MIND (Multimedia Artificial Intelligent Networking Database) Journal

MIND Journal | ISSN (p): 2528-0015 | ISSN (e): 2528-0902 | Vol. 8 | No. 2 | Halaman 230 - 241
DOI : https://doi.org/10.26760/mindjournal.v8i2.230-241 Desember 2023

Ektraksi Fitur menggunakan *Regular Expression* pada Naïve Bayes Classifier untuk Analisis Sentimen

ACEP IRHAM GUFRONI[1], SITI YULIYANTI[2], EUIS NUR FITRIANI DEWI[3]

[1]Program Studi Sistem Informasi Universitas Siliwangi
[2,3]Program Studi Informatika Universitas Siliwangi
Email : acep@unsil.ac.id

Received 19 September 2023 **|** *Revised* 03 November 2023 **|** *Accepted* 28 November 2023

ABSTRAK

Regular expression atau regex merupakan metode ekstraksi fitur yang menemukan substring pada sebuah teks yang cocok dengan harapan dapat meningkatkan kompleksitas waktu atau akurasi dengan melakukan preprocessing teks. Permasalahan praproses teks salah satunya kurang memperhatikan ektraksi fitur untuk proses klasifikasi sentiment, sehingga akurasi yang diperoleh kurang optiomal. Inovasi utama dari pendekatan penelitian ini yaitu mengembangkan pengklasifikasi teks berbasis ekspresi reguler sehingga menghasilkan performance kinerja algoritma yang baik. Tahapan penelitian ini, yaitu pengumpulan dataset lalu mengklasifikasikan sentiment dengan Naïve Bayes dan dalam praproses teks dilakukan ektraksi fiitur regular expression. Hasil rata-rata akurasi yang dihasilkan dengan ekstraksi ciri sebesar 88,05% dan yang tidak menggunakan 79,26% sehingga dapat disimpulkan bahwa penggunaan ekstraksi fitur pada praproses dapat meningkatkan akurasi sebesar 8,08% dari 1000 data latih dan 400 data uji.

***Kata kunci**: ekstraksi fitur, regex, regular expression, substring*

ABSTRACT

Regular expression or regex is a feature extraction method that finds matching substrings in a text in hopes of increasing time complexity or accuracy by preprocessing the text. One of the problems with text preprocessing is the lack of attention to feature extraction for the sentiment classification process, so the accuracy obtained is not optimal. This research stage begins with collecting a dataset and then classifying sentiment using Naïve Bayes, which pre-processes the text by extracting features with regular expressions. The main innovation of this research approach is to develop a text classifier based on regular expressions so as to produce good algorithm performance. The average accuracy produced by feature extraction is 88.05% and 79.26% is not used, so it can be concluded that the use of feature extraction in pre-processing can increase accuracy by 8.08% from 1000 training data and 400 test data.

***Keywords**: extraction feature, regex, regular expression, substring*

1. PENDAHULUAN

Kemampuan mencari substring yang cocok dengan *regular expression* atau *regex* dalam teks yang diproses sebelum tahapan klasifikasi membantu aplikasi yang tak terhitung jumlahnya **(Gibney & Thankachan, 2021)**. Ini terbukti dari sudah banyak dibangun dalam bentuk package atau library *regex* di perangkat lunak dan bahasa pemrograman yang memudahkan peneliti. Beberapa mesin pencarian untuk repository kode yang secara umum digunakan untuk pencarian *string* pada database seperti SQL dan non-relasional. Namun yang menjadi permasalahan yaitu teks yang digunakan untuk objek dalam penelitian sudah ada sebelum *regular expression* disediakan pada *package* sehingga kebanyakan peneliti berharap dapat langsung mengimplementasikan algoritma setelah dilakukan praproses yang didalamnya sudah ada ektraksi fitur **(Cox, 2012).** Hal tersebut dapat meningkatkan akurasi sehingga dapat dilakukan perbandingan praproses teks yang menggunakan ektraksi fitur dengan *regex* dengan yang tidak.

