Tuning Model Analisis Sentimen Tweeter Sepakbola pada Dataset Kecil dan Seimbang

Wijanarto[1], Adelia Puspita Sari[2], Asih Rohmani[3]
Fakultas Ilmu Komputer Universitas Dian Nuwantoro
e-mail: [1]wijanarto@dsn.dinus.ac.id , [2]adell.puspita@gmail.com, [2]aseharsoyo@dsn.dinus.ac.id

Diterima: 10 Januari 2020; Direvisi: 3 Mei 2020; Disetujui: 6 Mei 2020

Abstrak

Suporter bola adalah orang yang mendukung dan memberikan motivasi serta semangat untuk pemain klub bola yang memiliki fanatisme positif maupun negatif, baik dalam dunia nyata atau social media, tweeter. Penelitian ini menghasilkan model klasifikasi untuk prediksi tweet supporter sepakbola dengan sedikit data dan berimbang. Model klasifikasi dibangun berdasarkan ekplorasi analisis data dan penentuan baseline model dari akurasi null, polarisasi dan subyektivitas, seleksi fitur, klasifikasi linier dan non linier. Model terpilih akan dilakukan tuning untuk mendapatkan hasil yang lebih presisi dan akurat serta dievaluasi dengan confusion matrik serta laporan klasifikasi untuk memberikan intuisi lebih dalam tentang perilaku classifier atas akurasi global. Hasil penelitian ditemukannya polarisasi kata bermakna negative yang berada dikelas positif sebesar 88% dengan frekuensi 4% dan rerata harmoni 8%. Model klasisfikasi Multinomial Naïve Bayes terpilih sebagai model terbaik dengan akurasi 99%, error 0.8% pada data train dan 100%, error 0% pada data validasi. Eksperimen untuk menguji model terhadap 30 entri data test baru, menghasilkan prediksi denganakurasinya 87% dengan error 13%, artinya hanya terdapat 4 kesalahan prediksi. Kedepan disarankan untuk menguji model ektraksi fitur atau melakukan boosting, bagging dan deep learning untuk mengetahui apakah hasilnya menjadi lebih baik.

Kata kunci: *analisis Sentimen, multinomial naïve bayes, twitter, suporter, fanastisme*

Abstract

Football supporters are people who support and provide motivation and enthusiasm for club players who have positive and negative fanaticism, whether in the real world or social media, tweeters. This study produced a classification model for predictions of supporting football tweets with little data and balance. The classification model is built on the exploration of data analysis and determination of the baseline model of null accuracy, polarization and subjectivity, feature selection, linear and non-linear classification. The selected model will be adjusted to obtain more precise and accurate results and evaluated with the confusion of matrices and classification reports to provide deeper intuition regarding the classification of classifications of global accuracy. The results of the study found the polarization of negative words in the positive class of 88% with a frequency of 4% and an average harmony of 8%. The Naïve Bayes Multinomial Classification Model was chosen as the best model with 99% approval, 0.8% error in the data train and 100%, 0% error in data validation. Experiments to study the model of 30 new test data entries, resulted in predictions with 87% accuracy with 13% errors, meaning that only contained 4 prediction errors. It is better to study the feature extraction model or to improve, pocket and study deeply to find out if it is better.

Keywords: *sentiment analysis, multinomial naïve bayes, twitter, supporters, fanaticism*

1. PENDAHULUAN

Sepak bola sangat popular dan digemari oleh masyarakat di seluruh dunia. Keberhasilan dari klub-klub sepakbola tidak terlepas dari yang namanya peranan suporter, seperti di eropa (UEFA) dengan 1 juta total penonton dari 11 klub terbesar [1], sedangkan di Indonesia Liga 1 dengan 4 klub terbesarnya, memiliki penonton lebih dari 20 ribu per laga [2]. Suporter bola merupakan orang-orang yang mendukung dan memberikan motivasi serta semangat untuk para pemain klub. Jenis-jenis suporter dibedakan menjadi 5, dari lima jenis suporter tersebut beberapa memiliki sifat fanatisme. Fanatisme adalah keyakinan atau kepercayaan yang sangat kuat terhadap sesuatu ajaran baik itu politik, agama dan lainnya, yang dalam hal ini adalah terhadap klub sepak bola [3]. Fanatik sendiri memiliki dampak posistif dan negatif contohnya seperti suporter bola. Dampak positif dengan adanya suporter yaitu memberikan dukungan terhadap klub yang digemari dimana dukungan tersebut berpengaruh untuk para pemain, penjulan tiket yang tinggi seperti pada pertandingan Liga 1 dimana suporter Persija Jakarta dengan jumlah rata-rata penonton tiap laga sebanyak 42.360 penonton[2]. Sedangkan dampak negatif yang ada seperti banyaknya tindakan kekerasan, kerusuhan, merusakkan fasilitas yang disediakan oleh pemerintahan dan jatuhnya korban baik itu luka-luka hingga tewas seperti yang terjadi pada Haringga Sirla suporter Jakmania yang tewas karena suporter dari Bobotoh[4]. Dalam tindakan kekerasan yang dilakukan suporter bola tidak berlangsung secara fisik saja. Dalam dunia maya juga terjadi kerusuhan yang dilakukan di sosial media, seperti bullying, mencela group lawan ataupun mencurahkan opini yang ada dalam pikiran. Twitter merupakan microblogging web-site yang dapat digunakan seseorang dalam mengemukakan pendapat yang tak jarang ada beberapa tweet yang berisi ujaran kebencian.

Ujaran kebencian menjadi menarik karena dapat menggiring opini public pada tingkat yang mengkhawatirkan, baik pada bidang politik, seperti pada penelitian Mahardika [5] yang berisi tentang tweet pemerintahan. Sedangkan dalam penelitian Putra,dkk [6] melihat seberapa besar ujaran kebencian pada media facebook dan dalam penelitian Lestari,dkk [7] melakukan analisis sentimen tentang opini pilkada DKI, yang ketiga tiganya menggunakan teknik *Naïve Bayes Classifier* dalam menganalisis sentiment terhadap ujaran media sosial. Banyak metode mesin learning dipakai untuk menganalisis sentiment, mulai dari *Null Model Accuracy*, polarisasi dan subyektivitas dari konsep *Bag of Word, Parts of speak* dan NLP, perhitungan dan frekuensi kata serta klasifikasi. Analisa awal dengan menetapkan model dasar *null accuracy* dipakai untuk baseline pemodelan, dengan membagi data dan kelasnya. Konsep *Bag of word* dalam analisa sentiment juga telah dilakukan oleh Madhu [8], suatu pendekatan untuk menolong seseorang dengan menganalisis kecenderungan dalam pesan mereka. Selain itu teknik pemodelan dengan TF-IDF seperti dalam penelitian Li [9] dan Das [10] yang membandingkan kinerja model *binary of words*, model TF-IDF dan TF-IDFdengan model *'next negation'* (TF-IDF-NWN) untuk klasifikasi teks, dan thesis untuk menganalisis tweet Twitter membedakan pendapat pengguna tentang *Genetically Modified Organisms* (GMOs). Selain itu juga pernah dilakukan pendeteksian sentimen secara otomatis diungkapkan oleh anak-anak menggunakan mesin pembelajaran yang dilakukan oleh de Vries [11]. Terakhir dari kelompok ini adalah studi dalam penelitian Ghag,dkk [12], dimana mereka menyajikan pendekatan untuk mengklasifikasikan istilah sebagai positif atau negatif berdasarkan distribusi jumlah frekuensi proporsional dan distribusi jumlah kehadiran proporsional secara positif menandai dokumen dibandingkan dengan tag negative dokumen.

