TESIS

TEXT MINING PADA SOSIAL MEDIA FACEBOOK UNTUK MENDETEKSI EMOSI PENGGUNA

Disusun oleh:

Nama : **Riska Dwi Handayani**
NIM : **18.52.1116**
Konsentrasi : **Business Intelligence**

PROGRAM STUDI S2 TEKNIK INFORMATIKA
PROGRAM PASCASARJANA UNIVERSITAS AMIKOM YOGYAKARTA
YOGYAKARTA
2020

TESIS

TEXT MINING PADA SOSIAL MEDIA FACEBOOK UNTUK MENDETEKSI EMOSI PENGGUNA

TEXT MINING ON SOCIAL MEDIA FACEBOOK TO DETECT USER EMOTIONS

Diajukan untuk memenuhi salah satu syarat memperoleh derajat Magister

Disusun oleh:

Nama : Riska Dwi Handayani
NIM : 18.52.1116
Konsentrasi : Business Intelligence

PROGRAM STUDI S2 TEKNIK INFORMATIKA
PROGRAM PASCASARJANA UNIVERSITAS AMIKOM YOGYAKARTA
YOGYAKARTA
2020

HALAMAN PENGESAHAN

TEXT MINING PADA SOSIAL MEDIA FACEBOOK UNTUK MENDETEKSI EMOSI PENGGUNA

TEXT MINING ON SOCIAL MEDIA FACEBOOK TO DETECT USER EMOTIONS

Dipersiapkan dan Disusun oleh

Riska Dwi Handayani

18.52.1116

Telah Diujikan dan Dipertahankan dalam Sidang Ujian Tesis
Program Studi S2 Teknik Informatika
Program Pascasarjana Universitas AMIKOM Yogyakarta
pada hari Senin, 02 November 2020

Tesis ini telah diterima sebagai salah satu persyaratan
untuk memperoleh gelar Magister Komputer

Yogyakarta, 02 November 2002
Rektor

Prof. Dr. M. Suyanto, M.M.
NIK. 190302001

HALAMAN PERSETUJUAN

TEXT MINING PADA SOCIAL MEDIA FACEBOOK UNTUK MENDETEKSI EMOSI PENGGUNA

TEXT MINING ON SOCIAL MEDIA FACEBOOK TO DETECT USER EMOTIONS

Dipersiapkan dan Disusun oleh

Riska Dwi Handayani

18.52.1116

Telah Diujikan dan Dipertahankan dalam Sidang Ujian Tesis
Program Studi S2 Teknik Informatika
Program Pascasarjana Universitas AMIKOM Yogyakarta
pada hari Senin, 02 November 2020

Pembimbing Utama	**Anggota Tim Penguji**
Dr. Kusrini, M.Kom.	**Prof. Dr. Ema Utami, S.Si., M.Kom**
NIK. 190302106	**NIK. 190302037**
	Dr. Wing Wahyu Winarno, MAFIS, Ak
Pembimbing Pendamping	**NIK. 555195**
Hanif Al Fatta, M.Kom	**Dr. Kusrini, M.Kom**
NIK. 190302096	**NIK. 190302106**

Tesis ini telah diterima sebagai salah satu persyaratan
untuk memperoleh gelar Magister Komputer

Yogyakarta, 02 November 2020
Direktur Program Pascasarjana

Dr. Kusrini, M.Kom.
NIK. 190302106

HALAMAN PERNYATAAN KEASLIAN TESIS

Yang bertandatangan di bawah ini,

Nama mahasiswa : Riska Dwi Handayani
NIM : 18.52.1116
Konsentrasi : Business Intelligence

Menyatakan bahwa Tesis dengan judul berikut:
Text Mining Pada Social Media Facebook untuk Mendeteksi Emosi Pengguna

Dosen Pembimbing Utama : Dr.Kusrini, M.Kom.
Dosen Pembimbing Pendamping : Hanif Al Fatta, M.Kom

1. Karya tulis ini adalah benar-benar ASLI dan BELUM PERNAH diajukan untuk mendapatkan gelar akademik, baik di Universitas AMIKOM Yogyakarta maupun di Perguruan Tinggi lainnya
2. Karya tulis ini merupakan gagasan, rumusan dan penelitian SAYA sendiri, tanpa bantuan pihak lain kecuali arahan dari Tim Dosen Pembimbing
3. Dalam karya tulis ini tidak terdapat karya atau pendapat orang lain, kecuali secara tertulis dengan jelas dicantumkan sebagai acuan dalam naskah dengan disebutkan nama pengarang dan disebutkan dalam Daftar Pustaka pada karya tulis ini
4. Perangkat lunak yang digunakan dalam penelitian ini sepenuhnya menjadi tanggung jawab SAYA, bukan tanggung jawab Universitas AMIKOM Yogyakarta
5. Pernyataan ini SAYA buat dengan sesungguhnya, apabila di kemudian hari terdapat penyimpangan dan ketidakbenaran dalam pernyataan ini, maka SAYA bersedia menerima SANKSI AKADEMIK dengan pencabutan gelar yang sudah diperoleh, serta sanksi lainnya sesuai dengan norma yang berlaku di Perguruan Tinggi

Yogyakarta, 02 November 2020
Yang Menyatakan,

Riska Dwi Handayani

V

HALAMAN PERSEMBAHAN

Puji syukur penulis panjatkan kepada Allah SWT atas anugerah dan nikmat yang yang tidak terkira. Pada kesempatan ini penulis ingin menyampaikan ucapan terimakasih yang sebesar-besarnya kepada semua pihak yang telah membantu dalam penyusunan laporan tesis ini. Penulis mengucapkan terimakasih kepada:

1. Ibu Dr. Kusrini, M.Kom, selaku pembimbing utama yang selalu memberikan bimbingan, motivasi dan arahan dalam proses pengerjaan tesis.
2. Bapak Hanif Al Fatta, M.Kom, selaku pembimbing pendamping yang telah memberikan bimbingan dan masukan-masukan dalam proses pengerjaan tesis.
3. Segenap keluarga yang senantiasa memberikan perhatian dan dukungan.
4. Calon suami yang telah memberikan semangat, dukungan serta bantuannya.
5. Rekan-rekan MTI serta rekan-rekan kerja di KB & TK ANAKQU yang telah memberikan segala macam bentuk bantuan dan do'a.

Penulis sadar bahwa karya tulis ini belum sempurna, namun semoga dapat memberi manfaat. Diharapkan hasil karya tulis ini juga dapat memberikan manfaat untuk penelitian selanjutnya.

HALAMAN MOTTO

Hidup berawal dari mimpi

Man Jadda Wa Jadda

KATA PENGANTAR

Alhamdulillah, puji syukur kehadirat Allah SWT yang telah melimpahkan rahmat dan karunia-Nya sehingga penulis dapat menyelesaikan laporan tesis ini dengan baik. Meskipun dalam penyelesaian ini banyak ditemui kendala. Tesis ini disusun guna memenuhi salah satu syarat untuk menyelesaikan Program Studi S2 Teknik Informatika di Universitas Amikom Yogyakarta.

