SISTEM INFORMASI MONITORING BENCANA ALAM DARI DATA MEDIA SOSIAL DENGAN ALGORITMA SUPPORT VECTOR MACHINE

Dimas Galih Setyo Pranowo
Fakultas Teknik Elektro
Telkom University
Bandung, Indonesia
dimasg@student.telkomuniversity.ac.id

Randy Erfa Saputra
Fakultas Teknik Elektro
Telkom University
Bandung, Indonesia
resaputra@telkomuniversity.ac.id

Casi Setianingsih
Fakultas Teknik Elektro
Telkom University
Bandung, Indonesia
setiacasie@telkomuniversity.ac.id

Abstract - Pertukaran informasi saat terjadinya bencana alam di media sosial khususnya twitter sudah menjadi kebiasaan pengguna di Indonesia terlebih saat situasi genting. Keadaan ini bisa dimanfaatkan untuk mengolah data tersebut menjadi informasi bencana alam yang relevan. Dalam hal ini tidak semua tweet yang berhubungan dengan bencana alam itu memiliki informasi yang valid, maka tujuan dari penelitian ini adalah untuk membuat sebuah sistem yang akan memetakan tweet bencana alam yang sedang terjadi secara otomatis berdasarkan lokasi dari pengguna tweet tersebut. Digunakan metode text mining untuk dapat memilah data yang memiliki informasi mengenai bencana alam yang sedang terjadi dari data tweet yang didapatkan secara real-time. Penelitian ini akan melakukan crawling data dari twitter untuk dianalisis menggunakan algoritma Support Vector Machine (SVM) sebagai classifier dan POS TF-IDF sebagai ekstraksi fitur, didapatkan nilai F1-score 83.56%, Precision 91.44%, Recall 85.42%, dan accuracy 91.5% dengan model SVM dengan parameter C = 0.7 dan γ(gamma) = 2.

Kata kunci – bencana alam, text mining, twitter

I. PENDAHULUAN

Situs *microblogging* twitter telah banyak digunakan oleh masyarakat Indonesia sebagai alat pertukaran informasi dan mengutarakan pendapat atau pemikiran, di Indonesia sendiri twitter adalah media sosial terpopuler sepanjang 2019 setelah facebook, whatsapp, youtube, instagram, dan fb messenger [1]. Pada tahun 2021 saja sudah terjadi total 763 bencana alam sepanjang tiga bulan pertama [2], tanpa kita sadari sosial media dalam hal ini twitter memberikan informasi terkait bencana alam secara tidak langsung. Karena sangat mudahnya diakses dari *website* maupun telepon pintar orang-orang sering kali berkomunikasi satu sama lain tentang pengalaman mereka terkait bencana alam melalui *retweets* [3], apalagi saat keadaan genting ini membuat twitter memiliki data yang sangat banyak dan *real-time*. Dalam hal ini pemanfaatan informasi di media sosial khususnya twitter akan berguna untuk pemetaan informasi bencana alam yang sedang terjadi, karena sosial media memungkinkan kita berinteraksi dengan skala yang luas.

Pada penelitian ini akan dibangun sistem monitoring bencana alam yang dengan sendirinya dapat mengklasifikasikan tweet dengan informasi bahwa pengguna sedang terdampak bencana alam dan akan dipetakan berdasarkan geolokasi dari masing-masing tweet. Dataset yang digunakan adalah tweet berbahasa indonesia yang diberi label secara manual, dan penelitian ini hanya dapat mengklasifikasikan tweet bencana alam banjir, gempa, dan longsor. Dalam penelitian juga digunakan algoritma s*upport vector machine* untuk dapat mengklasifikasikan data tweet dan POS TF-IDF sebagai ekstraksi fitur.