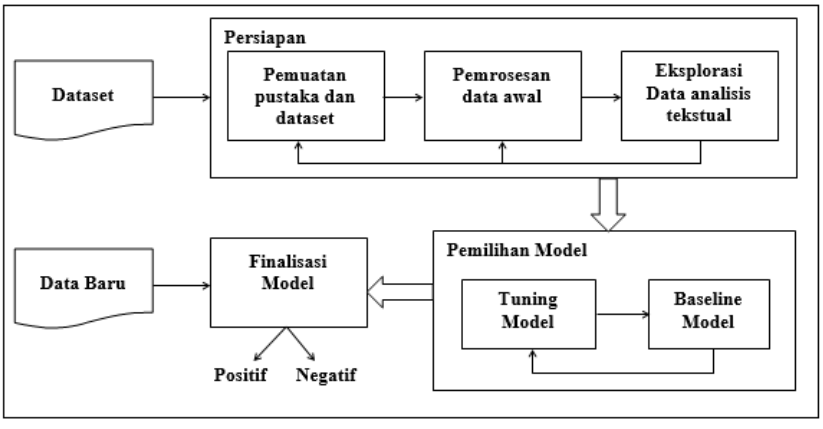
Sementara dalam kelompok model klasifikasi linier, sangat banyak penelitian yang sudah dilakukan, seperti yang dilakukan Umniy Salamah [13] yang mencoba menangani komplain dari suatu organisasi, Korovkinas,dkk [14] dan Tyagi,dkk [15] dalam penelitiannya yang membandingkan metode logistic, SVM dan naïve bayes untuk menganalisa sentiment. Kemudian Jurafsky [16] mengimplementasikan teknik bayes dan regresi logistic untuk fokus pada satu tugas kategorisasi teks yang umum. Untuk implementasi *Support Vector Machine*, Muscolino [17] mencoba menganalisis kelebihan fitur linguistik dalam ulasan media sosial untuk mendeteksi sentimen pesan apa pun. Sedangkan Bhumika,dkk [18] melakukan analisis

sentimen untuk mengklasifikasikan ulasan ini berdasarkan pendapatnya positif atau positif kategori negative pada ulasan pengguna produk apapun, sementara Gaurangi Patil,dkk [19] mempelajari tentang pengklasifikasi untuk sentimen analisis opini pengguna terhadap kandidat politik melalui komentar dan tweet.

Dalam klasifikasi non-linear yang dilakukan Aljuhani,dkk [20] membandingkan metode TF-IDF, Glove dan word2vec dengan multinomial naïve bayes untuk review pembeli produk mobile phone di Amazon, juga dalam penelitian Abbas,dkk [21] dan Farisi,dkk [22] menggunakan naïve bayes dalam analisa sentiment. Sementara Ismail,dkk [23] melakukan analisis komparatif pembelajaran mesin klasifikasi multinomial dan bernoulli untuk analisis sentimen Twitter, dan penggunaan SVM dan KNN dalam penelitian Ahmad,dkk [24] mencoba menganalisis ini secara langsung berada di bawah domain "analisis sentimen". Analisis sentiment meliputi bidang luas klasifikasi pengguna yang efektifteks yang dihasilkan di bawah polaritas yang ditentukan. Sementara Rezwanul,dkk [25] membuat system klasifikasi sentiment pada tweet seseorang dengan KNN dan SVM. Jotheeswaran,dkk [26] melakukan analisa opini mining menggunakan decision tree based seleksi fitur melalui manhattan tindakan klaster hierarkis dan Bharati,dkk [27] melakukan klasifikasi Sentimen menggunakan pohon keputusan dengan pemilihan fitur untuk opinion mining dan social web. Komparasi SVM, Decision Tree dan Random Forest juga dilakukan dalam penelitian Guia,dkk [28] untuk menganalisa informasi yang dapat berasal dari banyak orang dan tempat yang berbeda seperti situs ulasan online dan jejaring social. Beberapa studi diatas dapat dijadikan referensi dalam tulisan ini untuk menentukan model optimal dengan melakukan tuning pada data kecil dan seimbang ulasan tweeter sepakbola.

2. METODE PENELITIAN

Kerangka kerja metode diilustrasikan pada gambar 1, yang terdiri dari 3 (tiga) bagian utama, dimana didalamnya terdapat sub bagian yang merupakan detail dari bagian diatasnya. Bagian utama yang pertama adalah *Persiapan*, yang terdiri dari pemuatan library dan dataset, pemrosesan data, eksplorasi data analisis, Bagian kedua adalah pemilihan model , yang terdiri dari penentuan model dasar (baseline) dan tuning model , terakhir finalisasi model, yang terdiri dari penyimpanan model akhir dan pemanggilan model untuk prediksi data baru, berikut skema blok dari kerangka metode penelitian.



Gambar 1. Kerangka metode penelitian

2.1. Dataset

Data merupakan bahan utama penelitian, dalam hal ini data yang sepakbola dari 3 klub besar di liga 1 Indonesa, Persib Bandung, Persija Jakarta dan Arema Malang, diambil dengan aplikasi TweetScraper [29] dengan hashtag #jakmania, #bobotoh, dan #aremania. Data berhasil

diambil sebanyak 300 tweet setelah melakukan pemilihan secara manual dengan kriteria pertama, tweet yang digunakan menggunakan bahasa Indonesia baku, kedua, bukan merupakan komentar atau spam dan terakhir tweet yang belum dipilih sebelumnya (duplikat). Class atau target data dibagi 2 (biner) yang diberi label secara manual, yaitu positif dan negative.

2.2. Persiapan

Dalam tahap persiapan ini, kita akan mengolah data sesuai dengan tujuan kita yaitu membuat model klasifikasi yang optimal untuk data kecil dan seimbang, yang terdiri dari 3 langkah seperti berikut.

2.3. Penggunaan Pustaka

Dalam melakukan penelitian dan eksperimen akan menggunakan pustaka dari python panda, numpy, scipy, scikit-learn, matplotlib, seaborn, nltk, keras, textblob, wordcloud, beatifulsoup dan sebaginya.

2.4. Pemrosesan awal

Tahap ini terdiri dari 5 sekuen langkah pembersihan terhadap kata atau text sebagai berikut : Dekode HTML, '@' mention, Tautan URL, UTF-8 BOM (*Byte Order Mark*), tagar/angka, dengan menggunakan library Beautifullsoup dan regex dari python. Kita juga akan melakukan pemisahan data menjadi dataset train, yaitu sampel data yang digunakan untuk belajar, dataset validasi (*Hold-out cross-validation set*), yaitu sampel data yang digunakan untuk menyempurnakan parameter classifier, dan memberikan evaluasi model yang tidak bias dan terakhir dataset uji/data baru set yaitu sampel data yang digunakan hanya untuk menilai kinerja model akhir.Rasio yang diputuskan adalah dengan membagi data menjadi 80/10/10, 80% data sebagai train set, dan 10% untuk set validasi, dan 10% akhir untuk set tes. Alasan di balik rasio ini berasal dari ukuran seluruh kumpulan data yang kecil. Dataset hanya memiliki 300 entri dengan class positif 50% dan negatif 50%. Dalam hal ini, hanya 10% dari seluruh data hanyalah 30 entri. Ini dianggap cukup untuk mengevaluasi model dan memperbaiki parameter dan walaupun data seimbang belum tentu menghasilkan prediksi dengan akurasi tinggi.

2.5. Eksplorasi data analisis

Sebelum kita melakukan langkah untuk pemodelan, sebaiknya kita harus menganalisis dari beberapa sudut pandang, dalam hal ini kita akan melakukan eksplorasi data visualisasi teks pertama yang kita pilih adalah *word cloud* yang kontroversial. Selanjutnya kita akan menghitung kata serta melihat polarisasinya, dengan menghitung vector, token, frekuensi serta mengimplementasikan zipf law [30], [31], yang dapat dituliskan sebagai berikut: kata yang paling sering memiliki frekuensi f(r) yang berskala sebagai berikut :

$$f(r) \propto \frac{1}{r^{\alpha}} \text{untuk} \alpha \approx 1 \quad (1)$$

Persamaan (2) merupakan perhitunga rerata nilai positif, yaitu frekuensi kemunculan kata positif pada kelas tertentu dibandingkan jumlah frekuensi positif dan negative.

$$pos_{rate} = \frac{frek_{pos}}{frek_{pos} + frek_{neg}} \quad (2)$$

Persamaan (3) mendefinisikan perhitungan metrik frekuensi kata-kata yang muncul dalam kelas.

$$pctfrek_{pos} = \frac{frek_{pos}}{\sum frek_{pos}} \quad (3)$$

Sedangkan metrik yang dipakai untuk menganalisa rerata harmonic yang didefinisikan sebagai bilangan real positif x_1, x_2, ... x_n, dapat dihitung dengan persamaan (4).

$$H = \frac{n}{\sum_{i=1}^{n} \frac{1}{x_i}} \quad (4)$$

Rerata harmonik dari *list* bilangan cenderung kuat terhadap elemen-elemen paling sedikit dari list, cenderung (dibandingkan dengan rata-rata aritmatika) untuk mengurangi dampak *outlier* besar dan memperparah dampak yang kecil.

