

## Klasifikasi *Tweet Cyberbullying* dengan Menggunakan Algoritma *Random Forest*

Stephen Kurnia<sup>1</sup>, Harvey Guharelvino Prabowo<sup>2</sup>, Berli Suharmanto<sup>3</sup>,  
Muhammad Nabil Nufail Pribadi<sup>4</sup>, Nurul Chamidah<sup>5</sup>,  
Informatika / Fakultas Ilmu Komputer  
Universitas Pembangunan Nasional Veteran Jakarta  
Jl. R.S Fatmawati No. 1 Pondok Labu Jakarta Selatan 12450  
stephenk@upnvj.ac.id<sup>1</sup>, harveygp@upnvj.ac.id<sup>2</sup>, berls@upnvj.ac.id<sup>3</sup>, muhammadnnp@upnvj.ac.id<sup>4</sup>,  
nurul.chamidah@upnvj.ac.id<sup>5</sup>

**Abstrak.** *Cyberbullying* adalah suatu aktivitas perundungan yang dilakukan di media online, banyak motif yang mendasari terjadinya kegiatan *cyberbullying*. Diantaranya adanya perbedaan status sosial, kritik terhadap pemerintah, dan banyak faktor lainnya. Twitter merupakan media sosial yang menjadi salah satu tempat terjadinya *cyberbullying*. Hal ini dikarenakan Twitter memiliki pengguna yang memiliki latar belakang yang sangat berbeda, dari budaya, sampai dengan pandangan politik. Dengan luasnya kebebasan berpendapat dan semakin berkembangnya penggunaan sosial media, dibuat penelitian klasifikasi *Tweet cyberbullying* dengan menggunakan algoritma *Random Forest*. Berdasarkan penelitian terhadap dataset yang diperoleh, dihasilkan akurasi sebesar 93,3% dengan kedalaman maksimum 350, menggunakan 100 pohon, bobot kelas yang tidak diberikan, jumlah daun minimum sebanyak 1, dan fitur maksimal bernilai akar 2. Sedangkan dengan kedalaman maksimum sebesar 200, menggunakan 250 pohon, bobot kelasnya seimbang, banyak daun minimum sebanyak 3, dan fitur maksimal bernilai  $\log_2$  menghasilkan akurasi sebesar 93,2%. Berdasarkan hasil akurasi yang didapat, semakin besar kedalaman pohon semakin besar juga nilai akurasi yang dihasilkan.

**Kata Kunci :** *Cyberbullying*, *Random Forest*, *Twitter*, Klasifikasi

### 1 Pendahuluan

Dewasa ini banyak sekali masalah-masalah yang berhubungan dengan UU ITE. Salah satu permasalahan ini adalah *cyberbullying* yang kerap terjadi kepada orang-orang. Permasalahan ini sangat meresahkan sebab banyak korban yang akhirnya depresi dan bunuh diri seperti artis-artis terkenal yang menghilangkan nyawanya setelah menjadi korban *cyberbullying* oleh banyak pihak.

Media sosial yang dikenal sebagai media sosial yang paling rawan dengan *cyberbullying* adalah *twitter*. Media sosial ini pertama kali muncul di internet sebagai sarana untuk berinteraksi orang-orang dari manapun dan kapanpun di seluruh dunia. Media sosial ini biasa digunakan oleh *public figure* sebagai *fanbase environment* [1].

Banyaknya orang yang menggunakan media sosial ini memungkinkan untuk memicu pelecehan berupa *posting tweet* kasar dan perundungan. Orang dari berbagai kalangan dapat mengakses media sosial ini dengan sangat mudah. Hanya dengan terhubung dengan internet semua orang dapat mengaksesnya.

*Cyberbullying* ini sudah sangat meresahkan netizen dunia maya [2]. Dengan hal ini diperoleh keinginan untuk mengklasifikasi *tweet* dari yang diambil secara acak dari *open dataset*. Data yang diambil terdapat 47692 data *tweet* yang diambil.

Dalam hal ini, penelitian ini akan mencoba menggunakan algoritma *Random Forest* untuk melakukan proses klasifikasinya. Karena data-data yang diterima masih sangat tidak teratur akan dilakukan *preprocess* terlebih dahulu. Kemudian akan dilanjutkan ke klasifikasi dengan *Random Forest*.

Suatu populasi dibagi menjadi segmen-segmen seperti cabang yang membentuk pohon terbalik dengan simpul akar, simpul internal, dan simpul daun. Algoritma non-parametrik, yang berarti dapat menangani kumpulan data yang besar dan kompleks tanpa memaksakan kerangka kerja parametrik yang kompleks. Data studi dapat dipisahkan menjadi set data pelatihan dan validasi setelah ukuran sampel cukup besar. Membangun model *Decision Tree* dengan dataset pelatihan dan menentukan ukuran pohon yang tepat dengan dataset validasi untuk menghasilkan model akhir terbaik [3].