II. PENELITIAN TERKAIT

1. Twitter for Crisis Communication: Lesson Learned from Japan's Tsunami Disaster (Acar, Adam, Yuya Muraki, 2011)

   Pada penelitian ini dijelaskan apa yang terjadi dengan twitter pada saat tsunami melanda jepan melanda, didapatkan tweet pengguna di daerah Miyagi dan Kesennuma jepang setelah gempa bumi melanda yaitu pertama tweet "peringatan" banyak diposting saat hari terjadinya bencana, kedua tweet "permintaan bantuan" rata-rata adalah tweet permintaan bantuan untuk diri mereka sendiri maupun sebuah keluarga dengan menunjukan lokasi mereka, ketiga "melaporkan keadaan lingkungan sekitar" orang-orang secara terus menerus memposting tentang apa yang mereka rasakan dan apa yang terjadi disekitar lingkungan mereka. Dan juga didapatkan masalah mengenai sejauh mana tweet dari pengguna dapat dipercaya, banyak user yang menyebutkan bahwa mereka tidak dapat membedakan informasi yang benar dan tidak terlebih lagi saat situasi genting [4].
2. Practical Extraction of Disaster Relevant Information from Social Media (Imran, Muhammad, et al, 2013)

   Penelitian ini menjelaskan proses klasifikasi data tweet yang memiliki informasi seputar bencana alam. Penelitian ini menggunakan dataset *tornado* Joplin 2011 dan *hurricane* Sandy 2012, dan digunakan metode CRF untuk mengklasifikasikan tweet. Hasil dari penelitian yang dilakukan didapatkan model yang dapat mengklasifikasikan data tweet pada dataset *hurricane* Sandy dengan data latih

tornado Jopin dan mendapatkan performa yang cukup baik dengan kelas yang terbanyak yang berhasil diklasifikasikan adalah "caution and advice" dibandingkan kelas yang lain [5].

3. Text Mining on Real Time Twitter Data For Disaster Response (Myneni Madhu, Navya, Shruthilaya, 2017)

   Pada penelitian ini menggunakan metode *text analysis* dengan algoritma *machine learning* KNN, penelitian ini mencari data dengan metode pencarian *multilevel* agar data yang didapatkan tidak terlalu melebar dan lebih spesifik sebagai contoh mereka tidak menggunakan *keyword* "*#disaster"* melainkan mereka menggunakan *keyword* "*#Chennaifloods*" maka data yang didapatkan memiliki informasi yang lebih spesifik dan relevan [6].

4. The Real-Time Monitoring System of Social Big Data for Disaster Management (Choi, Seonhwa, Byunggul Bae, 2015)

   Penelitian ini menjelaskan aplikasi monitoring bencana yang terletak di Korea Selatan, kegunaan aplikasi ini adalah untuk memonitoring trend sosial untuk penanggulangan bencana. Aplikasi yang dibangun ini dapat mengklasifikasikan 58 jenis bencana termasuk bencana sosial maupun bencana alam dengan membandingkan *keyword* utama dari sebuah tweet dengan *predefined keyword* yang ditentukan oleh ahli bencana [7].

III. METODE PENELITIAN

A. Text Mining

Text mining adalah proses mengekstraksi sebuah data teks yang bertujuan untuk mendapatkan informasi atau pola dari data teks tersebut [8]. Pada penelitian ini proses pengumpulan data dilakukan di platform sosial media Twitter lalu data diproses menggunakan salah satu teknik yang ada pada *text mining* yaitu *text analysis*. Sebelum data diproses untuk dianalisis, terlebih dahulu data harus melewati pre-processing untuk menghilangkan data yang tidak diperlukan yang ada pada data teks [9]. Pada penelitian ini juga teknik yang digunakan untuk pre-processing data adalah *remove number and punctuation, word tokenize, stopword removal, lemmetization, part of speech tagging,* dan yang terakhir adalah pembobotan menggunakan *POS TF - IDF.*

B. POS TF-IDF

TF-IDF menghitung nilai kata di sebuah dokumen untuk mengetahui apakah kata tersebut penting atau tidak didalam dokumen tersebut. *Term Frequency* untuk mengukur seberapa sering sebuah kata muncul didalam sebuah dokumen, dan *Inverse Document frequency* mengukur apakah kata tersebut penting atau tidak untuk dokumen tersebut.