Berbagai pihak telah banyak membantu penulis dalam penyelesaian tesis ini, untuk itu penulis ucapkan terima kasih kepada :

1. Bapak KH Ahmad Izzudin, Lc., M.Si. dan Bapak KH Nashrul Arief yang telah memberikan ijin belajar bagi penulis.
2. Ibu Dr. Kusrini, M.Kom, selaku pembimbing utama yang selalu memberikan bimbingan, motivasi dan arahan dalam proses pengerjaan tesis.
3. Bapak Hanif Al Fatta, M.Kom, selaku pembimbing pendamping yang telah memberikan bimbingan dan masukan-masukan dalam proses pengerjaan tesis.
4. Segenap keluarga yang senantiasa memberikan perhatian dan dukungan
5. Calon suami yang telah memberikan semangat, dukungan serta bantuannya.
6. Rekan-rekan MTI serta rekan-rekan kerja di KB & TK ANAKQU yang telah memberikan segala macam bentuk bantuan dan do'a.
7. Pihak-pihak lain yang tidak dapat penulis sebutkan satu per satu.

Penulis menyadari bahwa penulisan tesis ini masih banyak kekurangan, oleh karenanya kritik dan saran sangat penulis harapkan guna menyempurnakan

tesis ini.Akhir kata penulis mengucapkan banyak terima kasih dan semoga tesis inibisa memberi manfaat bagi kita semua.

Yogyakarta, 02 November 2020

Penulis

DAFTAR ISI

DAFTAR TABEL

DAFTAR GAMBAR

INTISARI

Media sosial merupakan salah satu dari bermacam-macam media komunikasi. Selain sebagai media komunikasi media social juga menjadi tempat yang memungkinkan orang untuk berbagi apa yang terjadi dalam hidup mereka. Konten yang dibagikan itu tidak hanya berisi suatu informasi yang terjadi akan tetapi juga berisi tentang perilaku pengguna facebook itu sendiri maupun emosi dari pengguna facebook tersebut.

Berdasarkan hal tersebut peneliti mencoba menggunakan data dari update status facebook untuk mendeteksi emosi pengguna. Dalam melakukan penelitian ini peneliti membandingkan 4 ekstraksi fitur yaitu Count Vector, TF-IDF Character Level, TF-IDF Word Level dan TF-IDF Ngram Level. Selain membandingkan 4 ekstraksi fitur , peneliti juga membandingkan 3 metode klasifikasi yang digunakan yaitu Naïve Bayes, K-Nearest Neighbour dan Suport Vector Machine.

Hasil dari penelitian ini adalah pasangan dari ekstraksi fitur dan metode penelitian apa yang menghasilkan nilai akurasi tertinggi. Pada penelitian pasangan ekstraksi fitur dan metode penelitian yang menghasilkan nilai akurasi tertinggi adalah ektraksi fitur count vector dan metode klasifikasi naïve bayes dengan nilai 60%.

Kata Kunci : Count vector, TF-IDF, Naïve Bayes, K-Nearest Neighbour, Suport Vector Machine

ABSTRACT

Social media is one of a variety of communication media. Apart from being a medium of communication, social media is also a place that allows people to share what is happening in their lives. The content that is shared does not only contain information that occurs but also contains the behavior of the Facebook user itself and the emotions of the Facebook user.

Based on this, researchers tried to use data from Facebook status updates to detect user emotions. In conducting this research, the researcher compared 4 feature extractions, namely Count Vector, TF-IDF Character Level, TF-IDF Word Level and TF-IDF Ngram Level. Apart from comparing the 4 feature extractions, the researcher also compared the 3 classification methods used, namely Naïve Bayes, K-Nearest Neighbor and Suport Vector Machine.

The results of this study are a pair of feature extraction and research methods that produce the highest accuracy value. In the paired study of feature extraction and the research method that produced the highest accuracy value was the count vector feature extraction and the naïve Bayes classification method with a value of 60%.

Keywords: Count vector, TF-IDF, Naïve Bayes, K-Nearest Neighbor, Suport Vector Machine

BAB I

PENDAHULUAN

1.1. Latar Belakang Masalah

Media sosial merupakan salah satu dari bermacam-macam media komunikasi. Media sosial digunakan sebagai wadah untuk bersosialisai (berhubungan baik secara personal, kelompok, dan lain sebagainya) diantara para penggunanya (Asiati & Septadiyanto, 2019). Indonesia merupakan salah satu negara dengan jumlah pengguna sosial yang terbesar di dunia. Pengguna facebook, twitter, instagram dan lan-lain di Indonesia memiliki jumlah pengguna yang besar dari pengguna media sosial tersebut secara keseluruhan.Situs jejaring sosial yang paling banyak diakses adalah Facebook dan Twitter. Indonesia menempati peringkat 4 pengguna Facebook terbesar setelah USA, Brazil, dan India (Kominfo, 2013).

Facebook adalah situs jejaring sosial yang memungkinkan orang untuk berbagi apa yang terjadi dalam hidup mereka. Mereka dapat berbagi foto, video, teks dan tautan ke situs web dengan teman-teman mereka (Nedwidek, 2010). Konten yang dibagikan pengguna facebook tersebut tidak hanya berisi suatu informasi yang terjadi akan tetapi juga berisi tentang perilaku pengguna facebook itu sendiri maupun emosi dari pengguna facebook tersebut (Ardiada et al., 2019).

Emosi adalah komponen integral dari semua aktivitas manusia termasuk interaksi manusia – komputer . Kemampuan mengolah emosi tersebut penting dalam suatu sistem komputer. Sehingga mengenali emosi pengguna dapat

meningkatkan kualitas interaksi antar kedua komponen tersebut (Lopatovska & Arapakis, 2011). Deteksi emosi adalah salah satu hal penting dalam pengolahan emosi dengan komputer. Sebelum komputer dapat mengenali emosi dari penggunanya maka perlu dilakukan proses deteksi emosi terlebih dahulu. Deteksi emosi dapat dilakukan dengan suara, ekspresi wajah, gestur, teks (Calvo & D'Mello, 2010). Dalam kasus pengguna facebook ini yang paling mungkin bisa kita lakukan adalah mendeteksi emosi dengan teks, dikarenakan tidak adanya tatap muka dan tidak mendengar langsung bagaimana suara dari pengguna tersebut.

Untuk mendeteksi emosi dengan teks dari layanan sosial media seperti facebook yang datanya tidak terstruktur maka perlu dilakukan analisis teks , salah satunya dengan Text Mining (Ardiada et al., 2019). Menurut hasil survey yang telah dilakukan oleh (Dandannavar et al., 2018), terdapat beberapa metode yang dapat digunakan untuk mengklasifikasikan kepribadian seseorang melalui media social. Beberapa metode tersebut adalah *Suport Vector Machine, K-Nearest Neighbour, Multinominal Naïve Bayes, Naïve Bayes, Multi Task Regression, Algoritma Incremental Regression.*