Sebelum penelitian ini dilakukan terdapat beberapa penelitian yang menggunakan dataset yang sama dengan metode yang berbeda. Berdasarkan jurnal yang menggunakan metode SVM untuk mendeteksi *cyberbullying* dengan melakukan klasifikasi dan dalam penggunaan SVM didapatkan akurasi sebesar 76.2% [4].

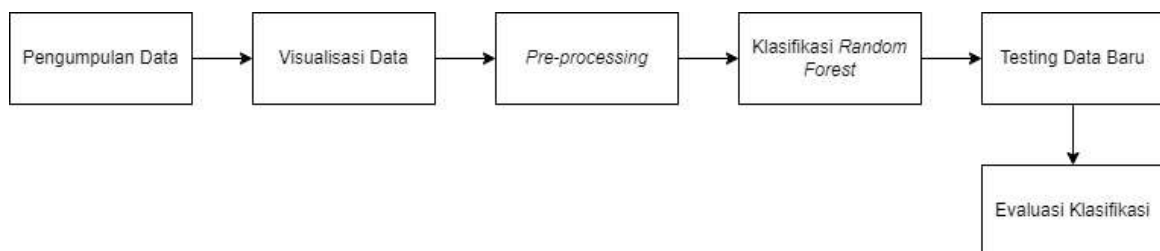
Kemudian jurnal yang kedua menggunakan metode Naive Bayes dan Fuzzy Logic untuk mendeteksi *cyberbullying* dengan melakukan klasifikasi dan dalam pencarian dengan mendapatkan akurasi sebesar 88.76% [5].

Berikutnya pada jurnal ketiga yang menggunakan metode *Decision Tree* untuk mendeteksi *cyberbullying* menggunakan data post dari twitter, didapatkan hasil akurasi sebesar 69.22% [6].

Pada penelitian terkait di atas bahwa nilai akurasi yang didapatkan masih belum mencapai angka 90% ketika menggunakan algoritma-algoritma yang dicantumkan di atas. Dari penelitian yang akan dilakukan ini diharapkan dapat mendapatkan akurasi yang lebih tinggi dari penelitian-penelitian sebelumnya.

## 2 Metodologi Penelitian

Metodologi penelitian adalah salah satu proses yang dilakukan untuk memperoleh data dan mengumpulkan data yang nantinya akan dijadikan penelitian yang bersifat ilmiah. Detail metodologi yang digunakan dapat dilihat sebagai berikut.



**Gambar 2.1** Alur Penelitian

Gambar 2.1 menyajikan alur penelitian yang terdiri atas tahapan pengumpulan data, visualisasi data, praproses data, klasifikasi data dengan menggunakan *Random Forest Classifier*, kemudian dengan tambahan Testing data baru yang kita ambil dari twitter secara acak, serta tahapan terakhir yakni evaluasi klasifikasi. Diharapkan hasil yang dapat diperoleh dari penelitian ini adalah klasifikasi *tweet cyberbullying* dengan menggunakan Algoritma *Random Forest Classifier*.

### 2.1 Pengumpulan Data

Data diambil data dari kaggle.com (<https://www.kaggle.com/datasets/andrewmvd/cyberbullying-classification>) mengenai *cyberbullying*. Pengambilan data ini diambil setelah mencari penelitian-penelitian sebelumnya dengan tema yang sama. Dari penelitian yang sudah ada tersebut diperoleh bahwa belum ada penelitian yang menggunakan algoritma *Random Forest* dalam pengerjaannya.

### 2.2 Visualisasi Data

Visualisasi ini menampilkan kata-kata terbanyak yang muncul pada dataset untuk dapat dievaluasi pada saat tahapan praproses [7]. Dapat dilihat kata-kata dan jumlahnya sebagai diagram batang untuk mengetahui kata-kata yang sering muncul.

Selain itu visualisasi di sini juga bisa menampilkan perbandingan data tiap-tiap kelas. Hal ini bertujuan untuk menghindari terjadinya *imbalance* data yang akan mempengaruhi hasil dari klasifikasi. Dari pengecekan jumlah kelas dari visualisasi didapatkan data sudah seimbang.

### 2.3 Preprocessing

Tahap *preprocessing data* yang dilakukan pada penelitian ini adalah NLP (*Natural Language Processing*) [8]. NLP yang digunakan untuk penelitian kali ini adalah penghilangan *stopword*, *stemming*, *sentiment analysis* [9], penghilangan simbol dan emoji, serta vektorisasi *tweet* yang telah dibersihkan.