2.6. Pemilihan Model

Tahap ini merupakan inti dari penelitian, yaitu melakukan pemilihan model yang terdiri dari penentuan baseline model untuk dilakukan tuning model yang terpilih. Baseline model ditentukan dengan tiga tahap, yaitu pertama, menentukan baseline awal yang didasarkan null akurasi saat pembagian dataset, lalu juga dibandingkan dengan melihat polarisasi dan subyektifitas kata dengan menggunakan textblob dari python. Kedua, fitur ekstraksi, ini analisis berbasis angka, karena kita akan melihat tweet dengan merubah kalimat kedalam vector angka baik untuk dilakukan perhitungan (*CountVectorizer*) dan analisis frequensi kata (TfidfVectorizer). Ketiga, klasifikasi yaitu penentuan baseline dengan model klasifikasi sesuai tujuan penelitian ini yaitu mencari model klasifikasi untuk menentukan tweet berada dalam 2 kelas yang berbeda. Ada 6 model yang dijadikan dasar yaitu model linier, Logistic Regression, Linear Support Vector Machine, serta model non-linear yaitu *Multinomial* dan *Bernoulli Naïve Bayes, DecisionTree, K-Nearest Neighbors, SupporVector Machine*.

2.7. Tuning Model

Tahap ini akan melakukan parameterisasi atau hyperparameter terhadap model klasifikasi terpilih dari baseline dan mencari hasil evaluasi terbaik untuk dataset yang telah dibersihkan.

2.8. Finalisasi Model

Setelah model terpilih langkah terakhir adalah menyimpan model untuk dapat digunakan sebagai model prediksi pada data baru atau data eksperimen

2.9. Eksperimen data baru

Tahap ini adalah melakukan experiman dengan data tweet baru terhapad model yang terpilih untuk mernetukan tweet tersebut dalam class positif atau negative. Data dapat dalam bentuk tunggal atau banyak yang dimuat dalam suatu dataframe.

3. HASIL DAN PEMBAHANSAN

3.1. Pemrosesan awal

Setelah dataset di scrap dengan *TweeterScrapper*, dipilih sesuai kebutuhan penelitian, dilabeli dan disimpan dalam file. Dataset terdiri dari 3 atribut yaitu id sebanyak 300 baris bertipe int64, text 300 baris bertipe object string dan sentiment sebanyak 300 baris bertipe int64 dan semua tanpa entri kosong. Selanjutnya data siap untuk dilakukan pemrosesan, seperti pada tabel 1 yaitu sampel 5 tweet teratas dari file dataset yang telah terpilih sebagai berikut.

Tabel 1. Sampel 5 data teratas dataset tweeter sepakbola

id	text	sentiment
1	Ngalah sama tuan rumah	2
2	Harus gila !!! #aremafc #aremania #aremanita #Wearemania	1
3	Siapa lagi yang akan datang ? #aremafc #aremania #aremanita #wearemania	2

4	Iya mungkin, saya kurang paham moment saat itu. karena itu agak malam terus sayup terdengar yel-yel " aremania bajinguk". Entah apakah saat itu di kereta yang sama ada supporter arema, saya juga kurang tau.	2
5	aremania ya lo? nulis dibalik2	2

Untuk keseluruhan dataset berdasarkan pelabelan yang telah dilakukan maka terdapat dua kelas yaitu sentiment positif dan sentiment negatif pada atribut *sentiment*, sementara atribut *id* sengaja dihapus karena tidak berpengaruh pada analisa sentiment selanjutnya, seperti terlihat dalam table berikut dibawah ini.

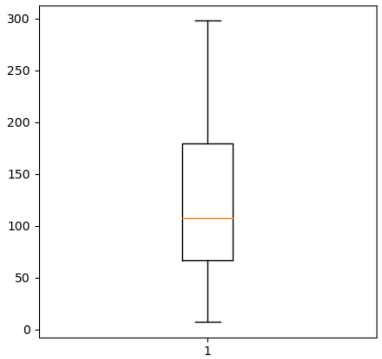
Tabel 2. Sampel atribut kelas sentiment

	Sentimen Positif			Sentimen negative	
	text	**sentiment**		**text**	**sentiment**
1	Harus gila !!! #aremafc #aremania #aremanita...	1	**0**	Ngalah sama tuan rumah	2
5	Mendukung lewat do'a	1	**2**	Siapa lagi yang akan datang ? #aremafc #arema...	2
8	Aremania supporter kelas dunia?? tidak harus ...	1	**3**	Iya mungkin, saya kurang paham moment saat itu...	2
11	Silahkan Saudara, Selamat Datang Di Bumi AREMA	1	**4**	aremania ya lo? nulis dibalik2	2
19	Mental garuda luar biasa	1	**6**	AREMA KERAS !!! #aremafc #aremania #aremanit...	2

Selanjutnya nilai atribut sentiment akan dirubah menjadi 0 untuk negative dan 1 untuk positif, karena lebih mudah diterima. Sementara distribusiatribut kelas sentiment setelah dilakukan analisa awal dengan menghitung panjang string pada kolom teks , karena panjang teks maksimum yang diperbolehkan hanya 140 karakter dan ini sangat berpengaruh saat pembentukan token dan disajikansampel 5 baris awal pada table 3 berikut ini,

Tabel 3. Menghitung panjang string atribut teks diatas 140 karakter

	text	**sentiment**	**pre_clean_len**
0	Ngalah sama tuan rumah	0	22
1	Harus gila !!! #aremafc #aremania #aremanita...	1	59
2	Siapa lagi yang akan datang ? #aremafc #arema...	0	74
3	Iya mungkin, saya kurang paham moment saat itu...	0	207
4	aremania ya lo? nulis dibalik2	0	31
5	Mendukung lewat do'a	1	20
6	AREMA KERAS !!! #aremafc #aremania #aremanit...	0	60
7	Iya min. Di luar dari sidoarjo sendiri, letak ...	0	270



Gambar 2. Grafik kotak sebaran jumlah karakter atribut teks.

Tampak terdapat 8 baris pertama dengan panjang lebih dari 140 karakter, dengan ploting kotak diharapkan dapat melihat lebih jelas dari keseluruhan dataset seperti gambar 2.

Dalam pembersihan data, masalah pertama yang kita sadari adalah bahwa, selama proses pembersihan, kata-kata negasi yang disingkat harus dipanjangkan, seperti "tdk" menjadi "tidak", dan sebagainya. Ini sepertinya bukan masalah sepele untuk tujuan analisis sentimen. Kedua yang kita sadari adalah, beberapa tautan url tidak dimulai dengan "http", kadang-kadang orang menempelkan tautan di formulir "www.cobatweet.com". Ini tidak ditangani dengan benar ketika kita menetapkan pola regex alamat url sebagai 'https?: // [A-Za-z0-9./] +'. Dan masalah lain dari pola regex ini adalah bahwa ia hanya mendeteksi alfabet, angka, titik, garis miring. Ini berarti akan gagal menangkap bagian tersebut dari url, jika mengandung karakter khusus lain seperti "=", "_", "~", dll. Dengan demikian akan dilakukan pembersihan data dengan 5 sekuen langkah sebagai berikut :

a. Dekode HTML : decoding HTML ke teks umum akan menjadi langkah pertama dalam persiapan data. Dalam penelitian ini kita akan menggunakan library BeautifulSoup.
b. '@'mention : meskipun @mention membawa informasi tertentu (yang disebutkan oleh pengguna lain yang tweet itu), informasi ini tidak menambah nilai untuk membangun model analisis sentimen.
c. Tautan URL : sama dengan @mention, meskipun membawa beberapa informasi, untuk tujuan analisis sentimen, ini dapat diabaikan.
d. UTF-8 BOM (Byte Order Mark) :Dengan melihat entri di atas, kita dapat melihat pola karakter aneh "\xef \xbf \xbd". Dengan mendekode teks dengan 'utf-8-sig', BOM ini akan diganti dengan unicode karakter khusus yang tidak dapat dikenali, maka kita dapat memproses ini sebagai "?"
e. Tagar/angka :kita memutuskan untuk membiarkan teksnya tetap utuh dan menghapus '#'. Kita akan melakukan ini dalam proses pembersihan semua karakter non-huruf termasuk angka.Kelima langkah tadi dapat dilihat hasilnya dalam perbandingan tabel 4 berikut,
f.