Part of speech (POS) adalah klasifikasi kata yang dikategorikan berdasarkan peran dan fungsinya dalam struktur bahasa inggris [10]. Berdasarkan struktur kalimat bahasa inggris *verb* dan *nouns* lebih penting didalam sebuah kalimat dibandingkan *adjectives* dan *adverbs,* maka dikembangkan algoritma *TF-IDF* menjadi *POS TF-IDF* dimana ada penambahan bobot pada sebuah kata *t* berdasarkan *part of speech* [11]. Berikut rumus POS TF-IDF:

t = kata
d = dokumen
N = total dokumen
k = total dokumen yang memiliki kata t didalamnya
C(t, d) = menghitung banyak kata t pada dokumen d
W_{pos} = bobot *tag part of speech*

$$\begin{cases} \frac{C(t,d)^{*}Wpos(t)}{\sum C(t_i,d)^{*}Wpos(t_i)} * \left(1+\log\left(\frac{N}{k}\right)\right) & if\ c(t,d)\ \geq 1 \\ 0 & otherwise \end{cases} \quad (1)$$

$$\text{Wpos} = \begin{cases} \text{W1} & if\ t\ is\ a\ verb\ or\ noun \\ \text{W2} & \text{if t is a adverb } or\ adjective \\ \text{W3} & \text{otherwise} \end{cases} \quad (2)$$

Dimana $W_1 > W_2 > W_3 > 0$ jadi setiap kata diberi bobot tambahan sesuai dengan *part of speech tag* yang berarti berdasarkan rumus (2) kata yang memiliki *tag verb* atau *noun* lebih penting dibandingkan kata yang memiliki *tag adverb* atau *adjective* [11].

C. Support Vector Machine

Support Vector Machine adalah algoritma yang memisahkan kelas dengan pemilihan *threshold* yang dibantu oleh *margin* dengan menghitung jarak antara masing-masing *class* untuk mendapatkan *threshold* yang terbaik, yang disebut *hyperplane* [12]. *Support vector machine* memiliki cara pada saat data tidak dapat dipisahkan dengan garis lurus, maka digunakan kernel trik seperti *polynomial kernel, radial basis,* dan *sigmoid*. Berikut rumus dari *support vector machine*:

$$\begin{aligned} &w^T.x_i + b \geq 1, \\ &w^T.x_i + b = 0, \\ &w^T.x_i + b \leq -1 \end{aligned} \quad (3)$$

Rumus (3) adalah rumus untuk membentuk hyperplane yang akan memisahkan data dimana $\omega^T.x_i + b = 0$ sebagai decision boundary, w adalah weight vector dan b adalah bias [13].

$$\begin{aligned} &minJ(w,b,\xi_i) = \frac{1}{2}w^T w + C\sum_{i=1}^{l}\xi_i \\ &\qquad subject\ to: \\ &y_i(\omega^T\,\varphi(x_i) + b) \geq 1 - \xi_i, \qquad i = 1,\dots,l \\ &\xi_i \geq 0, i = 1,\dots,l \end{aligned} \quad (4)$$

Rumus (4) adalah rumus teknik soft margin untuk memperbolehkan *misclassification* dengan syarat diberikan penalti, dimana w adalah vektor dari hyperplane, b adalah bias, $\varphi(x_i)$ adalah fungsi *non linear* yang merubah **x** menjadi *high dimensional feature space,* selanjutnya *C* adalah regularization konstan yang mengatur margin klasifikasi dan *cost misclassification* lalu ξ adalah nilai eror [13].

$$\begin{aligned} &max_\alpha \sum_{i=1}^{l}\alpha_i - \frac{1}{2}\sum_{i,j=1}^{l}\alpha_i\alpha_j y_i y_j K(x_i,x_j) \\ &subject\ to: \\ &\sum_{i=1}^{l}\alpha_i y_i = 0 \\ &0 \leq \alpha_i \leq C \end{aligned} \quad (5)$$

Rumus (2.5) adalah hasil dari proses optimasi dual expression dimana α adalah Lagrange multipliers dan $K(x_i, x_j)$ adalah fungsi kernel [13].

$$y(x) = \begin{cases} +1, \sum_{i=1}^{l} \alpha_i y_i K(x, x_i) + b \geq 0 \\ -1, \sum_{i=1}^{l} \alpha_i y_i K(x, x_i) + b \leq 0 \end{cases} \quad (6)$$

Rumus (6) diselesaikan dengan *sequential minimal optimization* yaitu untuk mencari nilai *alpha* hingga mendapatkan *margin width* terbesar. Jika SVM model sudah terselesaikan data *testing* dapat diprediksi dengan rumus (3) [13].