Natural Language Processing (NLP) adalah sekelompok teknik yang digunakan untuk memungkinkan komputer memahami bahasa alami yang belum diproses yang diucapkan oleh orang-orang dengan mengumpulkan struktur tata bahasa dan makna dari input [10].

Tahapan-tahapan *preprocessing data* terbagi menjadi beberapa alur yang berurutan. Mulai dari mengecek null dan data duplikat, menghilangkan emoji, menghilangkan *string* yang tidak teratur, mengganti kata singkatan dan simbol. Proses tersebut dilakukan terlebih dahulu untuk membuat *stemming* lebih efektif dan diteruskan dengan menghilangkan spasi berlebih dari proses *stemming* untuk dilakukan analisis sentiment dan akhirnya dimasukkan ke kolom baru. Berikut alur dari tahapan praproses data dan penjelasannya:

a. Mengecek jumlah *missing null*

Pengecekan *missing null* dapat dilakukan dengan menggunakan fungsi yang sudah disediakan oleh pandas. Setelah didapatkan jumlahnya, data yang kosong atau juga data yang berbeda sendiri tipe datanya dapat langsung dihilangkan dari *dataframe*. Dan pada data ini sudah tidak terdapat data yang bernilai kosong.

b. Mengecek duplikasi data

Pengecekan duplikasi data ini berguna untuk mengurangi jumlah data yang akan diproses. Hal ini karena data yang duplikat hanya akan memperlama proses dari latih data dan tidak berpengaruh apa-apa karena sudah ada data yang sama sebelumnya. Data yang duplikat ini dapat dihapus di tahapan *preprocessing* untuk meringankan sumber daya perangkat pada saat latih data dilakukan. Di sini mendapatkan 36 duplikasi data dan langsung dihapus dari *dataframe*.

c. Mengecek tipe data tiap kolom

Pengecekan tipe data dapat dilakukan dengan fungsi yang akan menampilkan semua nama kolom dan tiap-tiap tipe datanya. Pengecekan ini berguna untuk menentukan mana fitur yang kategorikal dan numerik sebelum dilakukan proses latih data.

d. Menghilangkan emoji

Emoji merupakan sebuah karakter yang tidak penting untuk diproses di dalam *text processing* pada *machine learning*. Penghilangan emoji ini dapat dilakukan dengan cara mengubah format emoji dengan sebuah *string* kosong.

e. Menghilangkan *string* yang tidak teratur

*String* yang tidak teratur di sini adalah bentuk *string* yang aneh atau tidak biasa dapat berupa *typo*, *url*, *id*, *http*, dan lain sebagainya. Penghilangan *string* ini bertujuan untuk mendapatkan kata-kata yang dapat dikenali saja yang bisa diproses dan dilakukan analisis sentimen.

f. Melakukan perubahan kata singkat menjadi kata baku

Karena *tweet* yang diambil adalah *tweet* dengan bahasa inggris, maka tidak jarang banyak sekali istilah singkatan seperti “*will*” menjadi “*ll*”. Singkatan ini dapat diubah dengan cara mengganti format dengan menjadi kata bakunya.

g. Menghilangkan simbol

Simbol-simbol tidak berguna untuk melakukan *text processing* karena hanya akan memperpanjang data saja dan tidak akan mempengaruhi hasil dan akan membuat analisis sentimen menjadi lebih rumit.

h. Melakukan *stemming*

Proses *stemming* ini adalah menghilangkan imbuhan tiap-tiap kata yang terdapat pada *dataframe*. Hal ini bertujuan untuk mempersempit kata-kata yang digunakan karena jika tidak dilakukan *stemming* akan terdapat banyak jenis kata yang positif dan negatif meskipun kata itu memiliki induk kata yang sama. Dengan begitu proses analisis sentimen akan menjadi lebih mudah dan cepat [11].

i. Menghilangkan spasi yang berlebih

Proses ini bertujuan untuk mengurangi ukuran *string* tiap data karena barangkali terdapat salah ketik berupa spasi yang berlebih sehingga memperpanjang data.

j. Analisis Sentimen

Proses ini dilakukan dengan cara memilah kata yang terdapat pada kelas-kelas. Proses ini bertujuan untuk menentukan tiap-tiap kata positif atau negatif. Dari yang positif akan menjadi data yang bukan *cyberbullying* dan sisanya tetap dikelasnya masing-masing[12].

k. Memasukkan data yang dibersihkan menjadi kolom baru

Proses ini bertujuan untuk tidak merubah dataset yang asli dan hanya menambahkan kolom baru pada *dataframe* bukan di datasetnya. Hal ini bertujuan untuk mempermudah memasukkannya sebagai fitur *input*.