Tabel 4. Perbandingan dataset sebelum dan sesudah dibersihkan

Sebelum dibersihkan	Sesudah dibersihkan
Ngalah sama tuan rumah	ngalah sama tuan rumah
Harus gila !!! #aremafc #aremania #aremanita...	harus gila aremafc aremania aremanita wearemania
Siapa lagi yang akan datang ? #aremafc #arema...	siapa lagi yang akan datang aremafc aremania aremanita wearemania
Iya mungkin saya kurang paham moment saat itu...	iya mungkin saya kurang paham moment saat itu karena itu agak malam terus sayup terdengar yel yel aremania bajinguk entah apakah saat itu di kereta yang sama ada supporter arema saya juga kurang tau
aremania ya lo? nulis dibalik2	aremania ya lo nulis dibalik

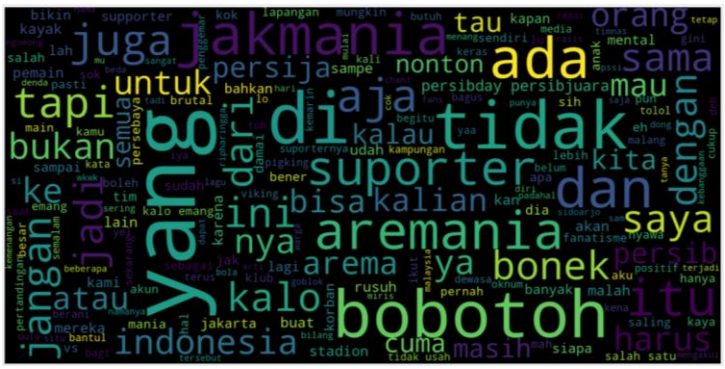
Langkah ini diikuti dengan menyimpan hasil dataset yang sudah bersih kedalam file baru, untuk digunakan dalam analisa data dan pemodelan.

3.2. Eksplorasi data analisis

A. *Word Cloud*

Word Cloud merepresentasikan penggunaan kata dalam dokumen dengan mengubah ukuran kata secara proporsional sesuai frekuensinya, dan kemudian mempresentasikannya dalam pengaturan acak. *World Cloud* hanya mendukung jenis analisis tekstual yang paling kasar, dan sering diterapkan pada situasi di mana analisis tekstual tidak sesuai, dan membuat pembaca untuk mencari tahu konteks data sendiri tanpa menyediakan narasi.Tetapi dalam kasus tweet, analisis tekstual adalah analisis yang paling penting, dan ini memberikan gambaran umum tentang kata-kata seperti apa yang sering muncul di corpus. Jadi, kita akan mencobanya, dan mencari tahu metode apa yang dapat digunakan untuk visualisasi teks. Berikut hasil

visualisasi dengan library worldcloud python, tweet negative yang didapatkan seperti pada gambar 3 dibawah ini.



Gambar 3. Analisis tweet negative

Beberapa kata besar dapat diartikan cukup netral, seperti "suporter", "pemain", dll. kita dapat melihat beberapa kata dalam ukuran yang lebih kecil masuk akal untuk berada di tweet negatif, seperti "kampungan", "dasar", "pigking", "turut berduka", dll. Tapi ada "cinta" dalam ukuran yang tidak besar, jadi kita ingin melihat apa yang terjadi, meskipun tweet itu mengandung kata "suka" yang muncul sebanyak 2 baris pada tweet negative, “cinta itu sederhana persib untuk bobotoh dan bobotoh untuk persib persibday persibjuara”, dalam kasus ini adalah sentimen negatif, karena tweet itu memiliki emosi yang campur aduk seperti "suka tapi benci", atau kadang-kadang digunakan dengan cara sarkastik, sekarang kita melihat tweet positif seperti disajikian gambar 4.

Sekali lagi kita melihat beberapa kata netral dalam ukuran besar, "suporter", "banyak", tetapi kata-kata seperti "turut berduka", "cok", juga muncul. Menariknya, kata "suka" muncul di *word cloud* negatif, tetapi juga ada di *word cloud* positif. Ini mungkin menyiratkan bahwa banyak orang mengekspresikan sentimen negatif terhadap kesedihan, tetapi juga banyak orang yang positif tentang kebahagiaan dan cinta ditemukan juga dalam tweet positif yaitu, “yappp sampai kapan kita mengikuti keegoan kita mulai sekarang kita harus jadi supporter yang di teladani oleh supporter yang lain nya viking bobotoh cinta damai demi persib bandung tercinta”, “terima kasih bang salam hangat dari saya bobotoh asal sukabumi semoga tak ada lagi saling mengejek kita cinta kedamaian”, “untuk indonesia lebih baik peran orang tua sangat besar untuk bisa lebih menyortir hal hal yang masuk dalam otak anak mau bikin anak jadi bobotoh support sepak bola lokal buat jadi cinta sepak bola boleh tapi tolong tanpa membenci siapapun tidak rugi kok”, “tidak mencintai bukan harus membenci bagitupun kalo cinta persib bukan harus benci persija bangga jadi bobotoh bukan berarti benci the jak kita tinggal dibumi yang sama negara yang satu dan kalian sendiri yang membuat pecah ripharingga jakmaniaberduka”

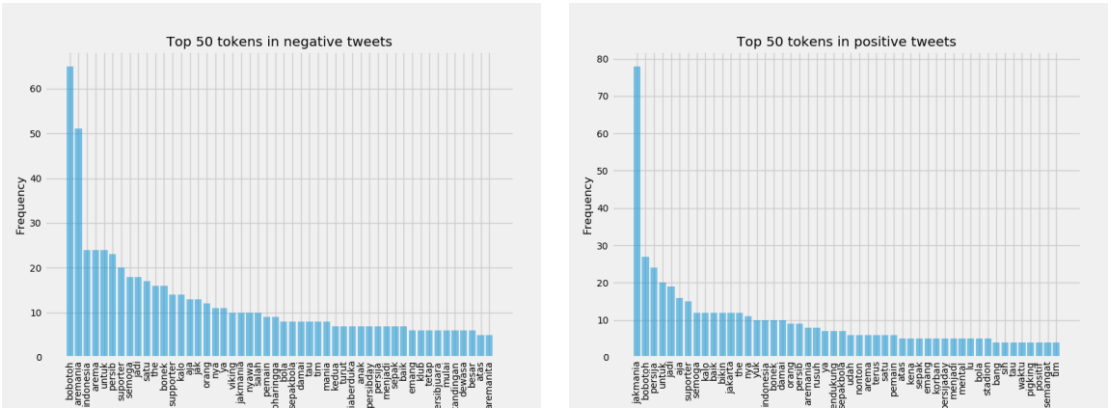
Gambar 4. Analisis tweet positif

Selanjutnya agar kita dapat mengimplementasikan beberapa visualisasi data pada langkah berikutnya, kita perlu data frekuensi. Kata-kata apa yang digunakan dalam ulasan, dan berapa kali digunakan di seluruh korpus. Kita menggunakan penghitung vektor untuk menghitung frekuensi istilah, meskipun penghitung vektor juga cocok, melatih, dan memprediksi, tetapi pada tahap ini, kita hanya akan mengekstraksi frekuensi term untuk visualisasi.Ada beberapa opsi parameter yang tersedia untuk penghitung vektor, seperti menghapus stopword, membatasi jumlah istilah maksimum. Namun, untuk mendapatkan gambaran lengkap dari dataset terlebih dahulu, kita menerapkannya dengan memasukkan stopwords, dan tidak membatasi jumlah term maksimum dan kita sudah membuang 1747 kata dari corpus dengan library CountVectorizer, lalu merubahnya menjadi matrik tersebar dan berikut hasil sebaran term negative , positif serta total keseluruhannya dalam table 4 dibawah ini.

Tabel 4. Frekuensi term negative, positif dan total

	negative	positive	total
bobotoh	65	27	92
jakmania	10	78	88
aremania	51	8	59
untuk	24	20	44
jadi	18	19	37
suporter	20	15	35
indonesia	24	10	34
persib	23	9	32
persija	7	24	31
semoga	18	12	30

Kerangka data frekuensi telah dibuat dan seperti yang kita lihat, kata-kata yang paling sering adalah kata-kata berhenti seperti “bobotoh”, “jakmania,” “aremania”. Untuk mendapatkan frekuensi term kita ingin menguji denganhukum Zipf dengan hasil yang kita dapatkan dari atas. Indeks adalah token dari dataset tweets, dan angka-angka di kolom "negatif" dan "positif" menunjukkan berapa kali token itu muncul di tweet negatif dan tweet positif, berikut distribusinya seperti pada gambar 6.