IV. PERANCANGAN SISTEM

A. Desain Sistem

Desain sistem adalah alur bagaimana proses sistem dari pengolahan data hingga sistem belajar mengklasifikasikan data. alur dari sistem klasifikasi yang dilakukan, pertama adalah pelabelan dataset yang dilakukan secara manual, setelah data memiliki label lalu data akan melewati proses *preprocessing*, setelah itu data akan diberikan *tag part of speech* untuk masing-masing kata pada dataset, selanjutnya data akan diberi pembobotan pada proses *POS TF*IDF*, setelah data diberi bobot maka data siap untuk dimasukan ke dalam algoritma SVM tetapi sebelum itu data akan dibagi menjadi 2 yaitu data latih dan data testing yang bertujuan untuk mengukur akurasi dari model SVM nanti, setelah itu data *training* digunakan untuk melatih model SVM, setelah didapatkan model SVM maka model akan memprediksi data *testing*, lalu hasil prediksi akan diuji akurasinya diproses evaluasi.

B. Pengumpulan Data

Proses *crawling* data dari twitter menggunakan library Twitter Intelligence Tool (*TWINT)* yang ada di bahasa pemrograman *python.* Data yang digunakan untuk *training* adalah data *tweet* sejak 2013 hingga 2021 dengan *keyword* banjir, gempa, longsor. Setelah diseleksi didapatkan 3841 *tweet* yang dilabelkan menjadi 4 kelas yaitu banjir, gempa, longsor, dan lainnya. Berikut adalah contoh bentuk dataset:

Tabel 1 contoh dataset tweet

Tweet	Label
parah depan rumah gw banjir banget gengs #banjir #manapemerintah https://t.co/3k8PYt1WuW?amp=1	banjir
parah td gempa lg ngampus d lt 12	gempa
rumahku sudah rata dgn tanah, di tunggoro dan jalur ke dieng jga longsor	longsor
NUSANTARA : Longsor Landa Tujuh Desa di Purworejo warga diminta untuk tetap waspada http://bit.ly/b72wK8	lainnya

Pada tabel 1 adalah hasil *tweet* yang diambil dari twitter menggunakan *library* python, *tweet* tersebut adalah contoh beberapa data yang dicari menggunakan *keyword* banjir, gempa, dan longsor. Pelabelan dataset dilakukan secara manual, dataset dibagi menjadi 4 kelas label yang pertama banjir dengan ketentuan pelabelan data untuk kelas banjir, gempa, dan longsor adalah *tweet* pengguna yang memiliki informasi bahwa pengguna sedang terdampak salah satu dari ketiga bencana tersebut pada saat itu, untuk kelas lainnya adalah *tweet* berita atau opini masyarakat terhadap bencana alam banjir, gempa, dan longsor yang sedang terjadi dan pengguna tersebut tidak terdampak bencana alam tersebut.

C. *Preprocessing Data*

Pada proses ini data akan dibersihkan dan dihilangkan kata, angka, link, maupun tanda baca yang tidak dibutuhkan untuk proses klasifikasi nanti. Pertama dataset dibersihkan dengan menghilangkan angka, tanda baca yang tidak digunakan dalam penelitian ini, lalu data akan melewati proses *lemmatization* untuk mengembalikan sebuah kata menjadi kata dasar kata itu sendiri, setelah itu kata-kata yang tidak merubah arti dari keseluruhan kalimat akan dihilangkan dengan proses *remove stopwords*, yang terakhir adalah proses *word tokenization*.

D. *POS Tagging*

Pada proses ini data dalam hal ini kata dalam sebuah kalimat akan dikategorikan berdasarkan peran dan fungsinya. Digunakan *Pre-trained POS-Tagger* dataset untuk dapat diimplementasikan ke bahasa Indonesia.

Tabel 2 data yang telah melewati proses POS Tagging

Input	Output
juang tembus banjir macet	[(‘juang’, ‘NN’), (‘tembus’, ‘VB’), (‘banjir’, ‘NN’), (‘macet’, ‘JJ’)]
banjir betis saat santai bolos ngantor	[(‘banjir’, ‘NN’), (‘betis’, ‘JJ’), (‘saat’, ‘NN’), (‘santai’, ‘NN’), (‘bolos’, ‘NN’), (‘ngantor’, ‘NN’)]
parah td gempa ngampus d lt	[(‘parah’, ‘JJ’), (‘td’, ‘FW’), (‘gempa’, ‘NN’), (‘ngampus’, ‘VB’), (‘d’, ‘NNP’), (‘lt’, ‘NN’)]

Pada tabel 2 adalah proses pemberian *part of speech tag* kepada masing-masing kata dalam sebuah kalimat, proses ini bertujuan untuk mengetahui seberapa penting sebuah kata dibandingkan dengan kata lainnya dalam sebuah kalimat.