## 2.4 Klasifikasi dengan *Random Forest*

Karena dataset yang digunakan hanya terdapat dua kolom, maka ditambahkan dalam *dataframe* yang telah data *tweet* yang telah dibersihkan sebagai kolom tambahan. Data *tweet* bersih sebagai X dan tipe *cyberbullying* sebagai Y. Pembagian data yang dilakukan dengan membagi data latih dan data uji dengan perbandingan 4:1.

*Random Forest* terdiri dari algoritma klasifikasi yang berbeda dengan pohon keputusan atau struktur pohon. Metode yang digunakan oleh *Random Forest* dikenal sebagai *bootstrap* atau agregasi, biasa disebut sebagai *bagging*. Dengan *bootstrap* dari sampel pelatihan dan variabel *input* pada setiap node, *Random Forest* membuat beberapa pohon keputusan dengan himpunan bagian yang dipilih secara acak [13].

Algoritma *Random Forest* dibangun dalam serangkaian langkah yang melibatkan distribusi data pelatihan dan pengujian dengan perbandingan dalam persen (%), menggunakan pohon untuk menghasilkan nilai acak terbaik, mencari tahu jumlah maksimum fitur untuk pengacakan data, dan membangun pohon keputusan dan memprediksi nilai mayoritas untuk setiap kelas. Hasil ini kemudian dibandingkan dengan yang diprediksi oleh algoritma terkait pelabelan data [14].

Proses *modeling* pada penelitian ini adalah proses latih data dengan algoritma *Random Forest*, namun dibuat dua model yang berbeda. Maksudnya adalah dibuat beberapa model dengan algoritma *Random Forest* dengan perbedaan mulai dari kedalaman pohon, daun, bobot kelas, dan fitur maksimum [15].

## 2.5 Testing Data Baru

Pada tahapan ini diambil data secara acak dari *twitter* yang sedang *trending*. Data yang baru ini bertujuan untuk menguji apakah hasil prediksi dari model cukup valid untuk digunakan sebagai klasifikasi. Data ini akan diprediksi dan dilihat bagaimana hasil prediksinya.

## 2.6 Evaluasi Klasifikasi

Selanjutnya, tahap yang dilakukan adalah proses Evaluasi Klasifikasi. Tahap ini dapat dilakukan dengan cara membandingkan *confusion matrix* dari kedua model yang telah dibuat [16]. Seberapa beda dan jauh *confusion matrix*-nya yang akan menjadi kesimpulan dari penelitian. Acuan yang diukur dari evaluasinya adalah nilai *true positive* yang didapat dari *confusion matrix* kedua model, model yang memiliki nilai *true positive* yang lebih banyak merupakan kesimpulan dari klasifikasi kedua model.

Setelah itu mengambil data baru yang digunakan adalah data yang didapat pada *twitter* secara acak. Pada tahap ini dilakukan pengujian apakah data baru yang masuk memiliki hasil prediksi yang sesuai ketika menggunakan model yang telah dibuat. Dan hasil dari uji data baru ini memperlihatkan jawaban prediksi yang sedikit berbeda.

## 3 Hasil dan Pembahasan

Hasil dari penelitian ini merupakan pengujian data yang pada tahap klasifikasi menggunakan metode yang telah ditentukan yaitu algoritma *Random Forest Classifier*. Pengujian dilakukan untuk menentukan klasifikasi tweet cyberbullying. Dataset yang digunakan akan dibagi menjadi data latih dan juga data uji.

### 3.1 Pengumpulan Data

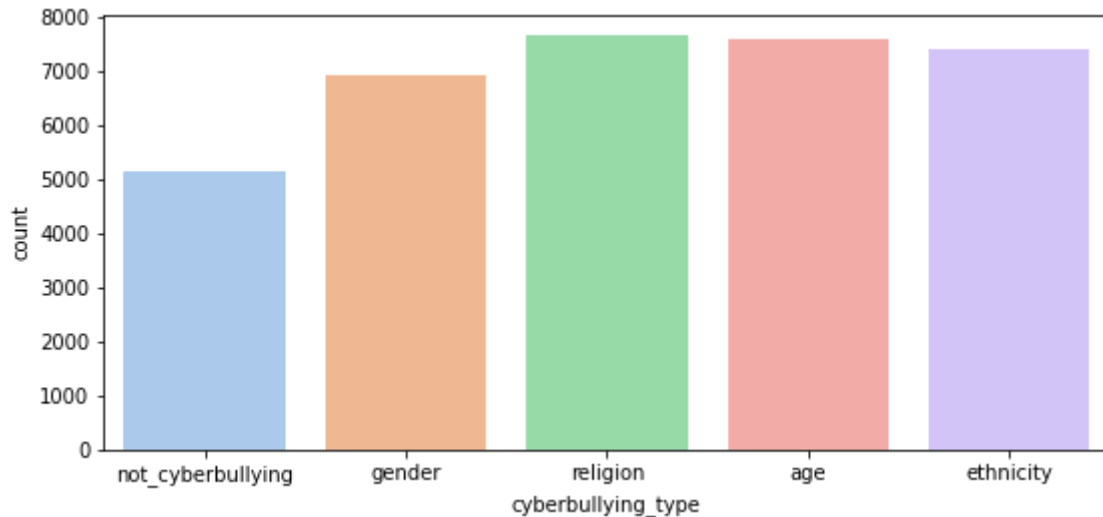
Pengumpulan data dilakukan dengan mengambil data dari website [www.kaggle.com](http://www.kaggle.com) dengan judul dataset *cyberbullying tweets* Dengan jumlah data sebanyak 47.692 baris dan 2 kolom. Data ini berisi dengan *tweets* yang diambil secara acak dari *twitter*. Berikut nama kolom dan keterangannya pada table 3.1.