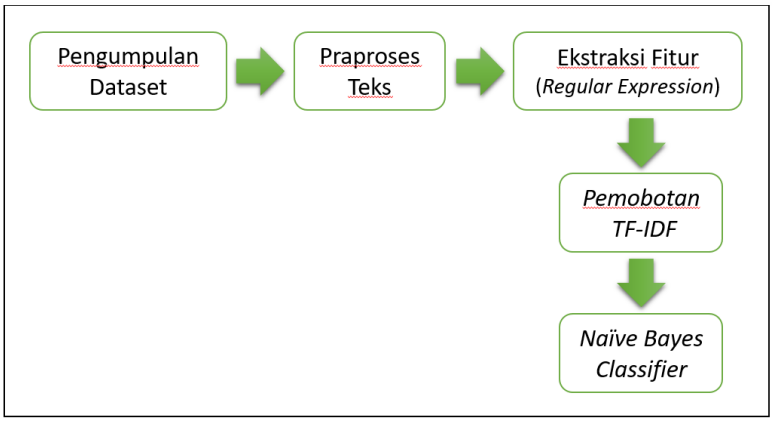
Regex merupakan cara klasik untuk pencocokan pola string dan dianggap sebagai alat yang efisien dalam berbagai domain aplikasi seperti ekstraksi informasi dan penambangan teks **(Nisa dkk., 2019)**. Meskipun membuat ekspresi reguler secara manual jelas memakan waktu, rawan kesalahan, dan bergantung pada pengalaman, ada sedikit pekerjaan yang menunjukkan bahwa ekspresi reguler yang dibuat secara otomatis mampu memberikan kinerja yang sebanding dengan pekerjaan manual (Cui dkk., 2019). Salah satu tantangan utama memahami regular ekspresi oleh komputer adalah ruang pencarian besar karena sejumlah besar kandidat kata dan kombinasinya melalui operator yang berbeda. Selain itu, sebagian besar pekerjaan sebelumnya dirancang tanpa mempertimbangkan kemampuan model yang akan disempurnakan oleh pakar manusia **(Setiawan dkk., 2021)**.

Penelitian ini mengangkat penggunaan *regex* berdasarkan referensi jurnal sebelumnya, seperti tahapan praproses dapat mereduksi fitur yang tidak sesuai sehingga akurasi meningkat **(Yuliyanti & Sholihah, 2021).** Hasil penelitian yang mengimplementasikan *regex* dengan index pada Naive Bayes dan Support Vector Machines, mampu meningkatkan akurasi sebesar 9% dalam presisi dan 4,5% dalam *recall* **(Cui dkk., 2019).** Kemudian penambahan *regex* juga dilakukan pada penelitian dengan akurasi 94,44%, presisi 94,44%, perolehan kembali 92%, dan skor_f1 93%. Untuk MLP Classifier dan akurasi 96,42%, presisi 94,44%, recall 96,66% dan f1_score 95,56% untuk Naïve Bayes **(Asian dkk., 2022).** Sedangkan pada penelitian **(Aprisadianti, 2021),** *regex* mampu mengidentifikasi pada level rendah yaitu menemukan sebuah penggalan kata sedangkan pada level tinggi mampu mengontrol data teks seperti mencari, menhapus dan mengubah **(Setiawan dkk., 2021).** Sedangakan perbandingan akurasi pada analsisi sentiment antara Naïve Bayes Classifier dengan support Vector Machine menunjukan akurasi tertinggi yaitu Naïve Bayes walaupun belum ditambahkan ekstraksi fitur **(Raharjo dkk., 2022**)**.** Penelitian dengan menambahkan ekstraksi fiitur pada Naïve Bayes Classifier juga dilakukan sehingga didapati tingkat Acuraccy 90.18%, Precission 96.61%, Recall 83.43%, AUC 0.988% **(Pratama Putra dkk., 2022)** sedangkan pada penelitian Naïve Bayes Classifier yang tidak menggunakan *regex* akurasi hanya mencapai 71,0% **(Legianto, 2019).** Penelitian ini mengambil beberapa metode dari penelitian terdahulu dengan penambahan ektraksi fitur menggunakan regular expression pada Naïve Bayes Classifier yang belum dilakukan pada penelitain terdahulu bertujuan meningkatkan akurasi dalam proses klasifikasi sentiment, baik dari tahapan pengumpulan data ataupun praproses teks.

2. METODE PENELITIAN

2.1. Kerangka Penelitian

Penelitian ini terdiri dari beberapa tahapan yaitu pengumpulan dataset, praproses teks, ekstraksi fitur dan yang terakhir klasifikasi menggunakan Naïve Bayes sebagaimana diilustrasikan pada Gambar 1. Untuk memastikan adanya peningkatan akurasi pada saat proses klasifikasi, penelitian ini juga melakukan *training* dan *testing* untuk dataset yang tidak menggunakan ekstraksi fitur setelah tahapan praproses.



Gambar 2. Kerangka Penelitian

2.2. Pengumpulan Dataset dan NoSQL

Tahapan pengumpulan dataset, sebagaimana ditunjukan pada Gambar 1 merupakan tahapan awal penelitian ini. Dataset berasal dari proses teknik *crawling* twitter menggunakan *Twitter Stream* API, dengan mengakses *endpoint* ini dengan parameter yang diberikan, twitter akan merespons data dalam format JSON. Format ini sangat cocok dikonsumsi oleh aplikasi *web* dalam hal ini adalah Node JS, selain itu ukuran file JSON ini sangat ringan. Adapun versi Standard v1.1 *Twitter API* dengan API GET *search/tweets* yang dapat memberikan data *realtime* sebanyak 400 kata kunci yang dibatasi per 15 menit. *Twitter API* akan memberikan hasil data berupa *JSON (Javascript Object Notation),* format JSON menyimpan informasi terstruktur dan digunakan untuk mentransfer data antara server dengan klien. Data JSON ini yang akhirnya akan dikonsumsi oleh aplikasi web.