Beberapa penelitian mengenai deteksi emosi yang telah dilakukan lebih dulu. Yang pertama adalah penelitian yang dilakukan oleh (Rohman et al. 2019) tentang deteksi tingkat emosi pada media sosial dengan menggunakan pendekatan *leksikon* dan *natural language processing*. Hasil penelitian ini menunjukkan bahwa perbaikan kata yang dinilai dengan leksikon emosi yang telah dibuat sebelumnya. 26 dari 100 update status dapat diketahui label emosinya. Hasil

validasi menunjukkan terdapat 16 update status atau 61,53% emosinya akurat. Kedua penelitian yang dilakukan oleh (Ardiada et al., 2019) tentang text mining pada sosial media untuk mendeteksi emosi pengguna menggunakan metode Support Vector Machine dan K-Nearest Neighbour mengatakan bahwa penggunakan metode SVM dan KNN mengalami peningkatan nilai presisi, recall, accuracy dan kesesuaian disbanding dengan yang hanya menggunakan metode KNN saja. Penelitian yang menggunakan SVM dan KNN memperoleh hasil nilai presisi 0.4564, nilai recall sebesar 0.502, dan nilai accuracy sebesar 0.8104. Sedangkan penelitian yang hanya menggunakan metode KNN mendapatkan nilai rata-rata presisi sebesar 0.3421, nilai recall sebesar 0.4595, dan pada accuracy sebesar 0.797. Ketiga, merupakan penelitian yang dilakukan oleh (Sofiyana et al., 2012) tentang klasifikasi emosi untuk teks berbahasa Indonesia dengan menggunakan K-Nearest Neighbour menghasilkan prosentase sebesar 60% pada K=5. Pengklasifikasiannya menggunakan data latih yang sudah diketahui kelas emosinya yaitu senang, sedih, marah, bersalah, takut, dengan menggunakan metode *text mining* serta menggunakan algoritma K-Nearest Neighbour. Yang keempat, penelitian yang dilakukan oleh Afif Hijra, Andrew Briand, dan Casi Setianingsih (2018), menghasilkan nilai akurasi sebesar 75%. Metode yang digunakan adalah *data mining* dengan algoritma K-Nearest Neighbour. Kelima, adalah penelitian yang dilakukan oleh (Ilmiah et al., n.d.) yang membahas tentang klasifikasi emosi pada teks Bahasa Indonesia menggunakan metode K-Nearest Neighbour dengan pembobotan WIDF menghasilkan nilai akurasi sebesar 73,3% pada nilai K=5. Untuk nilai relevansi dari setiap kategori emosi dengan

menggunakan *precision* dan *recall* menghasilkan nilai rata-rata terbaik pada kategori "Takut" untuk *precisison*, dan kategori "Marah" untuk *recall*. Keenam, adalah penelitian yang dilakukan oleh (Saputra & Rosiyadi, 2019) yang membahas tentang perbandingan kinerja algoritma *K-Nearest Neighbour, Naïve Bayes Clasifier, dan Suport Vector Machine* dalam klasifikasi tingkah laku *bully* pada aplikasi *Whatsapp* menghasilkan nilai akurasi masing-masing 81,32%, 78,95%, dan 81,58%. Metode klasifikasi yang digunakan adalah metode *K-Nearest Neighbour, Naïve Bayes Clasifier, Suport Vector Machine*.

Berdasarkan latar belakang dan beberapa penelitian sebelumnya maka melalui penelitian ini penulis mengusulkan implementasi text mining pada social media facebook untuk mendeteksi emosi pengguna. Dalam penelitian ini penulis akan membandingkan metode mana yang menghasilkan nilai akurasi yang paling tinggi. Metode yang akan dibandingkan dalam penelitian ini diantaranya adalah metide *K-Nearest Neighbour (KNN), Naive Bayes, Suport Vector Machine (SVM)*. Deteksi emosi tersebut dapat digunakan pada berbagai bidang seperti perekrutan, konseling, perusahaan, profil psikologi, system rekomendasi, dan bidang yang lain yang membutuhkan deteksi emosi.

1.2. Rumusan Masalah

Berdasarkan latar belakang yang telah disampaikan penulis , maka rumusan masalahnya adalah sebagai berikut :

a. Dari metode *K-Nearest Neighbour, Naïve Bayes, dan Suport Vector Machine*, metode apakah yang menghasilkan nilai akurasi tertinggi?

b. Fitur apa saja yang mempengaruhi nilai akurasi tertinggi pada metode *K-Nearest Neighbour, Naïve Bayes, dan Suport Vector Machine*, untuk deteksi emosi pada social media facebook ?

1.3. Batasan Masalah

Adapun untuk batasan masalahnya adalah :

a. Penelitian ini menggunakan data status media sosial facebook yang dikumpulkan oleh peneliti sebelumnya (Rohman et al., 2019).
b. Dalam menyelesaikan penelitian ini, peneliti melakukan perbandingan menggunakan 3 metode yaitu *KNN, Naïve Bayes,* dan *SVM.*
c. Selain perbandingan metode , penelitian ini juga membandingkan 4 ekstraksi fitur yaitu Count Vector, TF-IDF Word level, TF-IDF Character level, TF-IDF N-gram level.
d. Tahap penelitian hanya sampai pada mencari nilai akurasi tertinggi dari deteksi emosi pengguna dalam sosial media tidak sampai pada pola emosi pengguna.
e. Deteksi emosi yang dilakukan meliputi deteksi emosi yang dikemukakan oleh Paul Ekman yang terdiri dari 6 klasifikasi emosi yaitu marah, bahagia, sedih, takut, jijik, terkejut seperti pada gambar 1.

	A	B	D	E
1	No	Update status	Keyword	Emosi
2	1	Awali pagimu dengan yang manis2	pagi	bahagia
3	2	Panas gaes	panas	sedih
4	3	Jngn hilang:)	jangan	sedih
5	4	ultah we	Ulang tahun	bahagia
6		gift skin m		
7	5	wes mytic rek cah ghopur karo cak mbatak	Sudah naek level	bahagia
8	6	Sekian lama usaha .. Sampaek menentang takdir dari server .. Dengan cara tidak murni..	Sekian lama	Kaget
9		Karena server buatan manusia...		
10		Dan manusia lah yg merusaknya...		
		kilo VSM Vsm Ghofur Vsm		

Sheet1 Sheet2 Sheet3

Gambar 1.1 Deteksi Emosi Menurut Paul Ekman

f. Akun yang diteliti pada penelitian ini merupakan akun dari perorangan atau pribadi.

1.4. Tujuan Penelitian

Adapun tujuan dari penelitian ini adalah :

a. Mengetahui metode yang menghasilkan nilai akurasi terbaik.

b. Mengukur presenstase nilai data training yang dibutuhkan untuk mendapatkan nilai accuracy terbaik.

1.5. Manfaat Penelitian

Bagian ini memuat penjelasan tentang:

a. Dapat mengetahui metode yang menghasilkan nilai akurasi tertinggi.

b. Dapat mengetahui presentase nilai data training yang dibutuhkan untuk mendapatkan nilai akurasi terbaik.

c. Sebagai media dalam mengembangkan ilmu pengetahuan teknologi dalam bidang *Machine Learning*.

d. Sebagai acuan bagi mahasiswa, terutama untuk mahasiswa yang ingin melakukan penelitian sejenis.

BAB II

TINJAUAN PUSTAKA

2.1. Tinjauan Pustaka

Tinjauan pustaka dari penelitian ini adalah yg pertama penelitian yang dilakukan oleh (Rohman et al. 2019) yang membahas tentang deteksi emosi media sosial dengan menggunakan pendekatan *leksikon* dan *natural language processing*. Dalam penelitian ini menggunak emolex sebagai leksikon yang digunakan untuk mendeteksi emosi pada suatu text. Kosa kata pada *emolex* diperluas dengan pencarian sinonim menggunakan *kateglo* API. EmoLex yang digunakan sebagai leksikon adalah 8 kategori dari Plutchik emosi dan sentimen. EmoLex tersedia dalam 105 bahasa berbeda termasuk Indonesia yang mana mengandung 14.182 kata yang kemudian diperluas dengan pencarian sinonim menggunakan Kateglo API. Pencarian sinonim menghasilkan 20.690 kata sehingga memperoleh hasil akhir leksikon emosi yang berisis 34.872 kata.

