *cyberbullying* atau non-*cyberbullying* yang disebutkan dalam penelitian oleh [15] serta pemberian label pada komentar yang akan dijadikan dataset

disesuaikan dengan unggahan yang telah dipilih untuk diambil data komentarnya.

Pada tahap ini dipilih 4000 data yang akan dijadikan dataset dari 34777 data yang telah dikumpulkan. Dimana 2000 data diberi label 1 yang menandakan data tersebut mengandung *cyberbullying* dan 2000 data lainnya diberi label 0, dimana menandakan data tersebut tidak mengandung *cyberbullying*. Contoh kalimat pada dataset yang telah diberi label dilihat Gambar 6.

	text	target
0	aaaa yaa allah bagus bngeet suara nyaa masya a...	0.0
1	aa games	0.0
2	aa games yg item matanya gx keliatan	0.0
3	aaa buat candu	0.0
4	aaa cantik banget	0.0
...	...	...
3995	oprasi plastik di thailand aja muka suneo	1.0
3996	mukanya kaga ada aura positif nya acan kusem	1.0
3997	bibir nya keluar eek apa yaa bisa sesombong it...	1.0
3998	muka apa rempeyek itu banyak kacangnya hahaha	1.0
3999	jadi pengen amplas mukanya	1.0

Gambar. 6 Pelabelan data

D. Pembobotan TF-IDF

Pembobotan TF-IDF digunakan pada penelitian ini untuk melakukan vektorisasi. Proses vektorisasi melibatkan perubahan data teks menjadi data numerik, dimana nilainya mewakili jumlah kemunculan kata dalam teks. Karena komputer hanya memahami dan menginterpretasikan data numerik, maka pembobotan ini diperlukan. Pada penelitian ini pembobotan TF-IDF dilakukan dengan menggunakan *library* Scikit-Learn.

E. Pemodelan

Terdapat dua pendekatan yang dipilih untuk melakukan klasifikasi pada penelitian ini, yaitu *Neural Network* dan *Bayesian Optimization*.

1) *Neural Network*: *Neural Network* merupakan bagian dari kecerdasan buatan yang memiliki kapasitas untuk belajar, berkembang, dan beradaptasi dengan lingkungan yang berubah. Arsitektur Multilayer Perceptron (MLP) adalah jenis jaringan saraf yang menggabungkan pembelajaran terawasi dan sebagian besar digunakan dalam bidang pendidikan [16]. Penelitian ini membangun model *Neural Network* dengan menggunakan bahasa pemrograman Python yang memanfaatkan fungsi *MLPClassifier* dari *library* Sklearn. Berikut bentuk model *Neural Network*:

$$J_{neural}(\theta) = \sum_k = 1K J_k(\theta) = \sum_k = 1K [1m \sum 1m 12 cost(h\theta(x(i))_k, y(i)_k)] \quad (1)$$

Arsitektur Multilayer Perceptron (MLP) berbasis backpropagation diimplementasikan oleh *MLPClassifier*. Backpropagation adalah metode pembelajaran yang menurunkan tingkat kesalahan (*error*) dengan cara memvariasikan bobotnya sesuai dengan ketidaksesuaian antara output dan target yang diinginkan. Jaringan saraf dengan arsitektur Multilayer Perceptron (MLP) juga dapat dilatih secara sistematis melalui backpropagation [17].

2) *Bayesian Optimization*: *Bayesian Optimization* merupakan teknik optimasi yang mencari nilai terkecil atau

maksimum dari suatu fungsi  $f(x)$  yang dibatasi oleh domain [18]. Sebelum melakukan optimasi menggunakan Bayesian Optimization, ada dua faktor yang harus diperhatikan. Pertama, model probabilitas  $M$  yang akan digunakan, dan kedua, fungsi akuisisi yang akan digunakan untuk memilih nilai hyperparameter selanjutnya yang akan dievaluasi. Dengan mengambil sampel dari domain  $X$  untuk menginisialisasi  $M$ , optimasi Bayesian beroperasi. Jumlah hyperparameter yang harus dioptimalkan untuk memilih nilai baru dalam domain  $X$  [19]. Fungsi akuisisi  $S(x)$  kemudian dioptimalkan untuk memilih nilai baru dalam domain  $X$ .

$$D = \{(x_1y_1), \dots, (x_iy_i)\} \quad (2)$$

Sebagai data *historical* akan digunakan untuk merepresentasikan hasil dari  $f(x)$ , dengan  $x$  berfungsi sebagai nilai hyperparameter dan  $y$  berfungsi sebagai output dari  $f(x)$ . Langkah-langkah ini akan terus dilakukan berulang hingga mencapai titik *convergence*. Dalam Bayesian Optimization, Gaussian Process (GP), biasanya digunakan sebagai model probabilistik [20].