Crawling dilakukan sesuai dengan parameter periode tertentu selama 7 hari sehingga dapat diperoleh data yang sesuai dengan penelitian dengan kata kunci, penelitian ini menggunakan 1400 *tweet* pada dataset dibagi menjadi dua yaitu 1000 data *training* dan 400 *tweet* untuk data *testing* jika dlam persentase pembagian dataset 60% : 40%. Berdasarkan penelitian tentang Perbandingan Rasio Split Data Training dan Data Testing dataset dengan model 70% : 30% menunjukan pola naik turun yang sangat dinamis dan mmembuat nilai prediksi model menagkap pola nilai prediksi seperti nilai asli sehingga akurasinya lebih naik **(Adinugroho, 2022)**. Data *training* dan data *testing* juga disimpan dalam dataset json kemudian diproses otomatis dengan luaran kelas sentimen **(Yuliyanti & Sholihah, 2021).**

NoSQL merupakan penyimpanan data atau *datastore*, yang memiliki ukuran fleksible, penyimpanan besar, hemat server, sehingga dapat meringankan fungsi Database Administrator. Penelitian ini menggunakan NoSQL MongoDB karena pemrosesan dalam *database* lebih cepat.

2.3. Analisis Sentimen

Sentimen adalah informasi tekstual yang berisi tentang fakta dan opini baik dari media cetak, media social maupun dokumen yang menyatakan persepsi subjektif terhadap suatu kejadian oleh seseorang **(Alnaz & Maharani, 2021).** Analisis sentiment merupakan studi komputasi yang berhubungan dengan pendapat serta berorientasi pada pengolahan bahasa alami atau NLP (MZ dkk., 2022), dengan SentiWordNet dan WordNet yang sangat akrab dikalangan peneliti.

Analisis sentimen terdiri dari beberapa subprose yaitu *Subjectivity Classification* (penentuan kalimat opini), *Orientation Detection* (tahapan kelanjutan klasifikasi), dan *Opinion Holder and Target Detection* (menentukan bagian Opinion Holder atau Target) **(Talita dkk., 2021).** Penelitian ini fokus membahas mengenai Orientation Detection terhadap suatu kalimat dalam opini, yaitu bagaimana menentukan opini positif dan negatif, maupun netral dalam sebuah *tweet* **(Setiawan dkk., 2021).** Pada penelitian ini untuk menganalisi pendapat masyarakat tentang kasus *bullying* yang marak dikalangan anak-anak, sehingga kata kunci yang diambil pada *tweet* di twitter seperti *bullying*, rudapaksa, kekerasan pada anak, dan istilah lainnya yang mengandung intimidasi.

2.4. Praproses Teks

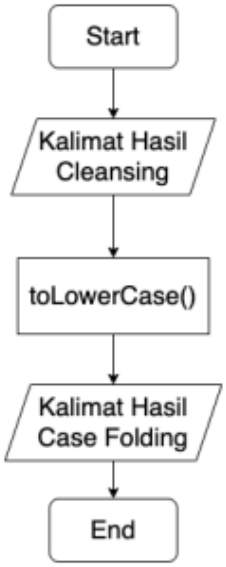
Praproses bertujuan untuk membersihakan dataset hasil *crawling* yang masih memiliki noise, iklan, link *url* atau hal lainnya yang dapat memperngaruhi kompleksitas kinerja algoritma yang akan diimplementasikan. Praproses penelitian ini yaitu *cleansing, case folding, tokenizing, stemming dengan menggunakan stopword removal.*

2.4.1 *Cleansing*

Bertujuan untuk menghapus data inkonsisten, tidak cocok, dan *noise*. Data dari *database* perusahaan atau hasil eksperimen biasanya berisi data yang tidak lengkap seperti data hilang, data salah, atau kesalahan ketik. Selain itu, beberapa karakteristik data tidak terkait dengan hipotesis ekstraksi data tertentu. *Cleansing* mempengaruhi kinerja praproses agar minim kompleksitas **(Legianto, 2019).**

2.4.2 *Case Folding*

Tahapan ini merubah menjadi huruf kecil dan menghilangkan angka dan tanda baca **(Pratama Putra dkk., 2022**), seperti diilustrasikan pada Gambar 2.