Hasil dari penelitian ini menunjukkan bahwa leksikon emosi mampu mendeteksi sebesar 55,45% atau 15.357 dari 27.696 kata yang diperoleh dari update status pengguna Facebook dalam melakukan pendeteksian emosi, sebanyak 100 update status diambil facebook. Kemudian update status tersebut diperbarui dengan *Natural Languange Processing* atau yang selanjutnya disebut NLP. Hasil perbaikannya dinilai dengan leksikon emosi yang telah dibuat sebelumnya. 26 dari 100 update status dapat diketahui label emosinya. Hasil validasi menunjukkan terdapat 16 update status atau 61,53% emosinya akurat.

Selanjutnya yang kedua adalah penelitian yang dilakukan oleh (Ferdinan et al., 2018) tentang klasifikasi emosi pada lirik lagu menggunakan metode *K-Nearest Neighbour* yang selanutnya disebut KNN. Dalam penelitian ini teks lirik lagu yang diambil adalah tesk lirik lagu yang berbahasa Indonesia. Peneliti mengambil dari situs *lirik.kapanlagi.com*. Selanjutnya data tersebut dibagi menjadi dua yaitu data latih dan data uji yang kemudian dilakukan pelabelan kepada keduanya. Pelabelan emosi dilakukan dengan menccokan kata-kata pada lirik lagu dengan rules yang berisi kata-kata yang mengandung label emosi. Rules yang dibuat sebelumnya telah divalidasi oleh Balai Bahasa Jawa Barat. Tujuan pelabelan manual pada data latih sebagai kelas yang digunakan untuk klasifikasi. Sedangkan tujuan pelabelan manual pada data uji adalah sebagai pengeckan kinerja sistem. Selanjutkan dilakukan proses *preprocessing* untuk mendapatkan data yang bersih. Dan selanjutkan dilakukan pembobotan dengan menggunakan metode TF-IDF. Masuk pada tahap terakhir yaitu klasifikasi menggunakan metode KNN.

Hasil dari penelitian ini adalah pada pengujian dengan metode KNN salah satu hal yang sangat mempengaruhi nilai akurasi adalah nilai K. Nilai K diuji dari nilai K=1 sampai nilai K=10 dan dari pengujian itu diketahui bahwa nilai akurasi tertinggi jatuh pada nilai K bernilai 6 yaitu sebesar 87.5%. Dan dihasilkan rata-rata akurasi dari pengujian 10 kali sebesar 75.

Selanjutkan penelitian ketiga yang dilakukan oleh (Bata et al. 2015) tentang leksikon untu deteksi emosi dari teks Bahasa Indonesia. Penelitian ini memaparkan tentang proses pengembangan leksikon emosi untuk teks Bahasa

Indonesia. Pengembangan leksikon tersebut terdapat 2 utama yaitu pemilihan seed words dan perluasan leksikon. Pemilihan seed words dipilih berdasarkan jenis emosi yaitu senang, cinta , marah, sedih, dan takut. Jumlah seed words yang digunakan sebanyak 124 kata. Perluasan leksikon dilakukan dengan kamus Tesaurus Bahasa Indonesia. Setiap kata dalam leksikon diberi bobot 1 atau 0. Leksikon yang dihasilkan sejumlah 1165 kata.

Penelitian yang keempat dilakukan oleh (Ardiada et al. 2019) tentang *text mining* pada sosial media untuk mendeteksi emosi pengguna menggunakan metode *Support Vector Machine* (SVM) dan *K-Nearest Neighbour* (KNN). Pada penelitian ini proses awal yang dilakukan adalah pengumpulan data pada twiter dengan menggunakan Stream Twitter API. Setelah itu masuk ke tahap Text Preprocessing yang terdiri dari tokenizing, filtering dan stemming. Selanjutnya tahap pembobotan dengan metode TF-IDF. Dan yang terakhir masuk ke klasifikasi dengan SVM dan KNN dan dilakukan pengujian.

Hasil pengujian yang dilakukan menunjukkan bahwa hasill klasifikasi dengan metode SVM dan KNN mengalami peningkatan nilai presisi, recall, accuracy dan kesesuain yang signifikan dibandingkan dengan menggunakan metode KNN saja dengan nilai rata-rata presisi sebesar 0,4564, nilai recall sebesar 0,502, dan pada nilai accuracy sebesar 0,8104. Sedangkan jika hanya menggunakan metode KNN diperoleh rata-rata nilai presisi sebesar 0,3421, nilai recall sebesar 0,4595 dan pada nilai accuracy sebesar 0,797. Dan setelah dilakukan pengujian komputasi selisih waktu dari metode SVM dan KNN dengan hanya menggunakan KNN tidak mempunyai selisih yang signifikan.

Kelima adalah penelitian yang dilakukan oleh (Ilmiah et al., n.d.) tentang klasifikasi emosi pada teks Bahasa Indonesia menggunakan metode K-Nearest Neighbour (KNN) dengan pembobotan WIDF. Data training yang digunakan oleh peneliti adalah data yang berasal dari ISEAR kemudian di terjemahkan menjadi Bahasa Indonesia tanpa menguramgi maksud dan tujuannya. Sedangkan data latih yang digunakan diambil dari lirik lagu dengan format .txt. Selanjutnya masuk ke dalam proses preprocessing yang meliputi case folding, convert negation, tokenizing, stopword removal, dan stemming. Setelah itu masuk kedalam proses pembobotna dengan WIDF. Dan yang terakhir ke tahap klasifikasi dengan metode KNN.

Hasil dari penelitian ini mengatakan keberhasilan pengkalsifikasin emosi ini dipengaruhi oleh nilai K pada KNN dan jumlah data latih. Penggunaan pembobotan WIDF dalam pengklasifikasian ini juga mendapatkan nilai akurasi yang lebih tinggi dari penelitian sebelumnya yaitu sebesar 73,3%. Untuk nilai relevansi dari setiap kategori emosi dengan menggunakan precision dan recall menghasilkan nilai maksimum sebesar 100%. Nilai tersebut didapat di kategori "Takut" untuk precission dan "Marah" untuk recall.

Penelitian terakhir yang menjadi tinjuauan pustaka dalam penelitian ini yaitu milik (Islam et al., 2018) yang meneliti tentang deteksi depresi dengan menggunakan Teknik K-Nearest Neighbour (KNN). Teknik metode KNN yang digunakan dalam penelitian ini adalah KNN akhir, KNN sedang, KNN kasar, kosinus KNN, kubik KNN dan KNN berbobot. Data yang digunakan pada penelitian ini data komentar pengguna facebook. Alat yang digunakan untuk

mengumpulkan data adalah *NCapture*. Setelah terkumpul data tersebut diproses dengan menggunakan LIWC2015. Data setelah diproses akan dilabeli oleh dua orang ahli menjadi data komentar yang terindikasi depresi atau data positif (YA) dan data tidak terindikasi depresi atau data negatif (TIDAK). Data yang didapat berjumlah 7145 komentar yang dibagi menjadi 58% komentar positif (YA) dan 42% komentar negative (TIDAK).