E. Pembobotan

Pada proses ini adalah untuk merubah dataset menjadi angka yang dihitung berdasarkan rumus dari *POS-TF*IDF* yang akan memberikan masing-masing kata bobot agar bisa lanjutkan ke proses klasifikasi. Berikut contoh perhitungan dari *POS TF*IDF* yang merujuk pada rumus (1):

Tabel 3 perhitungan frekuensi kata

POS TF			
Kata	D1	D2	D3
juangNN	0.278	0	0
tembusVB	0.278	0	0
banjirNN	0.278	0.167	0
macetJJ	0.167	0	0
betisJJ	0	0.167	0
saatNN	0	0.167	0
santaiNN	0	0.167	0
bolosNN	0	0.167	0
ngantorNN	0	0.167	0
parahJJ	0	0	0.227
tdFW	0	0	0.045
gempaNN	0	0	0.227
ngampusVB	0	0	0.227
dNNP	0	0	0.045
ltNN	0	0	0.227

Tabel 3 adalah hasil perhitungan POS-Term Frequency yang bertujuan untuk mendapatkan nilai frekuensi masing-masing kata. Kata yang memiliki POS-Tag “noun” (NN) dan “verb” (VB) akan dikalikan dengan 5, untuk kata yang memiliki POS-Tag “adjective” (JJ) dan “adverb” (RB) akan dikalikan dengan 3.

Tabel 4 perhitungan IDF

Kata	IDF
juangNN	1.477
tembusVB	1.477
banjirNN	1
macetJJ	1.477
betisJJ	1.477
saatNN	1.477
santaiNN	1.477
bolosNN	1.477
ngantorNN	1.477
parahJJ	1.477
tdFW	1.477
gempaNN	1.477
ngampusVB	1.477
dNNP	1.477
ltNN	1.477

Tabel 4 adalah hasil perhitungan Inverse document frequency yang bertujuan untuk mengetahui seberapa sering kata tersebut muncul di seluruh dokumen.

*Tabel 5 perhitungan POS TF * IDF*

POS TF * IDF			
Kata	D1	D2	D3
juangNN	0.4106	0	0
tembusVB	0.4106	0	0
banjirNN	0.278	0.167	0
macetJJ	0.4106	0	0
betisJJ	0	0.2467	0
saatNN	0	0.2467	0
bolosNN	0	0.2467	0
ngantorNN	0	0.2467	0
parahJJ	0	0	0.3353
tdFW	0	0	0.0665
gempaNN	0	0	0.3353
ngampusVB	0	0	0.3353

POS TF * IDF			
Kata	D1	D2	D3
dNNP	0	0	0.0665
ltNN	0	0	0.3353

Tabel 5 adalah hasil perhitungan *POS-Tf * IDF* yang mana hasil dari dataset yang telah diberi pembobotan dan sudah siap untuk dilakukan proses latih dengan SVM.

F. Klasifikasi

Setelah data selesai dihitung di proses pembobotan maka didapatkan *vector* yang sudah siap untuk di klasifikasikan mengunakan *multiclass Support Vector Machine* dengan metode *One vs Rest* dan parameter $C = 0.7$, *kernel = RBF* , *gamma = 2*.

G. Evaluasi Performansi

Proses evaluasi dengan *confusion matrix* untuk mengetahui nilai *precision, recall, dan f1-score.* Data dibagi menggunakan proses *k-fold cross validation* untuk mendapatkan hasil yang lebih akurat, pada proses ini digunakan nilai k=8. Berikut tabel *confusion matrix*:

Tabel 6 nilai pengujian confusion matrix untuk masing-masing kelas

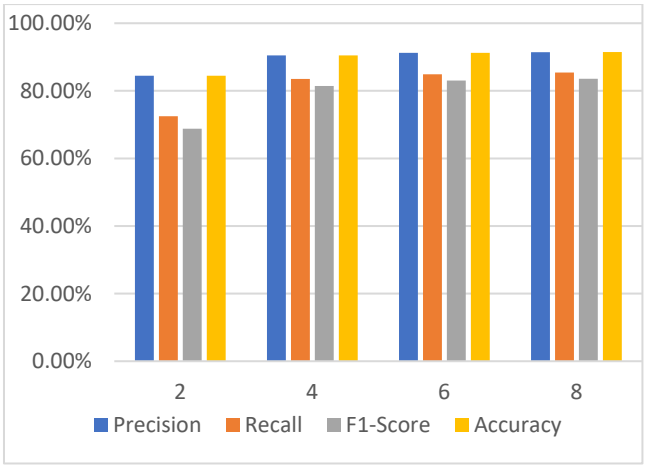
Kelas	Precision	Recall	F1-score	Accuracy	K
banjir	80.0%	97.0%	88.0%	93.0%	2
gempa	67.36%	75.96%	71.41%	87.34%	
longsor	56.38%	55.51%	55.94%	80.80%	
lainnya	37.40%	28.61%	32.42%	61.79%	
banjir	83.17%	98.0%	90.0%	94.50%	4
gempa	85.97%	95.11%	90.31%	95.76%	
longsor	66.95%	85.52%	75.10%	87.55%	
lainnya	76.28%	45.77%	57.21%	78.07%	
banjir	85.44%	98.24%	91.39%	95.33%	6
gempa	89.0%	96.24%	92.48%	96.74%	
longsor	68.36%	90.98%	78.1%	88.77%	
lainnya	83.94%	50.56%	63.11%	81.06%	
banjir	85.53%	98.34%	91.49%	95.39%	8
gempa	88.40%	96.37%	92.21%	96.61%	
longsor	68.92%	92.88%	79.13%	89.24%	
lainnya	86.11%	50.40%	63.58%	81.51%	

Hasil dari tabel 6 dihitung menggunakan library python Pycm [14] dengan nilai *TP, TN, FP, FN* didapat dari hasil rata-rata dari kelas itu sendiri dengan menggunakan teknik pembagian data *k-fold cross validation.*

Tabel 7 total nilai keseluruhan kelas confussion matrix

Precision	Recall	F1-Score	Accuracy	K
84.45%	72.49%	68.77%	84.45%	2
90.46%	83.51%	81.46%	90.47%	4
91.22%	84.91%	83.03%	91.22%	6
91.44%	85.42%	83.56%	91.5%	8

Nilai total dari *precision, recall, f1-score, accuracy* diatas didapatkan dengan cara menjumlahkan nilai dari masing-masing kelas lalu dibagi dengan banyaknya kelas.



Gambar 1 grafik pengujian confusion matrix dengan cross validation

Pada gambar 1 adalah grafik perbandingan *confusion matrix* dengan pembagian data menggunakan teknik *cross validation* dengan nilai k adalah 2, 4, 6, dan 8, didapatkan performa terbaik adalah nilai *k=8* dengan precision=91.44%, recall=85.42%, f1-score=83.56%, dan akurasi=91.5%.

V. IMPLEMENTASI DAN PENGUJIAN

A. Dataset

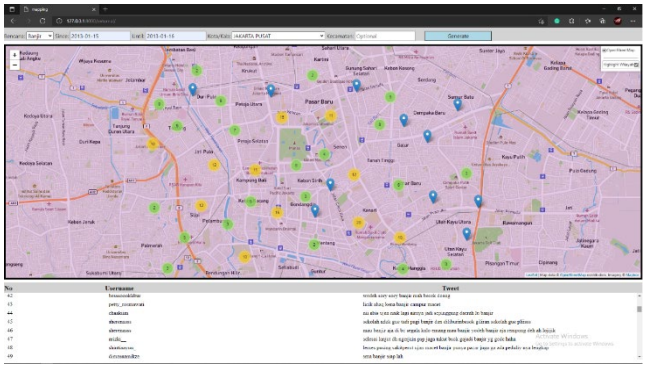
Dataset yang digunakan berjumlah 3841 dengan proporsi kelas banjir sebanyak 968 data, kelas gempa sebanyak 799 data, kelas longsor sebanyak 843 data, dan kelas lainnya sebanyak 1230 data.