Gambar 2. *Flowchart* Proses *Case Folding*

2.4.3 *Tokenizing*

Tahapan ini memotong kata setiap kalimat, sehingga *tweet* dipecah menjadi token lalu dihilangkan tanda baca, simbol tidak bermakna.

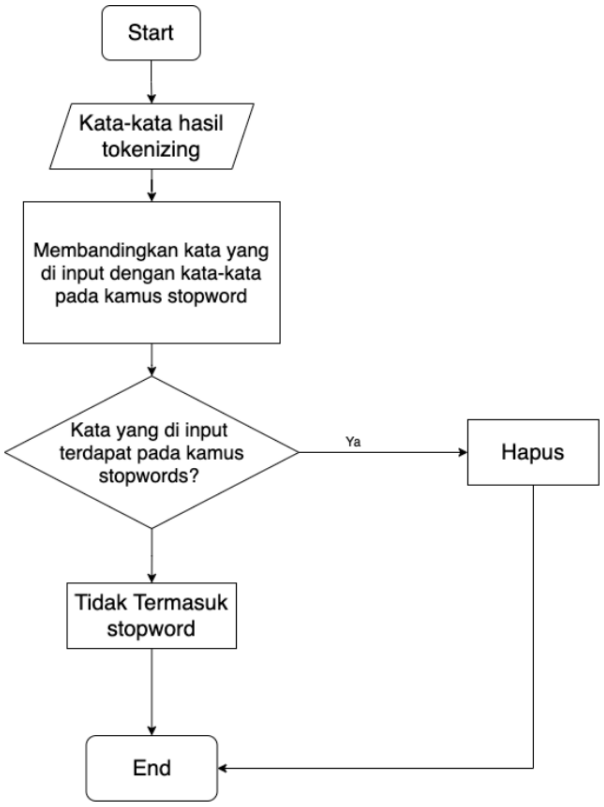
Gambar 3. *Flowchart* Proses *Tokenzining*

Kalimat *inputan* yaitu hasil *case folding*, kemudian membagi kata pada tweet berdasarkan spasi, koma(,) dan titik (.)**(Nisa dkk., 2019)** dan hanya bagian berarti yang akan menjadi kata hasil untuk proses selanjtnya seperti diilustrasikan Gambar 3.

2.4.4 *Filter* Redudansi dan *Stopword Removal*

Filter redudansi bertujuan untuk menaikan penghitungan frekuensi kata dalam pembobotan dalam penelusuran kata yang bersinonim yang terdokumentasi kamus pada database **(Darwis dkk., 2021) (Apriani dkk., 2019).**

Stopword removal bersumber dari *stopword list,* misalnya "oleh", "pada", "di", "sebuah", "karena" dan kata lain yang sejenis. Jika kata termasuk *stopword list* maka penghapusan diproses. Kamus stopword yang menjadi sumber dari Node JS yaitu stopwords-iso/stopwords-id. Dalam library ini telah mendukung kamus *stopword* berbahasa Indonesia, tidak semua kata dalam Kamus Stopword digunakan sebagai kata kunci dalam penelitian ini. Hal ini dikarenakan istilah tersebut mempengaruhi makna dan nilai emosi, terutama nilai-nilai emosi yang negatif.



Gambar 4. Proses *Stopword Removal*

2.4.5 *Stemming*

Tahapan ini mengubah kata yang memiliki imbuhan baik diawal atau di akhir menjadi kata dasar dengan perbandingan kamus kata dasar dan kata kunci kamus online bahasa Indonesia berjumlah 29.932 sedangkan algoritma Stemming pada penelitian bersumber dari Nazief & Andriani **(Herlingga dkk., 2020).**

2.4.6 *Regular expression* atau *Regex* dan Pembobotan TF-IDF

Term Frequency (TF) adalah perhitungan bobot tiap *term* dalam teks, dengan jumlah kemunculan term. *Document Frequency* (DF) adalah jumlah dokumen terhadap suatu term dan termasuk metode *feature selection* sederhana yang minimum waktu komputasinya (Arini dkk., 2020)**.** *Inverse Document Frequency* (IDF) merupakan kehadiran term pada lokasi test secara menyeluruh. *Term Frequency Inverse Document Frequency* (TF-IDF) **(Alnaz & Maharani, 2021)** adalah hitung bobot sesudah dilakukan ekstrasi dokumen yang terintegrasi antar *term frequency* (TF) dan *inverse document frequency* (IDF), rumusan pada Persamaan (1) dengan N sebagai jumlah total kata dalam *tweet* sedangkan n_j adalah jumlah kata yang muncul pada *tweet* **(Asian dkk., 2022).**