3) *Skenario Pengujian*: Pada penelitian ini terdapat lima skenario pengujian, yaitu skenario pengujian pertama dilakukan pada model Neural Network dengan default hyperparameter, skenario pengujian kedua dilakukan menggunakan menggunakan optimasi hyperparameter pada Neural Network dengan hyperparameter yang akan dioptimasi yaitu hidden layer, learning rate, dan momentum, skenario pengujian ketiga dilakukan menggunakan menggunakan optimasi hyperparameter pada Neural Network dengan hyperparameter yang akan dioptimasi yaitu hidden layer, skenario pengujian keempat dilakukan menggunakan menggunakan optimasi hyperparameter pada Neural Network dengan hyperparameter yang akan dioptimasi yaitu learning rate, dan skenario pengujian kelima dilakukan menggunakan menggunakan optimasi hyperparameter pada Neural Network dengan hyperparameter yang akan dioptimasi yaitu momentum.

TABEL I  
SKENARIO PENGUJIAN

Skenario Pengujian	Optimasi Hyperparameter dengan Bayesian Optimization	Hyperparameter yang akan dioptimasi		
		Hidden Layer	Learning Rate	Momentum
1	x	x	x	x
2	✓	✓	✓	✓
3	✓	✓	x	x
4	✓	x	✓	x
5	✓	x	x	✓

Dari skenario pengujian yang telah dibuat, maka pada penelitian ini akan dibangun lima model, antara lain:

1. Model pertama, hanya menggunakan Neural Network dengan default hyperparameter, dimana nilai default hidden layer adalah 1, nilai default learning rate adalah 0,001, dan nilai default adalah 0,9.

2. Model kedua, Neural Network dengan Bayesian Optimization sebagai algoritma optimasi dengan hyperparameter yang akan dioptimasi yaitu hidden layer, learning rate, dan momentum. Dimana range nilai (*lower bound* dan *upper bound*) dari hyperparameter hidden layer adalah 1 hingga 4, nilai learning rate adalah 0,001 hingga 0,6, dan nilai momentum adalah 0,5 hingga 0,9.
3. Model ketiga, Neural Network dengan Bayesian Optimization sebagai algoritma optimasi hyperparameter dengan hyperparameter yang akan dioptimasi yaitu hidden layer. Dimana range nilai (*lower bound* dan *upper bound*) dari hyperparameter hidden layer adalah 1 hingga 4.
4. Model keempat, Neural Network dengan Bayesian Optimization sebagai algoritma optimasi hyperparameter dengan hyperparameter yang akan dioptimasi yaitu learning rate. Dimana range nilai (*lower bound* dan *upper bound*) dari hyperparameter learning rate adalah 0,001 hingga 0,6.
5. Model kelima Neural Network dengan Bayesian Optimization sebagai algoritma optimasi hyperparameter dengan hyperparameter yang akan dioptimasi yaitu momentum. Dimana range nilai (*lower bound* dan *upper bound*) dari hyperparameter momentum adalah 0,5 hingga 0,9.

F. Validasi

Penelitian ini menggunakan *10-fold cross validation* sebagai alat validasi. Validasi dilakukan data latih untuk mencegah terjadinya *overfitting* pada model, selain itu validasi juga untuk mengetahui seberapa baik model memprediksi data baru yang belum pernah dilatih sebelumnya.

G. Evaluasi

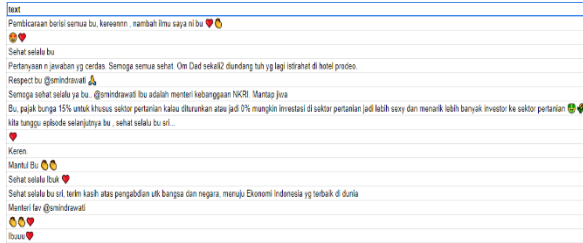
Pada penelitian ini, confusion matrix digunakan sebagai alat ukur performa klasifikasi dari metode yang digunakan. Evaluasi dengan confusion matrix untuk mendapatkan nilai *accuracy*, *precision*, *recall*, dan *F1-score* model *machine learning*.