B. Implementasi Desain Antarmuka

Implementasi desain antar muka dari sistem informasi monitoring bencana alam dari data media sosial pada gambar berikut:

Gambar 2 halaman depan web aplikasi

Pada gambar 2 ini pengguna diwajibkan untuk menginput parameter pada web aplikasi, yang pertama pengguna harus memilih 1 dari 3 pilihan bencana alam, setelah itu pengguna memasukan rentang tanggal twett yang diinginkan, lalu pengguna harus memilih kota/kabupaten lokasi tweet berada, untuk form kecamatan bersifat optional jika pengguna memasukan nama kecamatan maka web akan mengambil data tweet dari wilayah kecamatan tersebut, jika pengguna tidak memasukan nama kecamatan pada form tersebut maka web akan mengambil data tweet dari wilayah kota/kab yang dipilih.

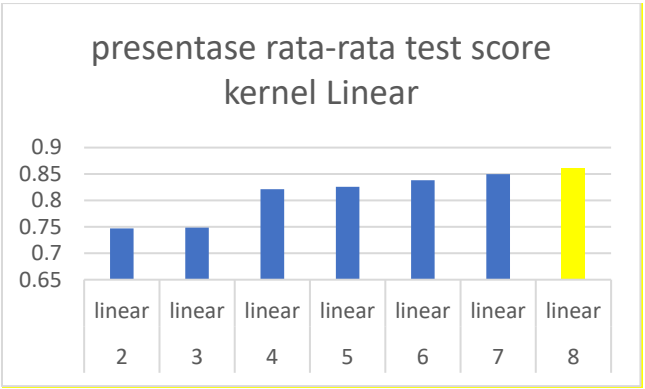


Gambar 3 halaman web hasil pemetaan tweet

Pada gambar 3 hasil pemetaan *tweet* adalah hasil pemetaan *tweet* berdasarkan geolokasi masing-masing *tweet* di wilayah yang sudah diinputkan oleh pengguna jika pengguna menyalakan geolokasi mereka maka web aplikasi akan memetakan sesuai geolokasi pengguna tersebut, jika pengguna *hide* atau menyembunyikan geolokasi mereka maka *tweet* dipetakan berdasarkan titik wilayah yang dipilih. *Tweet* hasil klasifikasi juga ditampilkan pada tabel dibawah map.

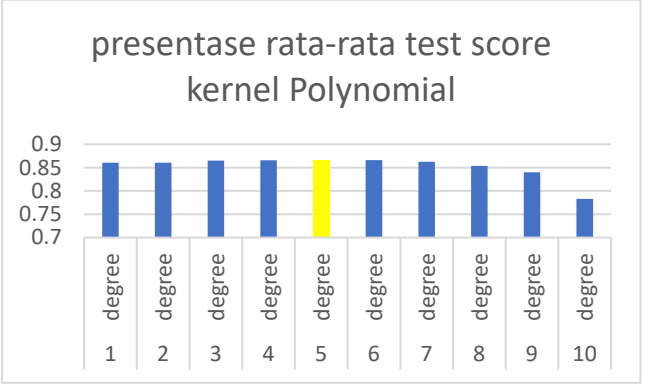
C. Pengujian Sistem Klasifikasi

Pada skenario pengujian model klasifikasi dilakukan perbandingan dengan 3 jenis kernel yaitu *linear, polynomial, rbf* dan 3 jenis parameter yaitu *C, gamma,* dan *degree*. Hasil akurasi didapatkan dari hasil rata-rata menggunakan metode *k-fold cross validation* dengan nilai *k=8*.



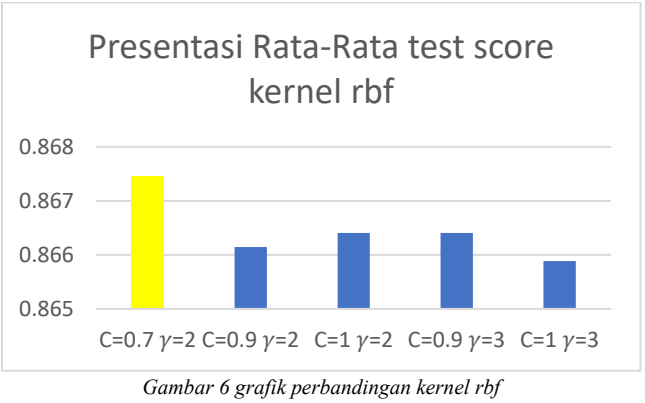
Gambar 4 grafik perbandingan kernel linear

Hasil proses pengujian yang ditunjukan pada gambar 4 grafik perbandingan diatas adalah model svm menggunakan kernel *linear*, karena kernel *linear* tidak memiliki parameter jadi pengujian berdasarkan distribusi data dengan *k-fold cross validation* dengan nilai k=2, 3, 4, 5, 6, 7, dan 8. hasil dari pengujian kernel linear dengan mean test score terbaik adalah 86.041% di nilai k=8 maka dapat disimpulkan semakin besar data latih maka akurasi akan semakin besar.