$$\mathrm{IDF} = Log_2(N/n_j) + 1 = Log_2(N) - Log_2(n_j) + 1 \quad (1)$$

Langkah meminimumkan kemunculan noise dengan menghilangkan fitur yang tidak relevan, sehingga menambah bear nilai akurasi **(Ayu & Kemala, 2017).** Penelitian ini menggunakan *regular expression* atau *regex* untuk ektraksi fitur. Setiap solusi untuk masalah (pengklasifikasi berbasis *regex* untuk satu kelas) dikodekan sebagai vektor *regex* gabungan seperti pada Persamaan (2) **(Cui dkk., 2019).**

$$< R1, R2, \ldots, Ri, \ldots > \quad (2)$$

Tabel 1. Fungsi dan Terminal yang digunakan dalam Model

Nama	Label	Deskripsi
word	*W*	daftar kata kunci yang diekstraksi dari kumpulan teks target menggunakan heuristik n-gram
expression	*E*	ekspresi atau istilah yang diperoleh melalui kombinasi kata dan fungsi
AND	.	berfungsi untuk menguji logika `AND` dua ekspresi
OR	|	berfungsi untuk menguji logika `OR` dua ekspresi
Distance	{*a,b*}	berfungsi untuk menguji apakah jarak antara dua kata w_1 dan w_2 dalam range [a,b]
NOT	#_#	berfungsi untuk meniadakan ekspresi tertentu

Untuk memeriksa apakah suatu teks inkuiri milik kelas tertentu, *regex* dalam vektor dicocokkan secara berurutan dalam urutan vektor yang sama dengan teks inkuiri yang dipertimbangkan. Penyelidikan teks diklasifikasikan ke kelas tertentu jika dicocokkan dengan salah satu *regex* di pengklasifikasi. Oleh karena itu, tugas ini diperlakukan sebagai klasifikasi biner **(Cui dkk., 2019).** Setiap *Regex* Ri diturunkan melalui kombinasi fungsi dan terminal yang ditentukan dalam Tabel 1 dan mengikuti struktur global dua bagian *Pi* dan *Ni* yang digabungkan dengan fungsi NOT yang dilambangkan sebagai #_#**(Cui dkk., 2019).** Artinya, setiap *Regex* Ri memiliki format berikut:

$$Ri = Pi * (\#\_\#(Ni)) \quad (3)$$

di mana *Pi* mencoba untuk mencocokkan semua permintaan teks positif dan Ni digunakan untuk memfilter permintaan teks potensial yang salah dicocokkan oleh *Pi*. Perhatikan bahwa dalam keadaan ketika bagian positif *Pi* saja sudah cukup tepat untuk klasifikasi yang benar (yaitu, presisi *Pi* melebihi ambang batas yang telah ditentukan), *Ni* tidak diperlukan. Dalam hal ini, struktur *Ri* disederhanakan menjadi Persamaan (4).

$$Ri = Pi \quad (4)$$

Mari kita definisikan ekspresi ei sebagai kumpulan kata yang digabungkan dengan fungsi OR, yang dapat dinyatakan sebagai Persamaan (5).

$$e_i = (w_{i1}|w_{i2}w_{i1}|\ldots|w_{in}) \quad (5)$$

di mana n adalah jumlah kata dalam ekspresi **(Cui dkk., 2019).**

2.4.7 *Naïve Bayes Classifier*

Implementasi prior probability pada Naïve Bayes yang berasal dari data *training* dari kelas yang kemunculannya berkontribusi pada masing-masing fitur **(MZ dkk., 2022).** Proses klasifikasi dihitung dengan P(H|X), dengan menggunakan Persamaan (6):

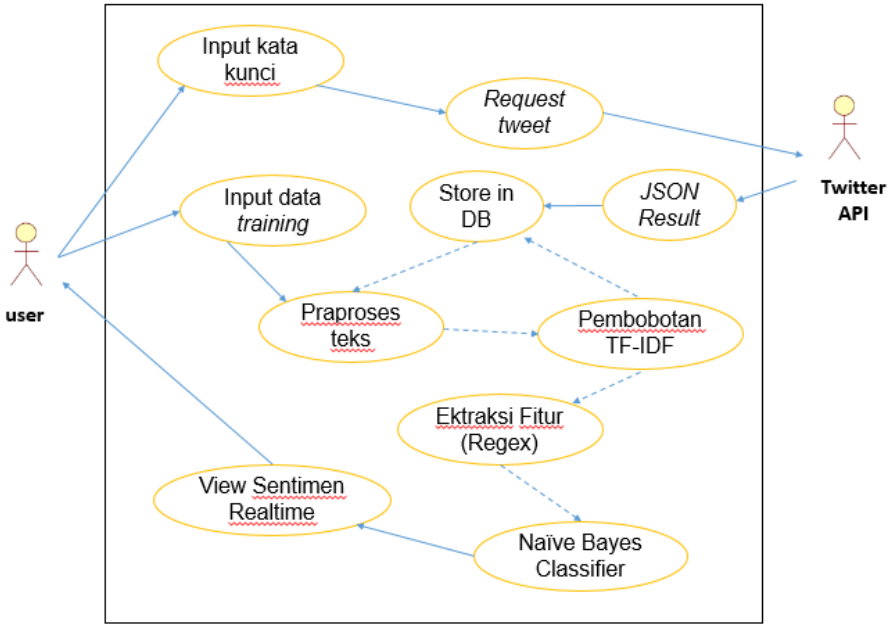
$$P(H \mid X) = (P(X|H)P(H))/(P(X)) \quad (6)$$