Hasil dari penelitian ini mengatakan bahwa teknik terbaik dari metode KNN yang telah diuji yaitu KNN kasar dengan rata-rata mendapat 60%-70% dalam hal tingkat metrik yang berbeda.

2.2. KEASLIAN PENELITIAN

Tabel 2.1 Matriks literatur review dan posisi penelitian
Text mining pada sosial media facebook untuk mendeteksi emosi pengguna

No	Judul	Peneliti, Media Publikasi, dan Tahun	Tujuan Penelitian	Kesimpulan	Saran atau Kelemahan	Perbandingan
1	Detecting Depression Using K-Nearest Neighbors (KNN) Classification Technique	Md R Islam , A R M. Kamal, N Sultana , R Islam, M A Moni , Anwaar ulhaq , IEEE, 2018	Untuk mengetahui depresi emosi dan dan proses evolusi dalam data jaringan sosial dengan menguraikan penelitian dari eksplorasi yang ada, memberikan kasus dan prosedur baru, memberikan tangga jalan untuk penelitian di masa yang akan dating.	Penelitian ini menunjukkan kemampuan memanfaatkan data Facebook sebagai sumber untuk mengukur dan mendeteksi depresi besar di antara pengguna. Kita belajar empat jenis faktor (emosional dalam proses, proses temporal, gaya linguistik, dan semua fitur) dan dilatih model untuk memanfaatkan setiap jenis independen dan bersama-sama. Temuan kami menunjukkan bahwa hasil dataset kebenaran tanah dan berbagai jenis hasil teknik KNN bervariasi antara 60-70% dalam hal tingkat metrik yang berbeda.	Untuk menggunakan teknik lain yang digunakan mengekstrak parafrasa dari lebih jenis fitur emosional.	Perbandingan dengan penelitian yang akan datang adalah pertama terletak pada data yang ditambang, dalam penelitian ini melakukan penambangan data pada komentar pengguna facebook sedangkan penelitian yang akan datang menambang data dari status facebook. Kedua dalam penelitian ini membahas berbagai macam teknik KNN, sedangkan penelitian yang akan datang membahas tentang beberapa metode seperti KNN, Naive Bayes dan SVM.

Tabel 2.1 Matriks literatur review dan posisi penelitian
Text mining pada sosial media facebook untuk mendeteksi emosi pengguna (Lanjutan)

No	Judul	Peneliti, Media Publikasi, dan Tahun	Tujuan Penelitian	Kesimpulan	Saran atau Kelemahan	Perbandingan
2	Deteksi Emosi Media Sosial Menggunakan Pendekatan Leksikon dan *Natural Language Processing*	Arif Nur Rohman, Ema Utami, Suwanto Raharjo, Jurnal Eksplora Informatika, 2019	Penelitian ini bertujuan untuk mendeteksi kondisi emosi pada media sosial Facebook menggunakan pendekatan leksikon dan NLP	Kesimpulan dari penelitian ini adalah hasil dari penelitian ini menunjukkan bahwa leksikon emosi mampu mendeteksi 55.45% atau 15.357 dari 27.696 kata yang diperoleh dari *update status* pengguna Facebook. NLP dapat dimanfaatkan untuk memperbaiki teks yang berasal dari *update* status. Hasil perbaikan dari NLP dicocokkan dengan leksikon yang telah dibuat untuk mengetahui label emosi dari suatu *update* status. Sebanyak 26 *update status* dapat terdeteksi label emosinya dan 61,53% di antaranya akurat.	Dalam penelitian ini peneliti memberi saran kepada penelitian selanjutnya untuk dapat meningkatkan nilai akurasi dari pelabelan emosi. Meningkatkan prapemrosesan dengan membuang nama orang, nama tempat dan kata sambung, serta kata yang tidak baku dalam bahasa Indonesia.	Data penelitian yang akan dilakukan beraal dari penelitian ini. Perbandingan dengan penelitian yang akan dilakukan yaitu dalam penelitian ini disebutkan bahwa untuk mencapai hasil peneliti menggunakan pendekatan leksikon dan NLP. Sedangkan penelitian yang akan dilakukan adalah membandingkan 3 metode yaitu KNN, Naive Bayes dan SVM untuk mendapatkan nilai akurasi tertinggi.

Tabel 2.1 Matriks literatur review dan posisi penelitian
Text mining pada sosial media facebook untuk mendeteksi emosi pengguna (Lanjutan)

No	Judul	Peneliti, Media Publikasi, dan Tahun	Tujuan Penelitian	Kesimpulan	Saran atau Kelemahan	Perbandingan
3	Klasifikasi Emosi Pada Lirik Lagu Menggunakan K-Nearest Neighbour	Afif Hijra Ferdinan, Andrew Briand Osmond, Casi Setianingsih, e-Proceeding of Engineering : Vol.5, No.3 Desember 2018 Page 6187, 2018	Adapun tujuan dari penelitian ini adalah merancang sistem yang dapat mengklasifikasikan emosi berdasarkan lirik lagu secara otomatis agar pengguna dapat memilih lagu yang sesuai dengan suasana hatinya.	Terdapat 3 point kesimpulan dalam penelitian ini , yang pertama proses klasifikasi data lirik lagu hanya diberikan 4 jenis kelas label emosi yaitu cinta, marah, senang, dan sedih menggunakan algoritma k-*Nearest Neighbor*. Kedua sistem yang dirancang berhasil mengklasifikasikan emosi berdasarkan lirik lagu secara otomatis. Dan yang terakhir Nilai k terbaik adalah k = 6 dengan akurasi sebesar 87,5% dan rata-rata pada 10 kali pengujian memiliki nilai akurasi sebesar 75%	-	Perbandingan dengan penelitian yang akan dilakukan adalah dalam penelitian ini dataset yang dilakukan adalah dataset lirik lagu Bahasa Indonesia yang didapat dari *lirik.kapanlagi.com*. Sedangkan untuk penelitian yang akan dilakukan menggunakan dataset dari update status facebook. Selain itu dalam penelitian ini menggunakan pembobotan TF-IDF dengan metode KNN, sedangkan yang akan dilakukan menggunakan pembobotan TF-IDF juga namun dalam penelitian ini metode yang akan digunakan ada 3. Sehingga dari ketiga metode tersebut dicari nilai akurasi yang paling tinggi.