**Tabel 3.1** Atribut dataset

Tweet	Data <i>tweet</i> dari <i>twitter</i> secara acak
<i>Cyberbullying Type</i>	Tipe <i>cyberbullying</i>

### 3.2 Visualisasi Data

Dapat dilihat dari visualisasi dibawah ini bahwa dataset yang digunakan memiliki jumlah yang signifikan antara kelas yang satu dengan kelas yang lainnya.



**Gambar 3.1** Perbandingan antar kelas

Gambar 3.1 menyajikan total data tiap kelas memiliki jumlah yang sudah cukup seimbang. Dalam hal ini data sudah tidak memerlukan perbaikan. Dari data yang sudah seimbang ini dapat langsung ke tahapan praproses.

### 3.3 Preprocessing

#### 3.3.1 Mengecek Null dan Menghilangkan Data yang Sama

Tahapan ini dilakukan untuk menghilangkan data *tweets* yang duplikat, serta mengecek apakah terdapat nilai null pada dataset yang akan digunakan untuk pemodelan. Dan pada data ini sudah tidak terdapat data yang bernilai null. Setelah itu mendapatkan 36 duplikasi data dan langsung dihapus dari *dataframe*.

#### 3.3.2 Mengganti Kata Singkatan

Pada tahapan ini mengganti kata singkatan bahasa Inggris. Di sini dilakukan secara manual dengan cara mengganti kata yang disingkat dengan kata aslinya agar dapat mengetahui apakah diperlukan atau tidak untuk diproses.

#### 3.3.3 Melakukan Penghilangan *Stopword*

Pada tahapan ini dilakukan penghilangan imbuhan dari kata-kata bahasa Inggris dan menghilangkan kata *stopword* atau kata yang tidak mempunyai makna seperti “*the*” dan “*very*”. Kemudian *stopword* tersebut diganti dengan menggunakan *string* kosong.

#### 3.3.4 Menghapus Karakter Aneh dan Tidak Beraturan

Tahapan ini dilakukan karena data yang diperoleh memiliki karakter-karakter yang tidak semuanya rapih dan masih berantakan. Data yang tidak rapih ini biasa didapat dari *spammer* ataupun bot. karakter-karakter ini hanya perlu dihapus dengan diganti dengan *string* kosong.

#### 3.3.5 Melakukan *Stemming*

Pada tahapan ini dilakukan menghilangkan imbuhan tiap-tiap kata yang terdapat pada *dataframe*.

#### 3.3.6 Menghilangkan Spasi yang Berlebih

Tahapan ini dilakukan untuk menghilangkan spasi yang berlebih karena salah ketik. Penghilangan spasi ini dilakukan dengan cara menggantinya dengan *string* kosong.

#### 3.3.7 Menghilangkan Emoji

Pada tahapan ini menghilangkan emoji karena emoji tidak dapat digunakan untuk *text processing*. Penghilangan emoji ini dilakukan dengan cara mengganti emoji dengan *string* kosong.

#### 3.3.8 *Text Cleaning* dan Kolom Baru

Pada tahapan ini menyatukan semua tahapan di atas menjadi satu fungsi secara berurutan dan kemudian dijalankan dengan memerlukan waktu setidaknya tiga menit. Setelah dijalankan semua *text* yang telah dibersihkan dimasukkan menjadi kolom baru untuk memudahkan pembagian data.

### 3.3.9 Pembagian Data

Setelah dilakukan praproses dihasilkan data-data yang sudah bersih dan dibagi dengan menggunakan *split data*. Kolom yang digunakan adalah kolom *cyberbullying type* sebagai Y dan kolom baru *clean tweet* sebagai X. Pembagian data ini memiliki perbandingan dengan data latih 80% dan data uji 20% dari total 34737 data.