X : contoh data dengan kelas tidak berlabel
H : Hipotesa X yang merupkan data dengan label C
P(H|X) : Peluang kebenaran sampel X yang diobservasi
P(X|H) : Peluang sampel X, dengan asumsi benar
P(H) : Peluang hipotesa H
P(X) : Peluang data sampel

3. HASIL DAN PEMBAHASAN

Aplikasi yang dibangun pada penelitian ini memiligi beberpa fungsi yang digambarkan pada Gambar 5, mulai pengumpulan dataset yaitu penginputan kata kunci oleh *user* yang kemudian dilanjutkan menjadi sebuang fungsi pemanggilan "request tweet" menggunakan Twitter API, yang kemudian dataset disimpan dalam JSON Result karena menggunaka NoSQL.

Dilanjutkan dengan praproses teks, pembobotan menggunakan TF-IDF dan ektraksi fitur menggunakan *regular expression* atau regex serta proses akhir dengan Naïve Bayes Classifier sehingga analisis sentimen dapat ditampilkan secara *realtime*.



Gambar 5. Use Case Aplikasi Analisis Sentiment

Pada penelitian ini, aplikasi pemodelan yang dikembangkan adalah analisa sentimen pada klasifikasi data *tweet* dengan algoritma Naïve Bayes Classifier yang setelah praproses teks dan pembobotan dilakukan ektraksi fitur dengan *regular expression* atau *regex* untuk meningkatkan akurasi sehingga proses evaluasi dapat menunjukan keakuratan metode Naïve Bayes Classifier meningkat. Sebagaimana digambarkan use case pada Gambar 4. Proses klasifikasi dilakukan dengan membagi data yang telah di ektraksi fitur kedalam 2 bagian yaitu 1000 *tweet* data *training* dan 400 *tweet* sebagai data *testing*.

Proses pembobotan 1 yaitu menghitung jumlah frekuensi tiap kata pada tiap dokumen (TF), dilanjutkan dengan menghitung banyaknya *tweet* serta query yang memuat kata, lalu

menghitung IDF dengan Persamaan 7 dilanjut dengan TF-IDF menggunakan Persamaan 8 dan untuk sebagian sampel perhitungan *tweet* yang digunakan dari 10 *tweet* yang diilustrasikan dengan simbol T1 sampai dengan T10 berdasarkan frekuensi tertinggi ditunjukan Tabel 2.

Tabel 2. Sample Hasil Pembobotan TF-IDF

Kata	TF										DF	IDF
	T1	T2	T3	T4	T5	T6	T7	T8	T9	T10		
rudapaksa	0	1	0	0	0	0	0	0	0	1	2	0,6120
kekerasan	0	0	0	0	1	1	1	1	2	0	6	0,2107
perundungan	1	0	1	0	0	1	1	0	0	0	4	0,3110
korban	0	1	0	0	0	0	0	1	0	0	2	0,6120
bulliying	1	0	0	0	0	1	0	1	0	2	5	0,2508
menghina	0	0	1	0	2	0	0	0	0	0	3	0,4113
ancam	0	1	0	0	0	0	0	1	1	0	3	0,4113
hina	2	0	0	1	0	0	0	1	0	0	4	0,3110
psikis	0	0	1	0	0	0	0	1	0	0	2	0,6120
lindungi	0	1	0	1	0	1	0	0	0	0	3	0,4113
penindasan	0	0	1	0	0	1	0	0	0	0	2	0,6120
pengucilan	0	0	0	1	0	1	0	0	1	0	3	0,4113

Hasil klasifikasi ditunjukan pada Tabel 3 dan Tabel 4, dimana akurasi, *precision* dan *recall* lebih tinggi untuk klasifikasi yang menggunakan ektraksi fitur *regex* dibandingan yang tidak menggunakan, sehingga terlihat kenaikan akurasi pada saat dataset diuji menggunakan ektraksi fitur menggunakan regex yang dikemudian dilanjutkan dengan klasifikasi Naive Bayes Maka dapat disimpulkan juga bahwakenaikan akurasi menggunakan Naive bayes Classifier selain bergantung pada dataset yang digunakan juga bergantung pada praproses, yang dalam penelitian ini dilakukan menggunakan ektraksi fitur regex.