III. HASIL DAN PEMBAHASAN

Tahap-tahap yang dilakukan untuk membuat model *machine learning* mulai dari pengumpulan data, preprocessing, pelabelan data, preprocessing, pelabelan data, pembobotan TF-IDF, hingga pemodelan telah dilakukan. Terdapat lima model yang dibuat pada penelitian ini. Data latih pada setiap model dibagi menggunakan *10-fold cross validation*. Hasil prediksi data uji dibandingkan dengan label untuk dievaluasi menggunakan confusion matrix.

A. Hasil Pengumpulan Data

Pengumpulan data komentar yang dilakukan menggunakan metode *scraping* data dengan tools Phantombuster.com, didapatkan data sebanyak 34777 baris data. Sampel data dapat dilihat pada Gambar 7.



Gambar. 7 Sampel data

**B. Hasil Text Preprocessing**

Hasil sampel dataset dari keseluruhan *Text Preprocessing* ditampilkan dalam bentuk tabel yang dapat dilihat pada Tabel II.

TABEL II  
HASIL TEXT PREPROCESSING

Text	Text Preprocessing
Semoga sehat selalu ya bu @smindrawati, Ibu adalah menteri kebanggaan NKRI. Mantap jiwa!	Sebelum Preprocessing
Semoga sehat selalu ya bu smindrawati Ibu adalah menteri kebanggaan NKRI Mantap jiwa	Cleaning
semoga sehat selalu ya bu smindrawati Ibu adalah menteri kebanggaan nkri mantap jiwa	Case Folding

**C. Hasil Pelabelan Data**

Hasil pelabelan data secara manual yang dilakukan oleh 4 orang mahasiswa sebagai partisipan dapat dilihat pada Gambar 8 dan Gambar 9.

aaaa yaa allah bagus bngeet suara nyaa masya allah	0
aa gemes	0
aa gemes yg item matanya gx keliatan	0
aaa buat candu	0
aaa cantik banget	0
aaa jadi iri	0
aaa keren banget	0
aaa pengen	0
aaaa cayanggg bagus bgtt	0

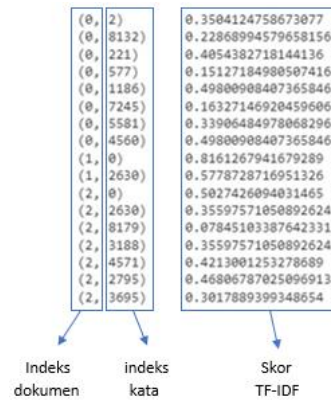
Gambar. 8 Data cyberbullying

zidan tempik	1
vidio sebagai ini tidak ada yg muji lo netizen udah benci lo	1
aaakkk banyak banget uang negara buat bayar asn	1
abi seorang ulama anak nya malah gak mamakai hijab hijab itu wajib dalam islam mbak	1
absen dari cepu	1
baca sampai selesai isinya bukan seperti yang anda bilang menteri kok gitu isin aku	1
ada kaca ga yah dirumah ny kok bisa se pd ituuu ngejekin musisi yg punya karya sendiri bukan cover an	1
ada masalah apa ya sama koq benci segitunya sama bb woy fachri gk ada ceritanya keluarga syahputra jumawa atau sombong ya semoga bb dilindungi dan di jauhkan sejauh jauhnya dari circle nya prik mental lo fachri lihat ber	1
ada tatto anjayyy	1
bagus lagu roma irama kau rusak pake cengkok kaya gitu kontol	1
ada yg aneh dengan dirimu kenapa dibikin miring miring bibirnya	1

Gambar. 9 Data non-cyberbullying

**D. Hasil Pembobotan TF-IDF**

Berikut hasil pembobotan TF-IDF dari penelitian ini.



Gambar. 10 Hasil pembobotan TF-IDF

1. Indeks dokumen merupakan representasi indeks dari corpus yang ada, contoh angka 0 adalah representasi kalimat pertama dari corpus, angka 1 adalah representasi kalimat kedua dari corpus dan seterusnya.
2. Indeks kata merupakan representasi indek dari feature names yang dihasilkan dari bag of words. Contohnya pada indeks ke-0 ada 2, 8123, 221, 557, 1186, ... dan 4560 mengartikan kalimat pada indeks ke-0 mengandung token indeks ke-2, ke-8123, ke-221, ke-557, ke-1186, ... dan ke-4560.
3. Skor TF-IDF merupakan representasi dari bobot setiap IDF, yang merupakan hasil kalkulasi dari *tfidfvectorizer*.