### 3.3.10 Vektorisasi input

Data *input* dibuat menjadi vektor untuk dilakukan proses fitting. Hal ini karena *string* yang panjang tidak dapat dibuat sebagai *input* sehingga akan menyebabkan *error*. Dengan menjadikan *string* menjadi vektor dengan Tf-Idf yang merupakan fungsi untuk melakukan vektorisasi *text processing* dan selanjutnya akan dapat dilakukan proses latih data.

## 3.4 Klasifikasi *Random Forest*

Setelah dilakukan proses klasifikasi, kemudian dilanjutkan ke tahap evaluasi. Evaluasi klasifikasi yang digunakan berupa perbandingan antara dua model yang berbeda mulai dari kedalamannya, jumlah pohon, dan daunnya.

Hasil yang didapatkan dengan kedalaman maksimum 350, menggunakan 100 pohon, bobot kelas yang tidak diberikan, jumlah daun minimum sebanyak 1, dan fitur maksimum bernilai akar 2 menghasilkan model yang memiliki akurasi sebesar 93,3%. Sedangkan dengan kedalaman maksimum sebesar 200, menggunakan 250 pohon, bobot kelasnya seimbang, banyak daun minimum sebanyak 3, dan fitur maksimal bernilai log2 menghasilkan akurasi sebesar 93,2%.

Jika dilihat nilai akurasi keduanya, seperti tidak begitu berpengaruh namun pada nyatanya akan berpengaruh pada saat memasukan data baru. Kedua model yang dibuat memiliki perbedaan yang sedikit sekali, yaitu hanya pada akurasinya.

**Tabel 3.2** Model pertama

<b>Model 1</b>	<b>0</b>	<b>1</b>	<b>2</b>	<b>3</b>	<b>4</b>
<i>Support</i>	1516	1481	1384	1032	1535
<i>Precision</i>	0.97	0.99	0.96	0.76	0.96
<i>F1-Score</i>	0.98	0.98	0.91	0.81	0.95
<i>Recall</i>	0.98	0.98	0.86	0.87	0.95
<i>Accuracy</i>	0.933				

Pada tabel 3.2 model pertama memiliki hasil yang sudah cukup tinggi. Dapat dilihat bahwa mayoritas hasil *precision* tiap kelas yang didapat sudah di atas 0,90 walaupun terdapat satu kelas yang masih dibawah 0,80. Begitu juga dengan *f1-score* yang didapat, kelas ketiga merupakan kelas yang memiliki *f1-score* yang paling rendah. Sedangkan *recall* memiliki hasil kelas kedua dengan nilai terendah. Hasil model pertama sudah cukup baik karena sudah memiliki akurasi dengan nilai lainnya yang setara.

**Tabel 3.3** Model kedua

<b>Model 2</b>	<b>0</b>	<b>1</b>	<b>2</b>	<b>3</b>	<b>4</b>
<i>Support</i>	1516	1481	1384	1032	1535

<i>Precision</i>	0.97	0.99	0.96	0.76	0.96
<i>F1-Score</i>	0.98	0.98	0.91	0.81	0.95
<i>Recall</i>	0.98	0.98	0.86	0.87	0.95
<i>Accuracy</i>	0.932				

Pada tabel 3.3 dapat dilihat bahwa hasil dari kedua model ternyata memiliki nilai *support*, *precision*, *f1-score*, dan *recall* yang identik. Walaupun sebagian besar identik, tetapi memiliki akurasi yang sedikit berbeda. Dalam hal ini perbedaan 0,1% dapat mempengaruhi hasil dari prediksi data yang akan diprediksi.

### 3.5 Testing Data Baru

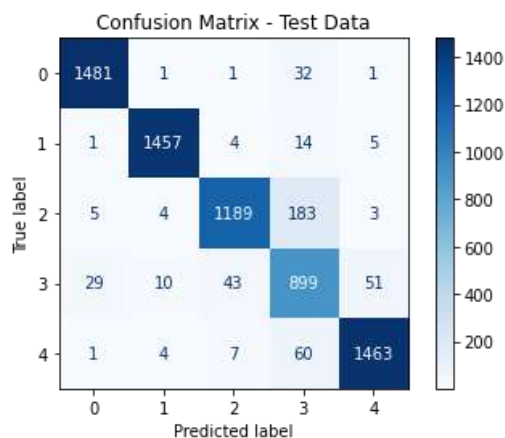
Pada testing dengan data baru akan menguji model pertama dan kedua menggunakan data yang diperoleh secara acak dari *twitter*. Tahap ini bertujuan untuk membuktikan bahwa perbedaan sebesar 0,1% akan mempengaruhi hasil dari klasifikasi. Berikut perbandingan hasil prediksi kedua model.