Tabel 2.1 Matriks literatur review dan posisi penelitian
Text mining pada sosial media facebook untuk mendeteksi emosi pengguna (Lanjutan)

No	Judul	Peneliti, Media Publikasi, dan Tahun	Tujuan Penelitian	Kesimpulan	Saran atau Kelemahan	Perbandingan
4	Leksikon Untuk Deteksi Emosi Dari Teks Bahasa Indonesia	Jullus Bata, Suyoto, Pranowo, Seminar Nasional Informatika ISSN: 1979-2328, 2015	Penelitian dalam makalah ini merupakan bagian dari penelitian pengembangan model deteksi emosi dari teks bahasa Indonesia. Leksikon emosi pada penelitian ini adalah hasil awal dalam pengembangan leksikon emosi bahasa Indonesia. Leksikon emosi dikembangkan menggunakan buku tesaurus bahasa Indonesia. Kontribusi utama penelitian ini adalah mengembangkan leksikon emosi untuk bahasa Indonesia.	Makalah ini memuat hasil awal pengembangan leksikon emosi untuk bahasa Indonesia. Pada makalah ini sumber yang digunakan untuk mengembangkan leksikon adalah Tesaurus Bahasa Indonesia. Proses utama dalam pengembangan leksikon adalah pemilihan *seed words* dan perluasan leksikon. Kedua proses tersebut menghasilkan 1165 kata dalam leksikon emosi bahasa Indonesia. Pada makalah ini hanya melakukan pengembangan leksikon. Beberapa hal yang dapat dilakukan pada penelitian selanjutnya adalah mengukur unjuk kerja dari leksikon dengan melakukan deteksi emosi dari teks dengan menggunakan bobot serupa yaitu biner	Penelitian selanjutnya adalah menggunakan pembobotan lain seperti tfidf. Penelitian selanjutnya perlu menggunakan pendekatan lain seperti *corpus-based* untuk mengembangkan leksikon emosi.	Perbandingan dengan penelitian yang akan dilakukan adalah dalam penelitian melakukan pengembangan yang telah dilakukan dengan menggunakan leksikon pada Bahasa Indonesia. Sedangkan untuk penelitian yang akan dilakukan adalah membandingkan 3 metode yaitu metode KNN, Naive Bayes, dan SVM untuk mendapatkan nilai akurasi terbaik.

Tabel 2.1 Matriks literatur review dan posisi penelitian
Text mining pada sosial media facebook untuk mendeteksi emosi pengguna (Lanjutan)

No	Judul	Peneliti, Media Publikasi, dan Tahun	Tujuan Penelitian	Kesimpulan	Saran atau Kelemahan	Perbandingan
5	Evaluation of Classification Methods for Indonesian Text Emotion Detection	Muljono, Nurul Anisa Sri Winarsih, Catur Supriyanto, IEEE, 2016	Untuk mengetahui metode yang mencapai nilai akurasi tertinggi dan kinerja terbaik dari metode *Naïve Bayes*, J48, KNN dan SVM-SMO.	Kesimpulan yang didapat dari penelitian ini adalah pada evaluasi pertama, validasi 10fold diterapkan untuk mendapatkan akurasi global termasuk presisi, pemanggilan ulang, f-Measure, dan akurasi dari setiap kelas emosi. Nilai tertinggi dalam akurasi global ditunjukkan oleh SVM-SMO yang mencapai 85,5%. Dalam evaluasi kedua dengan menggunakan validasi Split, SVMSMO juga memberikan kinerja terbaik dengan mencapai akurasi tertinggi 86%. Dengan menggunakan validasi 10 kali lipat dan validasi Split, SVM-SMO menghasilkan akurasi tertinggi dan KNN menghasilkan akurasi terendah.	Saran dari penelitian ini yaitu untuk kedepannya penelitian ini dapat dikembangkan lebih lanjut dengan meningkatkan akurasi Svm-SMO dan mengintegrasikan dengan beberapa metode optimasi, seperti algoritma genetik dan partikel kawanan optimasi.	Perbandingan dengan penelitian yang akan dilakukan yaitu dalam penelitian ini mendeteksi emnosi dalam dongeng yang diambil dari www.dongengceritarakyat .com, www.dongeng.org, and www.pendongeng.com dan menghasilkan metode SVM-SMO memiliki akurasi tertinggi , sedangkan penelitian yang akan dilakukan menggunakan dataset dari timeline status pengguan facebook dan data yang diambil berupa teks update status. Selanjutnya membandingkan metode KNN, Naive Bayes, SVM untuk mendapatkan nilai akurasi terbaik.

Tabel 2.1 Matriks literatur review dan posisi penelitian
Text mining pada sosial media facebook untuk mendeteksi emosi pengguna (Lanjutan)

No	Judul	Peneliti, Media Publikasi, dan Tahun	Tujuan Penelitian	Kesimpulan	Saran atau Kelemahan	Perbandingan
6	Personality Classification Based on Twitter Text Using Naive Bayes, KNN and SVM	Bayu Yudha Pratama, Riyanarto Sarno, IEEE, 2015	Tujuan dari penelitian ini adalah untuk memprediksi kepribadian dari teks yang ditulis oleh pengguna media sosial Twitter.	Penelitian ini menghasilkan kesimpulan bahwa kepribadian pengguna berhasil diprediksi dari teks yang ditulis di Twitter. Dalam tiga metode yang digunakan, Naive Bayes sedikit mengungguli metode lain dengan 60%. Eksperimen gagal meningkatkan akurasi dari penelitian sebelumnya. Sistem memiliki 65% akurasi dibandingkan dengan tes kuesioner berdasarkan. Perbaikan lebih lanjut dapat dilakukan dengan menggunakan dataset yang lebih akurat untuk meningkatkan keakuratan dan penggunaan bahasa Indonesia asli klasifikasi Indonesia (tidak diterjemahkan).	Saran untuk Studi masa depan dapat mencakup pendekatan semantik untuk mempertimbangkan arti dari setiap kata.	Perbandingan dengan penelitian yang akan dilakukan untuk penelitian yang akan dilakukan menggunakan dataset dari update status facebook dan pada penelitian ini menggunakan dasaset menggunakan twiter. Dan untuk penelitian yang akan dilakukan berupa perbandingan metode KNN, Naive Bayes dan SVM untuk menghasilkan nilai akurasi tertinggi.

2.3. Landasan Teori

a. Pengertian Emosi

Emosi adalah keadaan perasaan yang banyak berpengaruh pada perilaku manusia. Biasanya emosi merupakan reaksi terhadap rangsang dari luar dan dalam diri individu. Emosi berkaitan dengan perubahan fisiologis dan berbagai pikiran. Jadi emosi merupakan salah saru aspek penting dalam kehidupan manusia. ltu sering dianggap negatif. Padahal tanpa adanya emosi kehidupan manusia akan sangat kering. Terutama dalam hubungan dengan orang lain emosi banyak berperan. Hubungan antar manusia akan lebih baik atau lebih buruk tergantung ungkapan emosi yang dilakukan mereka. Dua orang atau lebih yang banyak mengungkapkan rasa kasih melalui senyuman, kegembiraan, kehangatan, dan penerimaan akan lebih menyenangkan bagi mereka maupun yang memperhatikan. Sebalikuya dua orang atau Iebih yang banyak mengungkapkan kedengkian melalui cemoohan, ejekan, keirian, kemarahan, saling menjatuhkan akan menimbulkan kengerian di antara mereka ataupun bagi yang memperhatikannya (Prawitasllri, 1991).

Menurut (David, n.d.) Emosi adalah perasaan intens yang diarahkan pada seseorang atau sesuatu. Suasana hati adalah perasaan yang cenderung kurang intens daripada emosi dan bahwa sering (meskipun tidak selalu) tidak memiliki rangsangan kontekstual. Emosi dapat digambarkan sebagai keadaan yang pada umumnya disebabkan oleh suatu kejadian,

seseorang, atau lingkungan luar lainnya atau biasa disebut dengan istilah stimulus. Pada saat manusia bereaksi terhadap lingkungannya maka emosi akan tercipta disaat bersamaan. Ketika emosi tercipta pada diri seseorang, pada umumnya manusia akan menunjukkan ekspresi wajah atau reaksi tubuh sesuai dengan emosi yang tercipta. Menurut ahli filosofi emosi, Rene Descartes, emosi terbagi menjadi enam yaitu heran, cinta, marah, hasrat, senang, dan sedih.