Tabel 3. Hasil Klasifikasi dengan Penambahan Ektraksi Fitur *Regex*

	Naïve Bayes dengan Ektraksi Fitur *Regex*				
Kelas	**N(truth)**	**N (classified)**	**Acuraccy**	**Precision**	**Recall**
Positif	400	110	87,05%	84,45%	94,95%
Negatif	130	200	84,94%	83%	77%
Netral	370	190	92,17%	89,95%	80,947%

Pada Tabel 3 kenaikan akurasi mencapai 8,08% dimana dari 1000 data testing yang digunakan terdapat respon positif sebanyak 400 tweet, 130 negatif dan 370 netral. Sedangkan untuk ektraksi fitur menggunakan regex 420 tweet, 180 negatif dan 400 netral, hal ini membuktikan bahwa sentiment masyarakat terhadap tweet yang mengandung bullying memiliki respon positif dan netral hampir sama dimana hal tersebut menunjukan jika netral berarti masih banyak masyarakat yang bersentimen bahwa kasus bullying memiliki pembenaran dengan hubungan sebab-akibat.

Tabel 4. Hasil Klasifikasi Tanpa Ektraksi Fitur *Regex*

	Naïve Bayes dengan Ektraksi Fitur *Regex*				
Kelas	**N(truth)**	**N (classified)**	**Accuracy**	**Precision**	**Recall**
Positif	420	100	77%	75%	71,4%
Negatif	180	220	82,02%	72,727%	88,88%
Netral	400	380	78,047%	78,947%	75%

4. KESIMPULAN

Hasil pengujian menunjukan simpulan terhadap analisis sentiment dengan tiga kelas yaitu, positif, negative dan netral dnegan mengimplementasikan Naïve bayes Classifier menggunakan ektraksi fitur dengan rengular expression atau regex mampu meningkatkan performa yaitu diperoleh *accuracy* sebesar 87,05%, *precission* sebesar 84,45%, *recall* sebesar 94,95% untuk kelas positif sedangkan tanpa ekstrasksi fitur *regex* diperoleh *accuracy* sebesar 82,02%, precission sebesar 72,727%, *recall* sebesar 88,88%. Sedangkan rata-rata akurasi untuk Klasifikasi Naïve Bayes Classifier dengan ektraksi fitur *regex* sebesar 88,05% sednagkakan yang tanpa sebesar 79,26%, sehingga dapat disimpulkan kenaikan akurasi mencapai 8,08% setelah menggunakan ektraksi fitur regex.

Berdasarkan hasil analisis dan kesimpulan, untuk penelitian selanjutnya dapat menambahkan dataset yang berasal dari Facebook dan Instagram agar lebih variatif dan untuk praproses dapat ditambahkan lagi dengan implementasi gabungan n kata atau k-gram untuk menghasilkan akurasi yang lebih tinggi dan lebih spesifik pada pencarian makna kata.

UCAPAN TERIMA KASIH

Bagian ini berisi ucapan terima kasih kepada LPPM Universitas Siliwangi yang sudah mendanai penelitian ini dan seluruh pihak yang terlibat langsung maupun tidak dalam terselesaikannya penelitian ini.

DAFTAR RUJUKAN

Adinugroho, R. (2022). *Perbandingan Rasio Split Data Training dan Data Testing menggunakan Metode LSTM dalam MemprediksiHarga Indeks Saham.* Jakarta: UIN Syarif Hidayatulloh.

Alnaz, F. S., & Maharani, W. (2021). *Analisis Emosi Melalui Media Sosial Twitter Dengan Menggunakan Metode Naïve Bayes dan Perbandingan Fitur N-gram dan TF-IDF*. Laporan Penelitian Hal 1–14.

Apriani, R., Gustian, D., Program, S., Sistem, I., Putra, U. N., Indonesia, S., Raya, J., Kaler, C., 21, N., & Sukabumi, K. (2019). Analisis Sentimen dengan Naïve Bayes Terhadap Komentar Aplikasi Tokopedia. *Jurnal Rekayasa Teknologi Nusa Putra*, *6*(1), 54–62.

https://rekayasa.nusaputra.ac.id/article/view/86

Aprisadianti, S. N. (2021). Analisis Sentimen Twitter terhadap Content Creator Sisca Kohl Menggunakan Regular Expression. *Makalah IF2211 Strategi Algoritma*, *13519040*.