Gambar 5 grafik perbandingan kernel polynomial

Hasil proses pengujian yang ditunjukan pada gambar 5 grafik perbandingan kernel *polynomial* dengan pengujian berdasarkan nilai parameter d (*degree)*, untuk distribusi data pada pengujian ini digunakan *cross validation* dengan nilai *k*=8. didapatkan rata-rata nilai test score terbesar untuk parameter d(*degree) = 5* dengan *mean test score* 86.64% untuk kernel *polynomial*.



Gambar 6 grafik perbandingan kernel rbf

Pada gambar 6 adalah grafik perbandingan parameter kernel *rbf* diambil 5 akurasi terbaik dan didapatkan parameter terbaik untuk kernel *rbf* adalah C=0.7 dan $\gamma = 2$ dengan *mean test score* 86.67%, dari perbandingan diatas dapat ditarik kesimpulan untuk parameter C didapatkan rentang terbaik dari 0.7 hingga 1 untuk parameter γ didapatkan rentang terbaik adalah dari 2 hingga 3.

VI. KESIMPULAN

Dengan demikian dapat ditarik kesimpulan dari ketiga kernel yang diuji didapatkan model svm dengan terbaik adalah kernel *rbf* dengan parameter *C*=0.7 dan parameter $\gamma = 2$ dengan pembagian data dilakukan menggunakan *k-fold cross validation* didapatkan akurasi terbaik dengan nilai *k=8* dan didapatkan *F1-score* 83.56%, Precision 91.44%, Recall 85.42%, dan *accuracy* 91.5%.

VII. REFERENSI

[1] global web index, "Global Web Index," 2019. [Online]. Available: https://www.globalwebindex.com/hubfs/Downloads/Indonesia_Market_Snapshot.pdf. [Diakses 10 Mei 2021].

[2] BNPB, "Badan Nasional Penanggulangan Bencana," [Online]. Available: http://www.bnpb.go.id. [Diakses 15 July 2021].

[3] K. L. P. Starbird, "Pass it on?: Retweeting in mass emergency," *ISCRAM,* 2010.

[4] A. Y. M. Acar, "Twitter for crisis communication: lessons learned from Japan's tsunami disaster," *International journal of web based communities,* pp. 392-402, 2011.

[5] M. Imran, "Practical extraction of disaster-relevant information from social media," *Proceedings of the 22nd international conference on world wide web,* 2013.

[6] M. M. K. N. P. S. Bala, "Text mining on real time Twitter data for disaster response," *Int. J. Civ. Eng. Technol,* pp. 20-29, 2017.

[7] S. B. B. Choi, "The real-time monitoring system of social big data for disaster management," dalam *Computer science and its applications*, Berlin, 2015.

[8] A.-H. Tan, "Text mining: The state of the art and the challenges.," dalam *Proceedings of the pakdd 1999 workshop on knowledge disocovery from advanced databases.*, 1999.

[9] A. A. N. G. P. Hotho, "A brief survey of text mining," *Ldv Forum,* vol. 20, no. 1, 2005.

[10] A. Dinakaramani, "Designing an Indonesian part of speech tagset and manually tagged Indonesian corpus," dalam *2014 International Conference on Asian Language Processing (IALP)*, 2014.

[11] R. Xu, "POS weighted TF-IDF algorithm and its application for an MOOC search engine," dalam *2014 International Conference on Audio, Language and Image Processing*, 2014.

[12] W. S. Noble, "What is a support vector machine?," *Nature biotechnology,* pp. 1565-1567, 2006.

[13] X. Yang, "The one-against-all partition based binary tree support vector machine algorithms for multi-class classification," *Neurocomputing ,* pp. 1-7, 2013.

[14] J. M. H. S. Z. A. Haghighi S, "PyCM: Multiclass confussion matrix library in python," *Journal of Open Source Software,* vol. 3, no. 25, p. 729, 2018.