**Tabel 3.4** Perbandingan Prediksi Data Baru

	Kelas					
<b>Model 1</b>	3	4	1	4	0	4
<b>Model 2</b>	3	4	1	1	0	4

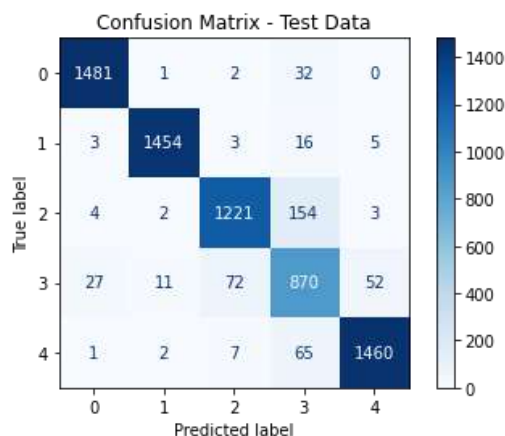
Dapat dilihat pada tabel di atas terdapat perbedaan dari salah satu data baru yang diprediksi. Perbedaan satu prediksi di atas membuktikan bahwa perbedaan sebesar 0,1% mempengaruhi hasil dari prediksi. Dari sini dapat disimpulkan bahwa mengambil yang akurasinya lebih besar sekecil apapun perbedaannya menjadi prioritas.

### 3.6 Evaluasi Klasifikasi



**Gambar 3.2** Confusion Matrix Model 1

Pada gambar 3.2 dapat dilihat *confusion matrix* dari model pertama memiliki nilai *true positive* yang beragam tiap kelasnya. Kelas ke-0 memiliki 1481 dari keseluruhan data kelasnya, kelas ke-1 memiliki 1457, kelas ke-2 memiliki 1189, kelas ke-3 memiliki 899, dan kelas ke-4 memiliki 1463. Jika dilihat dari tabel klasifikasi sudah terlihat memang kelas ke-3 memiliki nilai *precision* paling rendah menyebabkan hasil prediksi yang benar menjadi kurang sebanding dengan hasil prediksi kelas lainnya. Selain itu, kelas ke-2 juga memiliki kesamaan yang hampir mirip dengan kelas ke-3. Kelas ke-2 memiliki *recall* terendah dari kelas lain dan membuatnya terprediksi benar sedikit kurang dari tiga kelas lainnya.



**Gambar 3.3** Confusion Matrix Model 2

Pada gambar 3.3 dapat dilihat *confusion matrix* dari model kedua memiliki nilai *true positive* yang beragam tiap kelasnya. Kelas ke-0 memiliki 1481 dari keseluruhan data kelasnya, kelas ke-1 memiliki 1454, kelas ke-2 memiliki 1221, kelas ke-3 memiliki 870, dan kelas ke-4 memiliki 1460. Dapat dilihat terjadi kenaikan *true positive* pada kelas ke-2 yang sebelumnya hanya 1189. Meskipun terdapat kenaikan pada kelas ke-2, kelas-kelas lain selain kelas ke-0 dan kelas ke-2 mengalami penurunan. Penurunan yang paling signifikan terjadi pada kelas ke-3. Hal ini dapat membuktikan bahwa perbedaan akurasi sebesar 0,1% dapat mempengaruhi klasifikasi.

#### 4 Kesimpulan

Dari hasil implementasi program klasifikasi *tweet cyberbullying* di *twitter* dengan menggunakan algoritma *Random Forest Classifier*, didapatkan hasil bahwa dari data yang digunakan berhasil ditentukan *tweet* yang sesuai dengan kategori yang diinginkan. Dari hasil implementasi yang dilakukan, dapat ditemukan tingkat *tweet cyberbully* paling banyak untuk kata *tweet* “bulli” sebanyak lebih dari 10.000 kata.. Dari penelitian dan implementasi yang peneliti lakukan, peneliti dapat membuat model klasifikasi untuk menentukan suatu data *tweet* yang berpotensi adalah *cyberbullying*. Dan semakin dalam pohonnya, maka semakin akurat hasil klasifikasinya tetapi waktu eksekusi latih terhadap data yang digunakan akan memakan waktu yang lebih lama.

Selanjutnya, setelah dilakukan klasifikasi, dilanjutkan dengan evaluasi klasifikasi dengan Hasil yang didapatkan adalah dengan kedalaman maksimum 350, menggunakan 100 pohon, bobot kelas yang tidak diberikan, jumlah daun minimum sebanyak 1, dan fitur maksimum bernilai akar 2 menghasilkan model yang memiliki akurasi sebesar 93,3%. Sedangkan dengan kedalaman maksimum sebesar 200, menggunakan 250 pohon, bobot kelasnya seimbang, banyak daun minimum sebanyak 3, dan fitur maksimal bernilai log2 menghasilkan akurasi sebesar 93,2%.