**E. Hasil Validasi**

Hasil validasi dari seluruh skenario pengujian ditampilkan pada Tabel III berupa skor rata-rata dari skor 10-Fold Cross Validation.

TABEL III  
HASIL VALIDASI

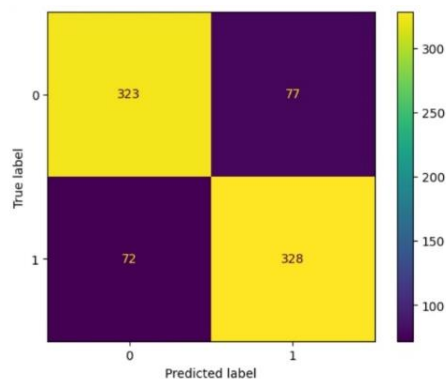
Mode l	Optimasi Hyperparamete r dengan Bayesian Optimization	Hyperparameter yang akan dioptimasi			Hasil 10-Fold Cross Validatio n
		Hidde n Layer	Learnin g Rate	Momentu m	
1	x	default	default	default	82,66%
2	✓	4	0,002	0,815	82,94%
3	✓	4	default	default	82,81%
4	✓	default	0,017	default	82,56%
5	✓	default	default	0,5	83,25%

**F. Hasil Evaluasi**

Berikut merupakan hasil evaluasi dari seluruh model yang dibangun. Seluruh hasil confusion matrix model yang dibangun dapat dilihat pada Tabel IV.

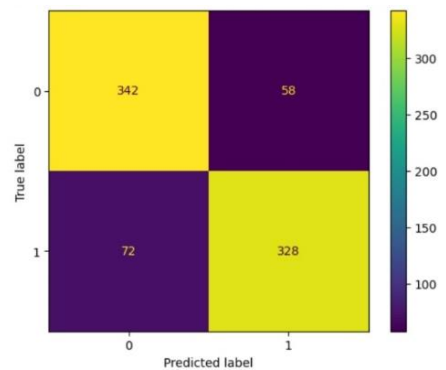
TABEL IV  
HASIL EVALUASI

Model	Accuracy	Precision	Recall	F1-Score
1	81,83%	81%	81%	81%
2	83,88%	84%	84%	84%
3	83,75%	84%	84%	84%
4	83,75%	84%	84%	84%
5	83,38%	84%	83%	83%



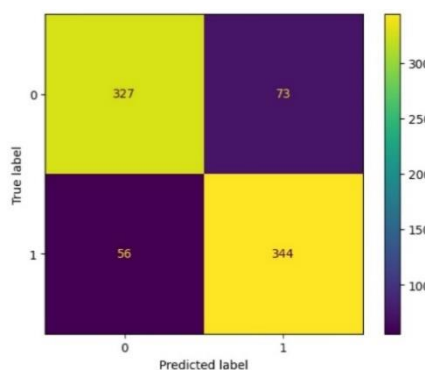
Gambar. 11 Confusion matrix model pertama

Confusion matrix pada model pertama (lihat Gambar 11) untuk mengevaluasi model yang dibangun, maka digunakan 4 alat ukur yaitu *accuracy* sebesar 81,375%, *precision* sebesar 81%, *recall* sebesar 81%, dan *f1-score* sebesar 8%. Berdasarkan Confusion matrix dari 800 data uji terdapat 323 data terprediksi benar tidak mengandung *cyberbullying* dan 77 data komentar yang tidak mengandung *cyberbullying* atau memiliki nilai target 0 namun terprediksi mengandung *cyberbullying* serta 72 data komentar yang mengandung *cyberbullying* atau memiliki nilai target 1 namun terprediksi tidak mengandung *cyberbullying* dan 328 data terprediksi benar mengandung *cyberbullying*.



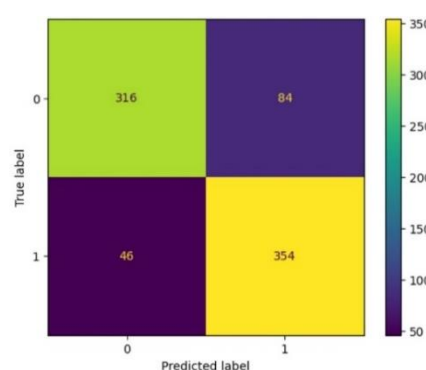
Gambar. 13 Confusion matrix model ketiga

Confusion matrix pada model ketiga (lihat Gambar 13) untuk mengevaluasi model yang dibangun, maka digunakan 4 alat ukur yaitu *accuracy* sebesar 83,75%, *precision* sebesar 84%, *recall* sebesar 84%, dan *f1-score* sebesar 84%. Berdasarkan Confusion matrix dari 800 data uji terdapat 342 data terprediksi benar tidak mengandung *cyberbullying* dan 58 data komentar yang tidak mengandung *cyberbullying* atau memiliki nilai target 0 namun terprediksi mengandung *cyberbullying* serta 72 data komentar yang mengandung *cyberbullying* atau memiliki nilai target 1 namun terprediksi tidak mengandung *cyberbullying* dan 328 data terprediksi benar mengandung *cyberbullying*.