Arini, A.-, Wardhani, L. K., & Octaviano, D.-. (2020). Perbandingan Seleksi Fitur Term Frequency & Tri-Gram Character Menggunakan Algoritma Naïve Bayes Classifier (Nbc) Pada Tweet Hashtag #2019gantipresiden. *Kilat*, *9*(1), 103–114. https://doi.org/10.33322/kilat.v9i1.878

Asian, J., Dholah Rosita, M., & Mantoro, T. (2022). Sentiment Analysis for the Brazilian Anesthesiologist Using Multi-Layer Perceptron Classifier and Random Forest Methods. *Jurnal Online Informatika*, *7*(1), 132. https://doi.org/10.15575/join.v7i1.900

Ayu, S., & Kemala, C. (2017). *Penerapan Regular Expression dalam Opinion Mining pada Twitter untuk Survei Opini Politik*. www.search.twitter.com

Cox, R. (2012). Regular Expression Matching with a Trigram Index. In *Website*. https://swtch.com/~rsc/regexp/regexp4.html

Cui, M., Bai, R., Lu, Z., Li, X., Aickelin, U., & Ge, P. (2019). Regular expression based medical text classification using constructive heuristic approach. *IEEE Access*, *7*, 147892–147904. https://doi.org/10.1109/ACCESS.2019.2946622

Darwis, D., Siskawati, N., & Abidin, Z. (2021). Penerapan Algoritma Naive Bayes Untuk Analisis Sentimen Review Data Twitter Bmkg Nasional. *Jurnal Tekno Kompak*, *15*(1), 131. https://doi.org/10.33365/jtk.v15i1.744

Gibney, D., & Thankachan, S. V. (2021). Text indexing for regular expression matching. *Algorithms*, *14*(5). https://doi.org/10.3390/a14050133

Herlingga, A. C., Prismana, I. P. E., Prehanto, D. R., & Dermawan, D. A. (2020). Algoritma Stemming Nazief & Adriani dengan Metode Cosine Similarity untuk Chatbot Telegram Terintegrasi dengan E-layanan. *Journal of Informatics and Computer Science (JINACS)*, *2*(01), 19–26. https://doi.org/10.26740/jinacs.v2n01.p19-26

Legianto, S. (2019). *Implementasi Text Mining Untuk Mendeteksi Hate Speech Pada Twitter*. 60.

MZ, Y., Bororing Edwin, J., Rahayu, S., & Faharani, F. (2022). Analisis Sentimen Masyarakat terhadap Tindakan Vaksinasi Covid 19 Menggunakan Algortima Naïve Bayes Classifier. *Smart Comp: Jurnalnya Orang Pintar Komputer*, *11*(3), 438–447. https://doi.org/10.30591/smartcomp.v11i3.3893

Nisa, A., Darwiyanto, E., & Asror, I. (2019). Analisis Sentimen Menggunakan Naive Bayes Classifier dengan Chi-Square Feature Selection Terhadap Penyedia Layanan

Telekomunikasi. *e-Proceeding of Engineering* , *6*(2), 8650.

Pratama Putra, A., Pratama, Y., Kharisma Krisnadi, E., Purnamasari, I., & Dwi Saputra, D. (2022). Text Mining untuk Sentimen Analisis dengan Metode Naïve Bayes, SMOTE, N-Gram dan AdaBoost Pada Twitter CommuterLine. *Jurnal Sains Komputer & Informatika (J-SAKTI*, *6*(2), 961–973.

Raharjo, R. A., Sunarya, I. M. G., & Divayana, D. G. H. (2022). Perbandingan Metode Naïve Bayes Classifier Dan Support Vector Machine Pada Kasus Analisis Sentimen Terhadap Data Vaksin Covid-19 Di Twitter. *Elkom : Jurnal Elektronika dan Komputer*, *15*(2), 456–464. https://doi.org/10.51903/elkom.v15i2.918

Setiawan, H., Utami, E., & Sudarmawan, S. (2021). Analisis Sentimen Twitter Kuliah Online Pasca Covid-19 Menggunakan Algoritma Support Vector Machine dan Naive Bayes. *Jurnal Komtika (Komputasi dan Informatika)*, *5*(1), 43–51. https://doi.org/10.31603/komtika.v5i1.5189

Talita, A. S., Nataza, O. S., & Rustam, Z. (2021). Naïve Bayes Classifier and Particle Swarm Optimization Feature Selection Method for Classifying Intrusion Detection System Dataset. *Journal of Physics: Conference Series*, *1752*(1). https://doi.org/10.1088/1742-6596/1752/1/012021

Yuliyanti, S., & Sholihah, S. (2021). Pemodelan Analisis Sentimen Masyarakat terhadap Adaptasi Kebiasaan Baru (AKB) mengunakan Algoritma Naïve Bayes. *MIND Journal*, *6*(2), 155–167. https://doi.org/10.26760/mindjournal.v6i2.155-167