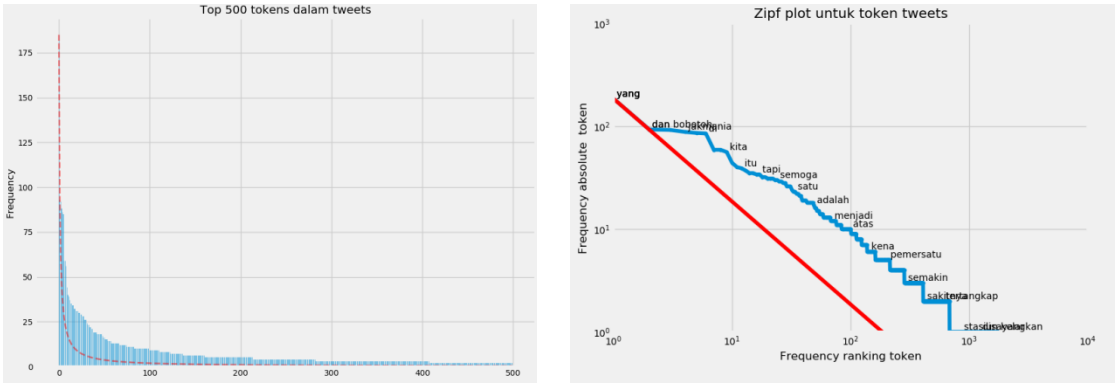
a. Distribusi 50 Token Negatif b. Distribusi 50 Token Positif

Gambar 6. Distribusi 50 token pertama

B. Hukum Zipf

Zipf Law menyatakan bahwa sejumlah kecil kata digunakan sepanjang waktu, sementara sebagian besar digunakan sangat jarang. Tidak ada yang mengejutkan bahwa kita sering menggunakan beberapa kata, seperti “dari”, ”pada”, dll, dan kita jarang menggunakan

kata-kata seperti ”aardvark” (aardvark adalah spesies hewan asli Afrika.). Namun, yang menarik adalah frekuensi setiap kata berbanding terbalik dengan peringkatnya dalam tabel frekuensi. Jadi kata yang paling sering muncul kira-kira dua kali lebih sering dari kata yang paling sering kedua, tiga kali lebih sering dari kata yang paling sering ketiga ” seperti pada persamaan (1) dan hasilnya disajikan dalam gambar 6.a, pada sumbu X adalah peringkat frekuensi tertinggi dari kiri ke peringkat 500 di sebelah kanan. Sumbu Y adalah frekuensi yang diamati dalam korpus (dalam hal ini, dataset tweet). Satu hal yang perlu diperhatikan adalah bahwa pengamatan ini mengikuti tren distribusi ”near-Zipfian”. Meskipun kita dapat melihat plotnya mengikuti tren Hukum Zipf, tetapi sepertinya plot tersebut memiliki lebih banyak area di atas kurva Zipf yang diharapkan dengan kata-kata yang berperingkat lebih tinggi. Cara lain untuk memplot ini adalah pada grafik log-log, dengan sumbu-X menjadi log (peringkat), sumbu-Y adalah log (frekuensi). Dengan memplot pada skala log-log, hasilnya akan menghasilkan garis linier kasar pada grafik.



a. Token tweet b. Zipf Plot Token tweet

Gambar 7. Token tweetdan . Zipf Plot Token tweet

Pada gambar 7b, sekali lagi kita melihat kurva linier kasar, dan tidak menyimpang di atas garis yang diharapkan (garis merah) pada kata-kata peringkat yang lebih tinggi, dan pada peringkat yang lebih rendah kita melihat garis pengamatan yang sebenarnya terletak di atas garis linear yang diharapkan. Setidaknya, kita membuktikan bahwa bahkan token tweet mengikuti distribusi "near-Zipfian", tetapi ini membuat kita penasaran tentang penyimpangan dari Hukum Zipf. Meskipun aturan itu sendiri menyatakan bahwa pengamatan yang sebenarnya mengikuti "near-Zipfian" daripada terikat dengan hukum, tetapi apakah area yang kita amati di atas garis yang diharapkan dalam peringkat yang lebih tinggi. Secara intuitif, jika sebuah kata muncul lebih sering di satu kelas dibandingkan dengan yang lain, ini bisa menjadi ukuran yang baik dari seberapa banyak kata tersebut bermakna untuk menggambarkan kelas tersebut. Di bawah ini kita menamakannya sebagai 'pos_rate', yang didefinisikan dalam persamaan (2) sebelumnya dan disajikan dalam table 5 berikut.

Tabel 5. Rerata token tweet positif

	negative	positive	total	pos_rate
maksud	0	1	1	1
kekondusifan	0	1	1	1
kebayakan	0	1	1	1
kebebasan	0	1	1	1
kecuali	0	1	1	1
kedatangannya	0	1	1	1
semarang	0	3	3	1
kehadiran	0	1	1	1
semalem	0	1	1	1
kek	0	1	1	1

Terlihat kata-kata dengan pos_rate tertinggi banyak memiliki frekuensi nol di tweet negatif, tetapi frekuensi keseluruhan dari kata-kata ini terlalu rendah untuk menganggapnya sebagai pedoman untuk tweet positif dan disajikan dalam prosentasi sebagai berikut dalam table 6.

Tabel 6 Prosentrase rerata positif token tweet

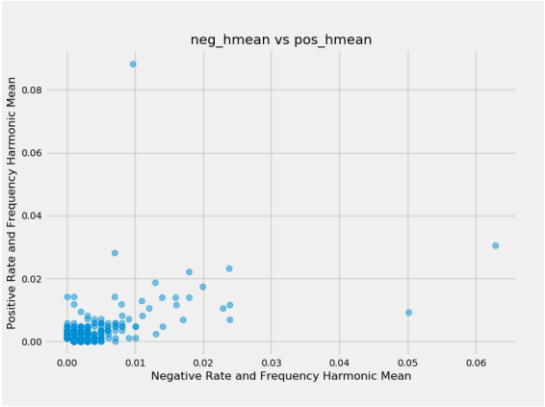
	negative	positive	total	pos_rate	pos_freq_pct
jakmania	10	78	88	0.886364	0.046456
bobotoh	65	27	92	0.293478	0.016081
persija	7	24	31	0.774194	0.014294
untuk	24	20	44	0.454545	0.011912
jadi	18	19	37	0.513514	0.011316
aja	13	16	29	0.551724	0.009529
suporter	20	15	35	0.428571	0.008934
semoga	18	12	30	0.4	0.007147
kalo	14	12	26	0.461538	0.007147
baik	7	12	19	0.631579	0.007147

Tetapi karena pos_freq_pct hanyalah frekuensi yang diskalakan dari jumlah total frekuensi, pangkat pos_freq_pct persis sama dengan frekuensi positif. Apa yang dapat kita lakukan sekarang adalah menggabungkan pos_rate, pos_freq_pct bersama-sama untuk menghasilkan metrik yang mencerminkan pos_rate dan pos_freq_pct. Meskipun kedua hal ini dapat mengambil nilai mulai dari 0 hingga 1, pos_rate memiliki rentang yang jauh lebih luas sebenarnya berkisar dari 0 hingga 1, sementara semua nilai pos_freq_pct terjepit dalam rentang yang lebih kecil dari 0,015. Jika kami meratakan dua angka ini, pos_rate akan terlalu dominan, dan tidak akan mencerminkan kedua metrik secara efektif.Jadi di sini kita menggunakan rata-rata harmonik bukan rata-rata aritmatika seperti pada persamaan (3) sebelumnya dan disajikan dalam table 7 sebagai berikut,