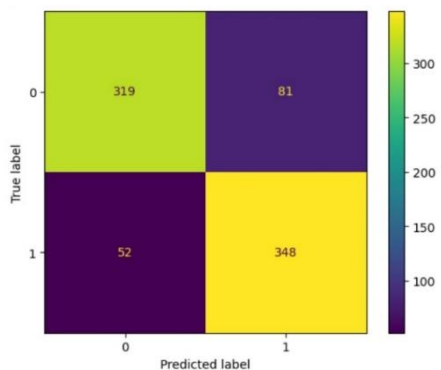
Gambar. 12 Confusion matrix model kedua

Confusion matrix pada model kedua (lihat Gambar 12) untuk mengevaluasi model yang dibangun, maka digunakan 4 alat ukur yaitu *accuracy* sebesar 83,88%, *precision* sebesar 84%, *recall* sebesar 84%, dan *f1-score* sebesar 84%. Berdasarkan Confusion matrix dari 800 data uji terdapat 327 data terprediksi benar tidak mengandung *cyberbullying* dan 73 data komentar yang tidak mengandung *cyberbullying* atau memiliki nilai target 0 namun terprediksi mengandung *cyberbullying* serta 56 data komentar yang mengandung *cyberbullying* atau memiliki nilai target 1 namun terprediksi tidak mengandung *cyberbullying* dan 344 data terprediksi benar mengandung *cyberbullying*.



Gambar. 14 Confusion matrix model keempat

Confusion matrix pada model keempat (lihat Gambar 14) untuk mengevaluasi model yang dibangun, maka digunakan 4 alat ukur yaitu *accuracy* sebesar 83,75%, *precision* sebesar 84%, *recall* sebesar 84%, dan *f1-score* sebesar 84%. Berdasarkan Confusion matrix dari 800 data uji terdapat 316 data terprediksi benar tidak mengandung *cyberbullying* dan 84 data komentar yang tidak mengandung *cyberbullying* atau memiliki nilai target 0 namun terprediksi mengandung *cyberbullying* serta 46 data komentar yang mengandung *cyberbullying* atau memiliki nilai target 1 namun terprediksi tidak mengandung *cyberbullying* dan 354 data terprediksi benar mengandung *cyberbullying*.



Gambar. 15 Confusion matrix model kelima

Confusion matrix pada model kelima (lihat Gambar 15) untuk mengevaluasi model yang dibangun, maka digunakan 4 alat ukur yaitu *accuracy* sebesar 83,25%, *precision* sebesar 84%, *recall* sebesar 83%, dan *f1-score* sebesar 83%. Berdasarkan Confusion matrix dari 800 data uji terdapat 319 data terprediksi benar tidak mengandung *cyberbullying* dan 81 data komentar yang tidak mengandung *cyberbullying* atau memiliki nilai target 0 namun terprediksi mengandung *cyberbullying* serta 52 data komentar yang mengandung *cyberbullying* atau memiliki nilai target 1 namun terprediksi tidak mengandung *cyberbullying* dan 348 data terprediksi benar mengandung *cyberbullying*.

G. Analisis

TABEL V  
HASIL ANALISIS

Model	Validasi (Score)	Evaluasi (Accuracy)	Waktu Pelatihan (Time)
1	82,66%	81,38%	10,203 menit/ 0,17 jam
2	82,94%	83,88%	137,74 menit/ 2,30 jam
3	82,81%	83,75%	443,229 menit/ 7,39 jam
4	82,56%	83,75%	135,82 menit/ 2,26 jam
5	83,25%	83,38%	589,99 menit/ 9,83 jam

Pada Tabel V disajikan data perbandingan validasi dan evaluasi dari seluruh model yang dibangun. Dari hasil tersebut, terdapat model yang mengalami penurunan dan peningkatan nilai evaluasi terhadap nilai validasi seperti dibawah ini:

1. Pada model pertama, nilai validasi yang diperoleh sebesar 82,66% dan setelah dievaluasi diperoleh nilai sebesar 81,38% atau menurun sebanyak 1,28%.
2. Pada model kedua, nilai validasi yang diperoleh sebesar 82,94% dan setelah dievaluasi diperoleh nilai sebesar 83,88% atau meningkat sebanyak 0,94%.
3. Pada model ketiga, nilai validasi yang diperoleh sebesar 82,81% dan setelah dievaluasi diperoleh nilai sebesar 83,75% atau meningkat sebanyak 0,94%.