#### 5 Referensi

- [1] R. Bayari and A. Bensefia, “Text mining techniques for cyberbullying detection: State of the art,” *Adv. Sci. Technol. Eng. Syst.*, vol. 6, no. 1, pp. 783–790, 2021, doi: 10.25046/aj060187.
- [2] S. C. R. Kumar, S. Parakh, and C. N. . V. Kumar, “Detection of Cyberbullying using Machine Learning,” *Int. J. Res. Appl. Sci. Eng. Technol.*, vol. 8, no. 7, pp. 1231–1240, 2020, doi: 10.22214/ijraset.2020.30403.
- [3] A. Muneer and S. M. Fati, “A comparative analysis of machine learning techniques for cyberbullying detection on twitter,” *Futur. Internet*, vol. 12, no. 11, pp. 1–21, 2020, doi: 10.3390/fi12110187.
- [4] N. L. P. M. S. Putri Waisnawa, D. Nurjanah, and H. Nurrahmi, “Cyberbullying Detection on Twitter using Support Vector Machine Classification Method,” *Build. Informatics, Technol. Sci.*, vol. 3, no. 4, pp. 661–666, 2022, doi: 10.47065/bits.v3i4.1435.
- [5] A. Akhter, K. A. Uzzal, and M. M. A. Polash, “Cyber Bullying Detection and Classification using Multinomial Naïve Bayes and Fuzzy Logic,” *Int. J. Math. Sci. Comput.*, vol. 5, no. 4, pp. 1–12, 2019, doi: 10.5815/ijmsc.2019.04.01.



- [6] F. Ihsan, I. Iskandar, N. S. Harahap, and S. Agustian, "Decision tree algorithm for multi-label hate speech and abusive language detection in Indonesian Twitter," *J. Teknol. dan Sist. Komput.*, vol. 9, no. 4, pp. 199–204, 2021, doi: 10.14710/jtsiskom.2021.13907.
- [7] N. A. Razmi, M. Z. Zamri, S. S. S. Ghazalli, and N. Seman, "Visualizing stemming techniques on online news articles text analytics," *Bull. Electr. Eng. Informatics*, vol. 10, no. 1, pp. 365–365, 2021, doi: 10.11591/eei.v10i1.2504.
- [8] D. Farid and N. El-Tazi, "Detection of Cyberbullying in Tweets in Egyptian Dialects," vol. 18, no. 7, pp. 34–41, 2020, [Online]. Available: <https://sites.google.com/site/ijcsis/>
- [9] A. Karami, M. Lundy, F. Webb, and Y. K. Dwivedi, "Twitter and Research: A Systematic Literature Review through Text Mining," *IEEE Access*, vol. 8, pp. 67698–67717, 2020, doi: 10.1109/ACCESS.2020.2983656.
- [10] B. Haidar, M. Chamoun, and A. Serhrouchni, "A multilingual system for cyberbullying detection: Arabic content detection using machine learning," *Adv. Sci. Technol. Eng. Syst.*, vol. 2, no. 6, pp. 275–284, 2017, doi: 10.25046/aj020634.
- [11] R. Shah, S. Aparajit, R. Chopdekar, and R. Patil, "Machine Learning based Approach for Detection of Cyberbullying Tweets," *Int. J. Comput. Appl.*, vol. 175, no. 37, pp. 51–56, 2020, doi: 10.5120/ijca2020920946.
- [12] S. Gupta, "Sentiment Analysis: Concept, Analysis and Applications | by Shashank Gupta | Towards Data Science," 2018. <https://towardsdatascience.com/sentiment-analysis-concept-analysis-and-applications-6c94d6f58c17> (accessed Jun. 11, 2022).
- [13] F. Y. Pamuji and V. P. Ramadhan, "Komparasi Algoritma Random Forest dan Decision Tree untuk Memprediksi Keberhasilan Immunotherapy," *J. Teknol. dan Manaj. Inform.*, vol. 7, no. 1, pp. 46–50, 2021, doi: 10.26905/jtmi.v7i1.5982.
- [14] N. Novalita, A. Herdiani, I. Lukmana, and D. Puspandari, "Cyberbullying identification on twitter using random forest classifier," *J. Phys. Conf. Ser.*, vol. 1192, no. 1, 2019, doi: 10.1088/1742-6596/1192/1/012029.
- [15] A. Primajaya and B. N. Sari, "Random Forest Algorithm for Prediction of Precipitation," *Indones. J. Artif. Intell. Data Min.*, vol. 1, no. 1, p. 27, 2018, doi: 10.24014/ijaidm.v1i1.4903.
- [16] M. A. Al-Garadi *et al.*, "Predicting Cyberbullying on Social Media in the Big Data Era Using Machine Learning Algorithms: Review of Literature and Open Challenges," *IEEE Access*, vol. 7, pp. 70701–70718, 2019, doi: 10.1109/ACCESS.2019.2918354.