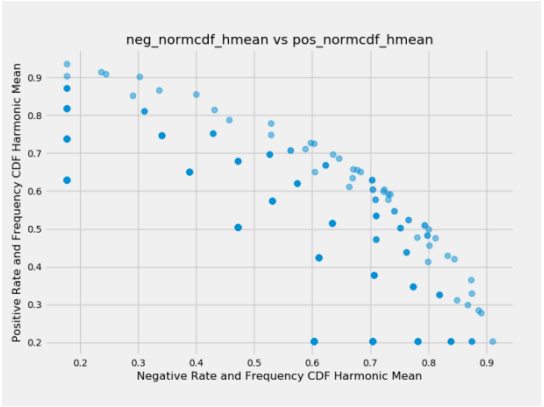
Tabel 7. Tabel rerata harmoni

text	negative	positive	total	pos_rate	pos_freq_pct	pos_hmean
jakmania	10	78	88	0.886364	0.046429	0.088235
bobotoh	65	27	92	0.293478	0.016071	0.030474
persija	7	24	31	0.774194	0.014286	0.028054
untuk	24	20	44	0.454545	0.011905	0.023202
jadi	18	19	37	0.513514	0.01131	0.022132
aja	13	16	29	0.551724	0.009524	0.018724
suporter	20	15	35	0.428571	0.008929	0.017493
jakarta	0	12	12	1	0.007143	0.014184
bikin	1	12	13	0.923077	0.007143	0.014176
baik	7	12	19	0.631579	0.007143	0.014126

Peringkat rata-rata harmonis sepertinya sama dengan pos_freq_pct. Dengan menghitung rata-rata harmonik, dampak nilai kecil (dalam hal ini, pos_freq_pct) terlalu diperparah dan akhirnya mendominasi nilai rata-rata. Sekali lagi ini sama persis dengan hanya peringkat nilai frekuensi dan tidak memberikan hasil yang berarti. Sehingga kita dapat melihat ploting grafik untuk melihat bagaimana nilainya dikonversi pada plot. Untuk membandingkan, pertama kita akan plot neg_hmean vs pos_hmean, lalu kita lihat Cumulative Distribution Function (CDF) normal dari Hmean yang disajikan pada gambar 8a dan b. Gambar 8.b menunjukan rata-rata harmonis dari tingkat CDF dan frekuensi CDF telah menciptakan pola yang menarik pada plot. Jika titik data dekat dengan sudut kiri atas, itu lebih positif, dan jika lebih dekat ke sudut kanan bawah, itu lebih negative, sementara 8.a tidak banyak perbedaan frekuensi negative dan positif.



a. Hmean Frekuensi



b. CDF Hmean Normal

Gambar 8.Hmean negative dan positif

C. Baseline model

Dalam menentukan baseline model kita akan membagi dataset terakhir yang sudah dibersihkan dan dianalisis dengan komposisi sebagai berikut, Train set: Sampel data training, Validasi set: Sampel data yang digunakan untuk menyempurnakan parameter classifier, dan evaluasi model yang tidak bias. Uji/data baru set: Sampel data yang digunakan hanya untuk menilai kinerja model akhir. Rasio yang kita putuskan untuk membagi data kita adalah 80:10:10, 80% data sebagai train set, dan 10% untuk set validasi, dan 10% akhir untuk set tes. Dataset hanya memiliki 300 entri.Artinya 10% dari seluruh data hanyalah 30 entri. Ini dianggap cukup untuk mengevaluasi model dan memperbaiki parameter. Hasilnya, Train set mempunyai total 240 entri dengan 57.92% tweet negative, 42.08% positive, validation set mempunyai total 30 entri dengan 40.00% negative, 60.00% positive, test set mempunyai total 30 entri dengan 50.00% negative, 50.00% positive.

a) Baseline 1 : Akurasi Null , Polaritas dan subyektifitas

Seperti yang kita lihat dari pembagian kelas set validasi di atas, kelas mayoritas negatif dengan 57.92%, yang berarti jika classifier memprediksi positif untuk setiap data validasi, itu akan mendapatkan akurasi 57.92%.Baseline lain yang ingin kita bandingkan dengan hasil validasi polaritas dan subyektifitas dengan *TextBlob*. Berikut hasilnya pada table 9, akurasi analisis sentimen 53.33% pada set validasi, yang 3% lebih akurat daripada akurasi nol (57.92%)

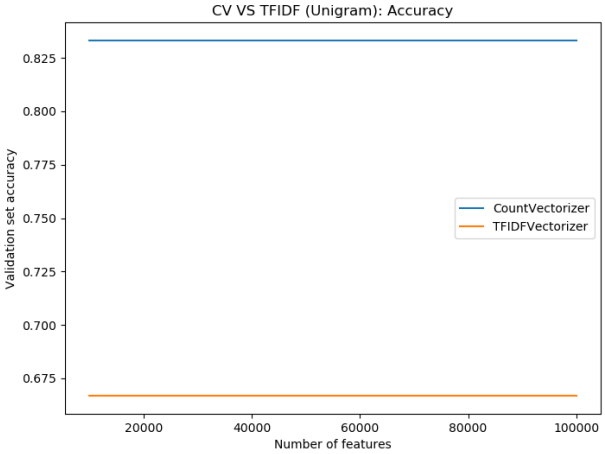
Tabel 9. Hasil evaluasi textblob

Accuracy	60.00%			
Confusion Matrix				
	predicted_positive	predicted_negative		
positive	18	0		
negative	12	0		
Classification Report				
	precision	precision	f1-score	support
0	0.00	0.00	0.00	12
1	0.60	1	0.75	18
accuracy			0.60	30
macro avg	0.30	0.5	0.37	30
weighted avg	0.36	0.6	0.45	30

b) Baseline 2 : Ekstraksi Fitur

Jika kita ingin menggunakan teks dalam algoritme Machine Learning, kita harus mengonversi text menjadi representasi numerik. Salah satu metode disebut bag-of-words (BOW) approach. Model BOW mengabaikan tata bahasa dan urutan kata. Setelah kita memiliki corpus (data teks), daftar kosa kata dibuat berdasarkan seluruh corpus. Kemudian setiap dokumen atau entri data direpresentasikan sebagai vektor numerik berdasarkan kosa kata yang dibangun dari corpus. Kita akan mengevaluasi dua model dari library python, CountVectorizer untuk menghitung vektor dan TFIDFVectorizer mengkonversi teks ke bentuk angka.

Kosakata tidak dibatasi walaupun kinerja model akan tergantung pada jumlah kosakata. Selain itu, kita juga menetapkan daftar stopwords khusus, yang berisi 10 kata paling sering dalam korpus: "ke", "pada", "ku", "nya", "dan", "kamu", "bukan", "adalah", "dalam", "untuk". Model yang kita pilih untuk mengevaluasi CountVectorizer berbeda adalah Multinomial Naïve bayes. Ini adalah salah satu model linier, sehingga dapat diukur secara komputasi untuk data kecil atau besar. Hasil validation seperti pada gambar 9, untuk 10.000 fitur dengan CountVectorizer dengan akurasi : 83.33%, sementara TFIDFVectorizer 66.67%, jelas terdapat selisi sebesar 16.66% lebih tinggi CountVectorizer dan 25.41% lebih tinggi dari akurasi null 57.92%.



Gambar 9. Perbandingan akurasi countvectorizer dan TFIDFVerctorizer

c) Baseline 3 : Klasifikasi

Untuk menentukan baseline 3 ini kita akan menggunakan 2 model linier (regresi logistic dan SVM Linier) dan non-linier (Multinomial dan Bernoulli Naïve Bayes, Decision Tree, K-Nearest Neighbor serta SVM). Evaluasi terhadap model dilakukan dengan pipeline , skor akurasi , penghapusan stopword dan membatasi jumlah fitur mulai 10000 hingga 100000 untuk perhitungan vektornya (CountVectorizer) dengan range ngram dari 1 sampai dengan 3, berikut hasilnya disajikan dalam table 9 dibawah ini.

Tabel 9. Hasil baseline klasifikasi

	1-gram	2-gram	3-gram
LogisticRegression	66.67%	66.67%	56.67%
LinearSVC	66.67%	63.33%	63.33%
MultinomialNB	**83.33%**	80.00%	80.00%
BernoulliNB	66.67%	43.33%	40.00%
DecisionTreeClassifier	60.00%	63.33%	63.33%
KNeighborsClassifier	50.00%	50.00%	40.00%
SVC	56.67%	53.33%	40.00%

Tampak bahwa akurasi tertinggi diperoleh algoritma MultinomialNB pada vektorisasi dengan unigram dan kosakata sebanyak 10000 serta penghapusan stopword. Dengan demikian kita akan melakukan tuning terhadap model MultinomialNB ini lebih lanjut.