4. Pada model keempat, nilai validasi yang diperoleh sebesar 82,56% dan setelah dievaluasi diperoleh nilai sebesar 83,75% atau meningkat sebanyak 1,19%.
5. Pada model kelima, nilai validasi yang diperoleh sebesar 83,25% dan setelah dievaluasi diperoleh nilai sebesar 83,38% atau meningkat sebanyak 0,13%.

Dari kelima model yang dibangun terdapat empat model dengan optimasi hyperparameter menggunakan Neural Network tidak mengalami overfitting yaitu model kedua, model ketiga, model keempat, dan model kelima, dimana model tersebut dilakukan optimasi hyperparameter menggunakan Bayesian Optimization. Sedangkan model yang mengalami *overfitting* adalah model pertama, yaitu model Neural Network tanpa optimasi hyperparameter.

Model yang dilakukan optimasi hyperparameter menggunakan Bayesian Optimization memiliki waktu eksekusi program yang lebih lama dibandingkan model Neural Network dengan default hyperparameter. Dimana model yang memiliki waktu eksekusi program paling lama adalah model kelima dengan hyperparameter yang dioptimasi yaitu momentum dengan lama waktu eksekusi adalah 589,99 menit atau 9,83 jam, kemudian yang kedua yaitu model ketiga dengan hyperparameter yang dioptimasi adalah hidden layer dengan lama waktu eksekusi adalah 443,229 menit atau 7,39 jam, selanjutnya yang ketiga yaitu model kedua dengan hyperparameter yang dioptimasi adalah hidden layer, learning rate, dan momentum dengan lama waktu eksekusi adalah 137,74 menit atau 2,30 jam, kemudian yang keempat yaitu model keempat dengan hyperparameter yang dioptimasi adalah learning rate dengan lama waktu eksekusi adalah 135,82 menit atau 2,26 jam, dan waktu eksekusi paling cepat terjadi pada model pertama, yaitu model Neural Network dengan default hyperparameter yang memiliki waktu eksekusi program dalam waktu 10,203 menit atau 0,17 jam.

TABEL VI  
DELTA NILAI HYPERPARAMETER

Model	Nilai hyperparameter setelah optimasi			Delta nilai default hyperparameter dengan nilai optimasi hyperparameter		
	Hidden layer	Learning rate	Momentum	Δhidden layer	Δlearning rate	Δmomentum
1	-	-	-	-	-	-
2	4	0,002	0,815	3	0,001	0,085
3	4	-	-	3	-	-
4	-	0,017	-	-	0,016	-
5	-	-	0,5	-	-	0,4

TABEL VII  
DELTA NILAI VALIDASI DAN EVALUASI

Model	Nilai validasi dan nilai evaluasi			
	Nilai validasi		Nilai evaluasi	
	Nilai validasi dengan optimasi hyperparameter	Δvalidasi	Nilai evaluasi dengan optimasi hyperparameter	Δevaluasi
1	-	-	-	-
2	82,94%	0,28%	83,88%	2,5%
3	82,81%	0,15%	83,75%	2,37%
4	82,56%	-0,01%	83,75%	2,37%
5	83,25%	0,59%	83,38%	2%

Pada tabel VI dan tabel VII disajikan data perbandingan delta nilai hyperparameter dan delta nilai validasi dan evaluasi dari seluruh model yang dibangun. Penjelasan dari hasil tersebut dapat dilihat dibawah ini:

1. Pada model pertama, nilai yang digunakan berupa nilai default yang kemudian hasil dari model ini akan dibandingkan dengan empat model lainnya untuk mendapatkan nilai selisih atau nilai delta.
2. Pada model kedua, selisih dari nilai hyperparameter yang didapatkan setelah optimasi pada hidden layer adalah 3, learning rate 0,001, dan momentum -0,085 (lebih kecil dari nilai default). Selisih nilai validasi yang diperoleh sebesar 0,28%. Model ini mengalami kenaikan akurasi dibanding model dengan default hyperparameter (model pertama) sebesar 2,5%.
3. Pada model ketiga, selisih dari nilai hyperparameter yang didapatkan setelah optimasi pada hidden layer adalah 3. Selisih nilai validasi yang diperoleh sebesar 0,15%. Model ini mengalami kenaikan akurasi dibanding model dengan default hyperparameter (model pertama) sebesar 2,37%.
4. Pada model keempat, selisih dari nilai hyperparameter yang didapatkan setelah optimasi pada learning rate adalah 0,016. Selisih nilai validasi yang diperoleh sebesar -0,01% (lebih kecil dibandingkan dengan yang dihasilkan oleh model dengan default hyperparameter). Model ini mengalami kenaikan akurasi dibanding model dengan default hyperparameter (model pertama) sebesar 2,37%.
5. Pada model kelima, selisih dari nilai hyperparameter yang didapatkan setelah optimasi pada momentum adalah -0,4 (lebih kecil dari nilai default). Selisih nilai validasi yang diperoleh sebesar 0,59%. Model ini mengalami kenaikan akurasi dibanding model dengan default hyperparameter (model pertama) sebesar 2%.

Hasil tersebut menunjukkan bahwa ketiga hyperparameter yang menjadi faktor penting dalam pelatihan algoritma backpropagation, yaitu hidden layer, learning rate, dan momentum ketika di optimasi secara bersamaan memiliki pengaruh besar dalam mencegah *overfitting*, menaikkan akurasi, dan memiliki waktu eksekusi yang baik daripada dioptimasi secara terpisah. Model yang memiliki performa paling baik pada penelitian ini adalah dimana nilai hidden layer yang besar, nilai learning rate kecil yang mendekati nilai default hyperparameter, dan nilai momentum yang besar namun tidak sebesar nilai default hyperparameter.

#### IV. KESIMPULAN

Algoritma Neural Network berhasil mengklasifikasi komentar *cyberbullying* Instagram. Hal tersebut dibuktikan dengan hasil evaluasi dari empat parameter yang digunakan, yaitu *precision*, *recall*, *F1-score* dan *accuracy* yang konsisten pada kisaran angka 80% pada seluruh model yang dibangun dan Bayesian Optimization dapat melakukan optimasi hyperparameter pada Neural Network. Optimasi hyperparameter pada Neural Network dengan Bayesian Optimization menunjukkan hasil yang lebih baik

daripada Neural Network tanpa menggunakan optimasi hyperparameter. Penggunaan optimasi hyperparameter menghasilkan kombinasi nilai hyperparameter yang berpengaruh dalam menaikkan nilai *accuracy* pada Neural Network. Hal tersebut dapat dilihat pada kenaikan akurasi dari model pertama ke model kedua sebesar 2,5%, model pertama ke model ketiga dan keempat masing-masing sebesar 2,37%, dan model pertama ke model kelima sebesar 2%. Model kedua menunjukkan kombinasi nilai hyperparameter yang memiliki performa yang lebih baik dari model lainnya yang dibangun dalam penelitian ini, dimana kombinasi nilai hyperparameternya yaitu hidden layer 4, nilai learning rate 0,002, dan nilai momentum 0,815. Optimasi hyperparameter dengan Bayesian Optimization dapat mencegah *overfitting* pada Neural Network. Hal tersebut dapat dilihat dari model kedua yang nilai evaluasi (*accuracy*) yang meningkat 0,94% dari nilai validasinya, model ketiga yang nilai evaluasi (*accuracy*) yang meningkat 0,94% dari nilai validasinya, model keempat yang nilai evaluasi (*accuracy*) yang meningkat 1,19% dari nilai validasinya, dan model kelima yang nilai evaluasi (*accuracy*) yang meningkat 0,13% dari nilai validasinya. Untuk penelitian selanjutnya dapat mencoba untuk menggunakan *tools* selain Google Colab untuk membandingkan performa eksekusi atau lama waktu eksekusi program, dan mencoba untuk menggunakan kalimat yang mengandung sarkasme dalam dataset yang digunakan, serta menambahkan kombinasi hyperparameter selain kombinasi hyperparameter yang telah dilakukan pada penelitian ini.