Menurut (Paul Ekman et al. 2013) ekspresi emosi dasar terdiri dari 6 ekspresi wajah seperti bahagia, terkejut, marah, jijik, sedih, dan takut.

b. Text Mining

Text Mining merupakan proses ekstraksi pola (informasi dan pengetahuan yang berguna) dari sejumlah besar sumber data tak terstruktur. Penambangan teks memiliki tujuan dan menggunakan proses yang sama dengan penambangan data, hanya saja masukannya yang berbeda. Masukan untuk penambangan teks adalah data yang tidak (atau kurang) terstruktur, seperti dokumen Word, PDF, kutipan teks, dll., sedangkan masukan untuk penambangan data yaitu data yang terstruktur (Ronen Feldman, 2007). Penambangan teks dapat dianggap sebagai proses dua tahap yang diawali dengan penerapan struktur terhadap sumber data teks dan dilanjutkan dengan ekstraksi informasi dan pengetahuan yang relevan dari data teks terstruktur dengan menggunakan teknik dan alat yang sama dengan penambangan data.

Area penerapan penambangan teks yang paling populer adalah:

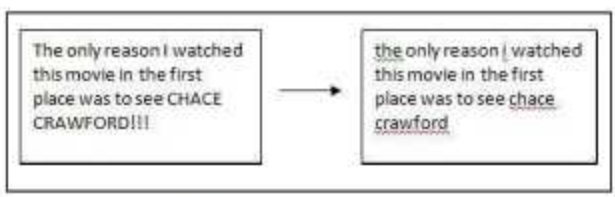
1. Ekstraksi informasi (information extraction): Identifikasi frasa kunci dan keterkaitan di dalam teks dengan melihat urutan tertentu melalui pencocokan pola.
2. Pelacakan topik (topic tracking): Penentuan dokumen lain yang menarik seorang pengguna berdasarkan profil dan dokumen yang dilihat pengguna tersebut.
3. Perangkuman (summarization): Pembuatan rangkuman dokumen untuk mengefisienkan proses membaca.
4. Kategorisasi (categorization): Penentuan tema utama suatu teks dan pengelompokan teks berdasarkan tema tersebut ke dalam kategori yang telah ditentukan.
5. Penggugusan (clustering): Pengelompokan dokumen yang serupa tanpa penentuan kategori sebelumnya (berbeda dengan kategorisasi di atas).
6. Penautan konsep (concept linking): Penautan dokumen terkait dengan identifikasi konsep yang dimiliki bersama sehingga membantu pengguna untuk menemukan informasi yang mungkin tidak akan ditemukan dengan hanya menggunakan metode pencarian tradisional.
7. Penjawaban pertanyaan (question answering): Pemberian jawaban terbaik

c. *Text Preprocessing*

Struktur data yang baik dapat memudahkan proses komputerisasi secara otomatis. Pada *Text Mining*, informasi yang akan digali berisi informasi-informasi yang strukturnya sembarang. Oleh karena itu, diperlukan proses pengubahan bentuk menjadi data yang terstruktur sesuai kebutuhannya untuk proses dalam data mining, yang biasanya akan menjadi nilai-nilai numerik. Proses ini sering disebut *Text Preprocessing* (Ronen Feldman, 2007). Setelah data menjadi data terstruktur dan berupa nilai numerik maka data dapat dijadikan sebagai sumber data yang dapat diolah lebih lanjut. Berberapa proses yang dilakukan adalah sebagai berikut :

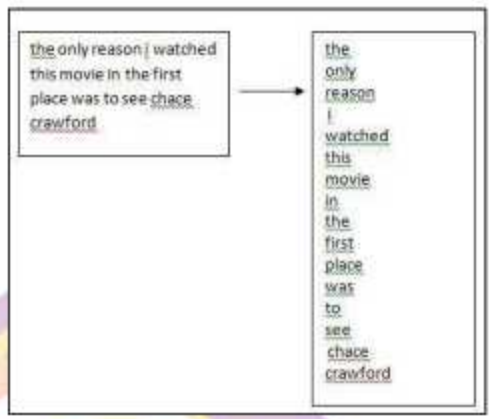
a) *Case folding*

Case folding adalah mengubah semua huruf dalam dokumen menjadi huruf kecil. Hanya huruf 'a' sampai dengan 'z' yang diterima. Karakter selain huruf dihilangkan dan dianggap delimiter (Ronen Feldman, 2007).



Gambar 2.1 Proses *Case Folding*

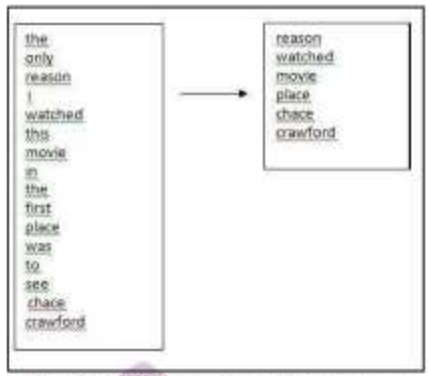
b) Tahap *Tokenizing* adalah tahap pemotongan *string input* berdasarkan tiap kata yang menyusunnya.



Gambar 2.2 Proses *Tokenizing*

c) *Stopword removal* atau *filtering*

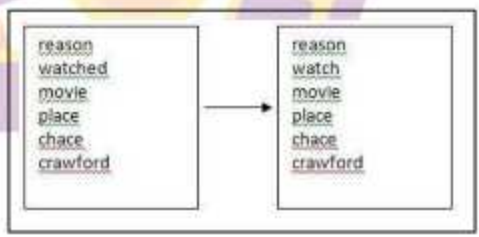
Tahap filtering adalah tahap mengambil kata - kata penting dari hasil token. Bisa menggunakan algoritma stoplist (membuang kata yang kurang penting) atau wordlist (menyimpan kata penting). Stoplist / stopword adalah kata-kata yang tidak deskriptif yang dapat dibuang dalam pendekatan bag-of-words (Ronen Feldman, 2007).



Gambar 2.3 Proses *Stopword removal*

d) *Stemming*

Tahap *stemming* adalah tahap mencari root kata dari tiap kata hasil filtering. Pada tahap ini dilakukan proses pengembalian berbagai bentukan kata ke dalam suatu representasi yang sama.



Gambar 2.4 Proses *Stemming*

d. K-Nearest Neighbour

Algoritma *k-nearest neighbor* (KNN) adalah sebuah metode untuk melakukan klasifikasi terhadap objek berdasarkan data pembelajaran yang jaraknya paling dekat dengan objek tersebut. KNN termasuk algoritma supervised learning dimana hasil dari query instance yang baru diklasifikan berdasarkan mayoritas dari kategori pada KNN. Tujuan dari algoritma ini

adalah mengklasifikasikan obyek baru bedasarkan atribut dan training sample. Classifier tidak menggunakan model apapun untuk dicocokkan dan hanya berdasarkan pada memori. Diberikan titik query, akan ditemukan sejumlah k obyek atau (titik training) yang paling dekat dengan titik query. Klasifikasi menggunakan voting terbanyak diantara klasifikasi dari k obyek.. algoritma k-nearest neighbor (KNN) menggunakan klasifikasi ketetanggaan sebagai nilai prediksi dari query instanceyang baru (Wu et al. 2009).