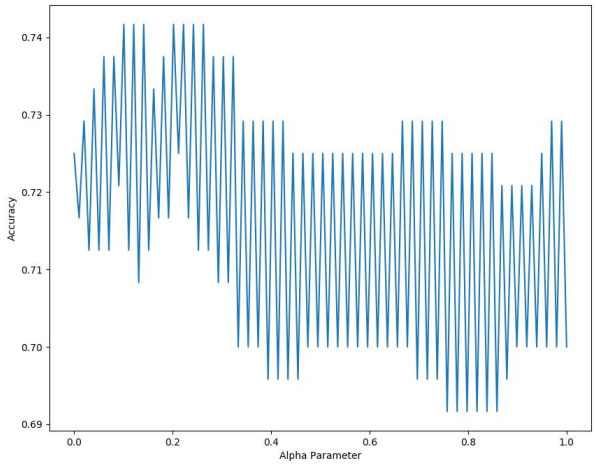
D. Tuning Model

Setelah terpilih baseline klasifikasi yaitu model Multinomial Naïve Bayes, kita akan lakukan tuning berdasarkan parameter seperti table 10 berikut :

Tabel 10. Parameter tuning model

Model	**Parameter**	
	CountVectorizer	**Cross validation**
Multinomian NB	- stop_words='indonesian' - max_features=10000 - analyzer='word' - ngram_range=(1, 2)	- k=10 - random seed= 3

Gambar 10 merupakan plot akurasi terbaik adalah 74.2% dengan alpha= 0.2040895918367347, 'dan fit_prior= True dari hasil tuning parameter terhadap model terpilih yaitu Multinomial naïve Bayes sebagai berikut



Gambar 10. Evaluasi akurasi tuning parameter model

Berikut gambar 10 hasil evaluasi akurasi terhadap model diatas semakin baik skornya setelah dilakukan tuning mulai dari baseline 1, 2 dan 3.

Tabel 11. Perbandingan akurasi baseline - tuning

Model	Akurasi Null	TextBlob	CountVectorizer	Tuning
MultinomialNB	57.92%	60%	83.3%	74%

Berdasarkan hasil tuning seperti pada table 11, maka dapat diputuskan untuk memilih model Multinomial Naïve Bayes sebagai model optimal untuk dataset tweeter sepak bola, walaupun terlihat penurunan dari CountVectorizer ke Tuning, tetapi sebenarnya akurasi pada kolom tersebut tidak dijadikan patokan untuk melihat evaluasi pada data validasi dan testing .

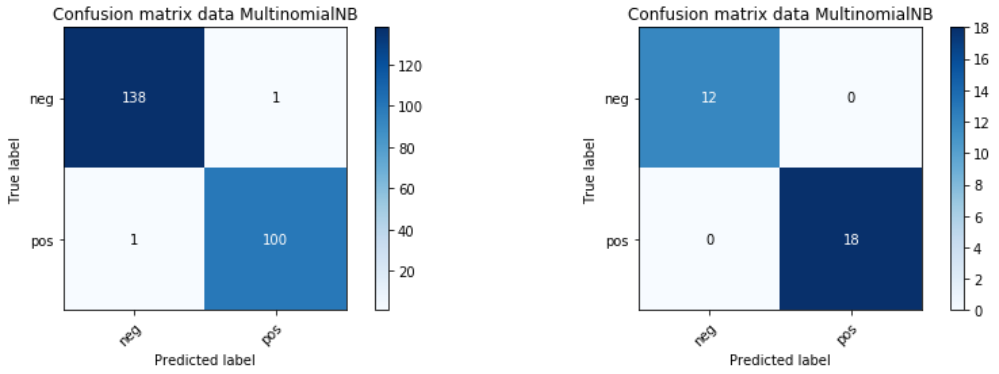
E. Finalisasi Model

Setelah mengetahui evaluasi dari model yang sudah dilakukan tuning maka kita akan simpan model dalam file yang dapat di panggil sewaktu-waktu untuk diberi data baru dan melakukan prediksi terhadapnya. Berikut hasil lengkap evaluasi data training dan testing pada table 12 terhadap model final yang sudah disimpan.

Tabel 11. Perbandingan akurasi traing dan testing

Dataset	Akurasi	Freq	Report				
				precision	recall	f1-score	support
Training	99.17%	0 : 139 1 : 101	0	0.99	0.99	0.99	139
			1	0.99	0.99	0.99	101
			accuracy			0.9	240
			macro avg	0.99	0.99	0.99	240
			weighted avg	0.99	0.99	0.99	240
				precision	recall	f1-score	support
Test	**100%**	0 : 12 1: 18	0	1	1	1	12
			1	1	1	1	18
			accuracy			1	30
			macro avg	1	1	1	30
			weighted avg	1	1	1	30

Akurasi data testing meningkat bagus dari 99% menjadi 100%, berdasarkan gambar 11 kita dapat mengukur error rate dari confusion matrik yang disajikan dalam gambar 11a dan 11 b berikut dibawah ini.



a. Data training b. Dataset testing

Gambar 11. Perbandingan confusion matrix data train dan test

Error rate dari data training sebesar 0.8%, sementara pada data testing 0%, sehingga terjadi penurunan yang cukup signifikan, namun mengingat total datasetnya kecil maka hal ini dapat dipandang sebagai hasil yang optimal, artinya kesalahan tersebut cukup kecil dibandingkan dengan 300 entri dataset yang ada. Berikut hasil percobaan sejumlah 30 test data pada model terpilih disajikan pada gambar 12 dibawah ini.

sepakbola seharusnya mer	1	True	usahakan untuk tidak hate sp	0	False
sepakbola itu hiburan raky	1	True	mendukung lewat doa	1	True
iya min di luar dari sidoarjo	0	True	takut banget bilang bonek ya	0	True
turut berduka cita atas me	1	True	ada bukti kalo bobotoh di ke	0	True
yes poin away yang berhar	1	True	walau agak telat tapi kami te	1	True
trima kasih arema atas per	1	True	yang jauh mendekat yang de	1	True
nek condong salah satu pih	0	True	aing aja males ngeladenin sa	0	True
terima kasih bang salam ha	1	True	yuk sama kawal terus tim ke	1	True
yang kurang baik kita kore	1	True	padahal niat baik dari jakmar	0	True
kata kata buruk ibarat paku	0	True	lambat laun semua akan tau	1	False
semoga semakin dewasa s	1	True	the jak yang maha benar liha	1	False
seorang yang cerdik adalah	0	True	sepak bola adalah persahaba	1	True
kejadian semalam tidak m	1	False	chant tak berfaedah masih te	0	True
rekrut striker yang hausss	0	True	dia tidak ngerti arti pemain c	0	True
yup tiap orang harus mera	1	True	kalo yang komen bukan aren	0	True

Gambar 12. Hasil eksperimen data baru untuk model terpilih

Berdasarkan hasil prediksi oleh model, dari 30 data baru didapatkan akurasi 87%, error rate 13%, berikut table laporan evaluasi dan confusion matrix

Tabel 12. Evaluasi data baru test

	prec	rec	f1	supp	Confusion matrix		
						0	1
0	0.92	0.80	0.86	15	0	12	3
1	0.82	0.93	0.87	15	1	1	14
accuracy			0.87	30			
macro avg	0.87	0.87	0.87	30			
weighted avg	0.87	0.87	0.87	30			

4. KESIMPULAN

Kesimpulan dari riset ini adalah pemilihan model klasifikasi untuk data kecil dan seimbang memerlukan usaha yang sedikit lebih dengan melakukan optimasi dan tuning mulai dari menentunkan baseline hingga tercapainya finalisasi dan berikut yang dapat dicatat sebagai kesimpulan dalam penelitian ini bahwa. Analisa tektual penting dilakukan untuk melihat polarisasi dan subyektifitas makna tweet, artinya kita menjadi faham jika ada kata bermakna negative tapi berada dalam kelas positif dan sebaliknya, dan bukan karena kesalahan prediksi. Kata jackmania menempati rerata tweet positif tertinggi sebesar 88% , frekuensi rerata sebesar 4% dan hmean sebesar 8%, hal ini menunjukan bahwa tweet positif mnegalahkan tweet negatiif. Model Multinomial Naïve Bayes berhasil dibangun dan menghasilkan akurasi optimal sebesar 99% pada data train dan 100% pada data testing, dengan error rate 0.8% dan 0%. Eksperimen data baru menghasilkan prediksi yang bagus untuk 30 entri data test akurasinya 87% dengan error rate 13%.