#### REFERENSI

- [1] A. M. Rizati, "Pengguna Internet di Indonesia Sentuh 212 Juta pada 2023," [Online]. Available: <https://dataindonesia.id/digital/detail/pengguna-internet-di-indonesia-sentuh-212-juta-pada-2023>. [Accessed 20 Februari 2023].
- [2] S Kemp, "Digital Indonesia: 2023," [Online]. Available: <https://datareportal.com>. [Accessed 20 Februari 2023].
- [3] A. S. Hutagalung, "Aplikasi Pendeteksi Cyberbullying Terhadap Komentar Postingan Media Sosial Instagram dengan Metode Naive Bayes Classifier Berbasis Website," *Jurnal Sistem dan Teknologi Informasi (Justin)*, vol. 9, no. 3, pp. 364-371, 2021.
- [4] A. Rachmatullah, "Optimasi Hyperparameter pada Gradient Boosted Trees Menggunakan Bayesian Optimization," 2019.
- [5] F. Sajid, "Perbandingan Algoritma Klasifikasi terhadap Emosi Tweet Berbahasa Indonesia," *Jurnal Sistem dan Teknologi Informasi (Jepin)*, vol. 7, no. 2, pp. 242-249, 2021.
- [6] W. A. Luqyana, "Analisis Sentimen Cyberbullying pada Komentar Instagram dengan Metode Klasifikasi Support Vector Machine," *Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer e-ISSN: 2548-964X*, vol. 2, no. 11, pp. 4704-4713, 2018.
- [7] I. T. A. Nur, "Perbandingan Performa Metode Klasifikasi SVM, Neural Network, dan Naive Bayes untuk Mendeteksi Kualitas Pengajuan Kredit di Koperasi Simpan Pinjam," *Jurnal Teknologi Informasi dan Ilmu Komputer*, vol. 6, no. 4, p. 444, 2019.
- [8] M. H. Sultan, "Optimasi Parameter Neural Network Pada Data Time Series (Studi Kasus Gempa Bumi di Maluku Utara)," *Ps Its*, vol. 3, no. 2, pp. 59-71, 2014.
- [9] Van Rijn, Jan N., "Hyperparameter importance across datasets," *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 2367-2376, August 2018.



- [10] P. Murugan, "Hyperparameter Optimization in Deep Convolutional Neural Network/Bayesian Optimization with Gaussian Prior Process," December 2017.
- [11] I. Kholis, S. Alam, "Analisis Variasi Parameter Backpropagation Artificial Neural Network dann Principal Component Analysis Terhadap Sistem Pengenalan Wajah," *Electrans*, vol. 14, no. 1, pp. 12-19, 2016.
- [12] I. Agustina, "Analisis Pengembangan Model Prediksi Kesuksesan Kickstrarter Menggunakan Algoritma Backpropagation Artificial Neural Network (ANN) da Random Forest," 2022.
- [13] H. Najjichah, A. Syukur, dan H. Subagyo, "Pengaruh Text Preprocessing dan Kombinasinya pada Peringkat Dokumen Otomatis Teks Berbahasa Indonesia," *Jurnal Teknologi Informasi*, vol. 15, no. 1, pp. 1-11, 2019.
- [14] H. marifah, A. P. Wibawa, dan M. I. Akbar, "Klasifikasi artikel ilmiah dengan berbagai skenario preproceasing," *Sains, Aplikasi, Komputasi, dan Teknologi Informasi*, vol. 2, no. 2, pp. 70-78, 2020..
- [15] R. D. Syarif, "Identifikasi Cyberbullying Pada Komentar Instagram Menggunakan Metode Lexicon-based Dan Naïve Bayes Classifier (studi Kasus: Pemilihan Presiden Indonesia Tahun 2019)," *eProceeding of Engineering*, vol. 6, no. 2, pp. 8838-8851, 2019.
- [16] Y. Ramdhani, "Penerapan Algoritma Neural Network Untuk Klasifikasi Kardiotokografi," *Jurnal Informatika*, vol. 5, no. 1, pp. 43-49, 2018.
- [17] Skcikit-learn.org, "Neural Network Models(Supervised)" [Online]. Available: <https://skcikit-learn.org>. [Accessed 20 Februari 2023].
- [18] J. Snoek, H. Larochelle, R. Adams, "Practical Bayesian Optimization of Machine Learning Algoritms," *Advances in Neural Information Processing System*, vol 4, pp. 2951-2959, 2012.
- [19] S. Morar, M.T., Knowles,J., dan Sampaio, " Initialization of Bayesian Optimization Viewed as Part of a Larger Algoritm Potofolio," 2017.
- [20] I. Dewancker, M. McCourt, S. Clark et al. "Evaluation System for a Bayesian Optimization Service," 2016.