5. SARAN

Kedepan disarankan untuk mencoba melakukan tuning pada metode vektorisasi, menambahkan stopwords baru yang bermakna negasi. Selain itu perlu juga dilakukan penggunaan metode bagging dan boosting dan jika dirasa masih kurang dapat berlanjut melakukkan metode deep learning.

DAFTAR PUSTAKA

[1] M. F. Ismawan, "Jumlah Penonton Championship Lebih Banyak daripada La Liga dan Serie A," *Detik.Com*, 2018. [Online]. Available: https://sport.detik.com/sepakbola/uefa/3818812/jumlah-penonton-championship-lebih-banyak-daripada-la-liga-dan-serie-a. [Accessed: 11-May-2019].

[2] R. Darmawan, "Empat Klub Teratas Liga 1 2018 dengan Penonton Terbanyak di Stadion hingga Pekan Keempat," *SuperBall.id*, 2018. [Online]. Available: https://superball.bolasport.com/read/331441565/empat-klub-teratas-liga-1-2018-dengan-penonton-terbanyak-di-stadion-hingga-pekan-keempat?page=2. [Accessed: 11-Apr-2019].

[3] T. N. Habibie, "Hubungan Antara Fanatisme Dan Solidatritas Sosial Di Komunitas ICI MORATTI Regional Malang," *J. Mhs. Sosiologi, Univ. Brawijaya*, vol. 2, no. Novembere, 2014.

[4] V. Widiastuti, "Haringga Sirla Tewas Dikeroyok, Bermula dari KTP Korban Dirazia Sejumlah Suporter Persib Bandung," *Www.Tribunnews.Com*, 2018. [Online]. Available: https://www.tribunnews.com/nasional/2018/09/24/haringga-sirla-tewas-dikeroyok-bermula-dari-ktp-korban-dirazia-sejumlah-suporter-persib-bandung. [Accessed: 11-May-2019].

[5] Y. S. Mahardika and E. Zuliarso, "Analisis Sentimen Terhadap Pemerintahan Joko Widodo Pada Media Sosial Twitter Menggunakan Algoritma Naives Bayes," *Pros. SINTAK 2018*, no. 2015, pp. 409–413, 2018.

[6] A. K. B. A. Putra, M. A. Fauzi, B. D. Setiawan, and E. Setiawati, "Identifikasi Ujaran Kebencian Pada Facebook Dengan Metode Ensemble Feature Dan Support Vector Machine," *J. Pengemb. Teknol. Inf. Dan Ilmu Komput.*, vol. 2, no. 12, 2018.

[7] A. R. T. Lestari, R. S. Perdana, and M. A. Fauzi, "Analisis Sentimen Tentang Opini Pilkada Dki 2017 Pada Dokumen Twitter Berbahasa Indonesia Menggunakan Näive Bayes dan Pembobotan Emoji," *J. Pengemb. Teknol. Inf. dan Ilmu Komput.*, vol. 1, no. 12, pp. 1718–1724, 2017.

[8] S. Madhu, "An approach to analyze suicidal tendency in blogs and tweets using Sentiment Analysis," *Int. J. Sci. Res. Comput. Sci. Eng.*, vol. 6, no. 4, pp. 34–36, 2018.

[9] H. Li, "Sentiment Analysis and Opinion Mining on Twitter With Gmo Keyword," North Dakota State University, 2016.

[10] B. Das and S. Chakraborty, "An Improved Text Sentiment Classification Model Using TF-IDF and Next Word Negation," 2018.

[11] M. de Vries, "Machine Learning for Sentiment Analysis of Children's Diaries," Utrecht University, 2017.

[12] K. Ghag and K. Shah, "SentiTFIDF – Sentiment Classification using Relative Term Frequency Inverse Document Frequency," *Int. J. Adv. Comput. Sci. Appl.*, vol. 5, no. 2, pp. 36–43, 2014.

[13] D. UmniySalamah, "Implementation of Logistic Regression Algorithm for Complaint Text Classification in Indonesian Ministry of Marine and Fisheries Abstract :," *Int. J. Comput. Tech.*, vol. 5, no. 5, pp. 74–78, 2018.

[14] K. Korovkinas and G. Garšva, "Selection of intelligent algorithms for sentiment classification method creation," *CEUR Workshop Proc.*, vol. 2145, pp. 152–157, 2018.

[15] A. Tyagi and N. Sharma, "Sentiment Analysis using logistic regression and effective word score heuristic," *Int. J. Eng. Technol.*, vol. 7, no. 2, pp. 20–23, 2018.

[16] D. Jurafsky, "Text Classification and Naive Bayes," 2016.

[17] A. M. Muscolino, "Sentiment Analysis, a Support Vector Machine Model Based on

Social Network Data," *Int. J. Res. Eng. Technol.*, vol. 07, no. 07, pp. 154–157, 2018.

[18] B. M. and V. B., "Sentiment Analysis using Support Vector Machine based on Feature Selection and Semantic Analysis," *Int. J. Comput. Appl.*, vol. 146, no. 13, pp. 26–30, 2016.

[19] M. K. D. Ms. Gaurangi Patil1, Ms. Varsha Galande2, Mr. Vedant Kekan3, "Sentiment analysis using Support Vector Machine," *I4CT 2014 - 1st Int. Conf. Comput. Commun. Control Technol. Proc.*, vol. 2, no. 1, pp. 333–337, 2014.

[20] S. A. Aljuhani and N. S. Alghamdi, "A comparison of sentiment analysis methods on Amazon reviews of Mobile Phones," *Int. J. Adv. Comput. Sci. Appl.*, vol. 10, no. 6, pp. 608–617, 2019.

[21] M. Abbas, K. Ali Memon, and A. Aleem Jamali, "Multinomial Naive Bayes Classification Model for Sentiment Analysis," *IJCSNS Int. J. Comput. Sci. Netw. Secur.*, vol. 19, no. 3, p. 62, 2019.

[22] A. A. Farisi, Y. Sibaroni, and S. Al Faraby, "Sentiment analysis on hotel reviews using Multinomial Naïve Bayes classifier," *J. Phys. Conf. Ser.*, vol. 1192, no. 1, 2019.

[23] H. M. Ismail, S. Harous, and B. Belkhouche, "A Comparative Analysis of Machine Learning Classifiers for Twitter Sentiment Analysis," *Res. Comput. Sci.*, vol. 110, no. April, pp. 71–83, 2016.

[24] M. Ahmad, S. Aftab, and I. Ali, "Sentiment Analysis of Tweets using SVM," *Int. J. Comput. Appl.*, vol. 177, no. 5, pp. 25–29, 2017.

[25] M. Rezwanul, A. Ali, and A. Rahman, "Sentiment Analysis on Twitter Data using KNN and SVM," *Int. J. Adv. Comput. Sci. Appl.*, vol. 8, no. 6, pp. 19–25, 2017.

[26] J. Jotheeswaran and Y. S. Kumaraswamy, "Opinion mining using decision tree based feature selection through Manhattan hierarchical cluster measure," *J. Theor. Appl. Inf. Technol.*, vol. 58, no. 1, pp. 72–80, 2013.

[27] A. S. and C. R. Bharathi, "Sentiment Classifi cation using Decision Tree Based Feature Selection Sentiment Classi fi cation using Decision Tree Based Feature Selection," *IJCTA*, vol. 9(36), no. January, pp. 419–425, 2016.

[28] M. Guia, R. R. Silva, and J. Bernardino, "Comparison of Naive Bayes, support vector machine, decision trees and random forest on sentiment analysis," *IC3K 2019 - Proc. 11th Int. Jt. Conf. Knowl. Discov. Knowl. Eng. Knowl. Manag.*, vol. 1, pp. 525–531, 2019.

[29] Jonbakerfish, "TweetScraper," *https://github.com*, 2020. [Online]. Available: https://github.com/jonbakerfish/TweetScraper. [Accessed: 12-Nov-2019].

[30] C. Gibrat, *Continuous Gibrat ' s Law and Gabaix ' s Derivation of Zipf ' s Law*, no. June 2014. 2010.

[31] L. Wentian, "Zipf's Law everywhere," *Glottometrics*, vol. 5, no. June, pp. 14–21, 2002.