Algoritma metode *K-Nearest Neighbor* (KNN) sangatlah sederhana, bekerja berdasarkan jarak terpendek dari query instance ke training sample untuk menentukan KNN-nya. Training sample diproyeksikan ke ruang berdimensi banyak, dimana masing-masing dimensi merepresentasikan fitur dari data. Ruang ini dibagi menjadi bagian-bagian berdasarkan klasifikasi training sample. Sebuah titik pada ruang ini ditandai kelas c jika kelas c merupakan klasifikasi yang paling banyak ditemui pada k buah tetangga terdekat dari titik tersebut. Dekat atau jauhnya tetangga biasanya dihitung berdasarkan Euclidean Distance. Jarak Euclidean paling sering digunakan menghitung jarak. Jarak euclidean berfungsi menguji ukuran yang bisa digunakan sebagai interpretasi kedekatan jarak antara dua obyek. yang direpresentasikan sebagai berikut (Banjarsari et al. 2016) :

$$D(a,b) = \sqrt{\sum_{k=1}^{d}(a_k - b_k)^2,}$$

Keterangan:

x1 = Sampel data

x2 = Data uji atau data testing

i = Variabel data

d = Jarak

p = Dimensi data

Dimana matriks D (a,b) adalah jarak skalar dari kedua vektor adan b dari matriks dengan ukuran ddimensi. Semakin besar nilai D akan semakin jauh tingkat keserupaan antara kedua individu dan sebaliknya jika nilai D semakin kecil maka akan semakin dekat tingkat keserupaan antar individu tersebut.

Nilai k yang terbaik untuk algoritma ini tergantung pada data. Secara umum, nilai k yang tinggi akan mengurangi efek noise pada klasifikasi, tetapi membuat batasan antara setiap klasifikasi menjadi semakin kabur. Nilai k yang bagus dapat dipilih dengan optimasi parameter, misalnya dengan menggunakan *cross-validation*. Kasus khusus dimana klasifikasi diprediksikan berdasarkan training data yang paling dekat (dengan kata lain, k = 1) disebut algoritma nearest neighbor. Ketepatan algoritma KNN sangat dipengaruhi oleh ada atau tidaknya fitur-fitur yang tidak relevan atau jika bobot fitur tersebut tidak setara dengan relevansinya terhadap klasifikasi. Riset terhadap algoritma ini sebagian besar membahas bagaimana memilih dan memberi bobot terhadap fitur agar performa klasifikasi menjadi lebih baik (Musa & Alang, 2017).

Langkah-langkah untuk menghitung metode *K-Nearest Neighbor* :

1. Menentukan parameter K (jumlah tetangga paling dekat).

2. Menghitung kuadrat jarak euclid (query instance) masing-masing obyek terhadap data sampel yang diberikan.
3. Kemudian mengurutkan objek-objek tersebut kedalam kelompok yang mempunyai jarak euclid terkecil.
4. Mengumpulkan kategori Y (Klasifikasi nearestneighbor)
5. Dengan menggunakan kategori nearest neighbor yang paling mayoritas maka dapat dipredisikan nilai query instance yang telah dihitung.

e. Naive Bayes

Naive Bayes merupakan sebuah pengklasifikasian probabilistik sederhana yang menghitung sekumpulan probabilitas dengan menjumlahkan frekuensi dan kombinasi nilaidari dataset yang diberikan. Algoritma mengunakan teorema Bayes dan mengasumsikan semua atribut independen atau tidak saling ketergantungan yang diberikan oleh nilai pada variabel kelas (Patil & Sherekar,2013). Naïve Bayes merupakan metode klasifikasi yang statistic berdasarkan teorema Bayes (Kabir et al., 2011). Naïve Bayes berpotensi baik untuk mengklasifikasikan data karena kesederhanaannya (Abraham et al. 2009) &(Ting at el. 2011) .

Persamaan yang digunakan pada Naïve Bayes:

$$P\ c_i\ X\ = \frac{P\ x\ c_i\ P\ c_i}{P_X}$$

P(c_i|X) yaitu, probabilitas $c_i$terjadi jika X sudah terjadi.

P( ) adalah kemungkinan didata, bersifat independent terhadap X

X adalah kumpulan atribut.

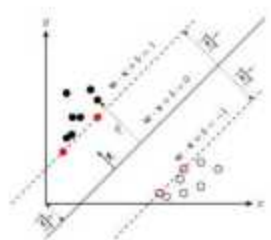
P(X| c_i) adalah probabilitas X terjadi jika c_i benar atau sudah terjadi berdasarkan data pelatihan.

Selama nilai P(x) konstan, maka dapat disederhanakan menjadi rumus berikut ini:

$$P\ C_i X\ = P\ X\ C_i\ P(C_i)$$

f. Suport Vector Machine (SVM)

Support Vector Machine merupakan salah satu metode klasifikasi dengan menggunakan machine learning (supervised learning) yang memprediksi kelas berdasarkan model atau pola dari hasil proses training. Klasifikasi dilakukan dengan mencari hyperplane atau garis pembatas (decision boundary) yang memisahkan antara suatu kelas dengan kelas lain (Novantirani et al. 2015). Support Vector Machine melakukan pencarian nilai hyperplane dengan menggunakan support vector dan nilai margin.



Gambar 2.5 SVM berusaha untuk menemukan hyperplane terbaik yang memisahkan kedua kelas -1 dan +1 (Susilowati et al. 2015).

Gambar 6. memperlihatkan beberapa pattern yang merupakan anggota dari dua buah class: +1 dan –1. Pattern yang tergabung pada class –1 disimbolkan dengan warna putih sedangkan pattern pada class +1,

disimbolkan dengan warna hitam. Kedua kelas tersebut dipisahkan oleh garis yang disebut hyperplane Persamaan garis hyperplane adalah:

$$w.x + b = 0 \quad (2.1)$$

(w adalah normal bidang dan b adalah bias)

Pattern yang memiliki jarak paling dekat dengan hyperplane disebut support vector, disimbolkan dengan warna merah. Support Vector Machine memisahkan data menggunakan hyperplane dengan batas antar kelas terbesar. Maka dari itu dibentuk garis pembatas yang sejajar dengan support vector semua kelas.

Persamaan yang terbentuk dari garis pembatas tersebut adalah

$$\text{Garis pembatas positif} = w.x + b = 1 \quad (2.2)$$

$$\text{Garis pembatas negatif} = w.x + b = -1 \quad (2.3)$$

Dari persamaan tersebut dapat diketahui data yang digolongkan positif adalah data yang memiliki nilai persamaan (2.2) sedangkan data yang digolongkan negatif adalah data yang memiliki nilai persamaan (2.3). Dari hasil tersebut dapat dibentuk pertidaksamaan diantara kedua gari pembatas dengan rumus:

$$yi\,(xi.w + b) - 1 \geq 0 \quad (2.4)$$

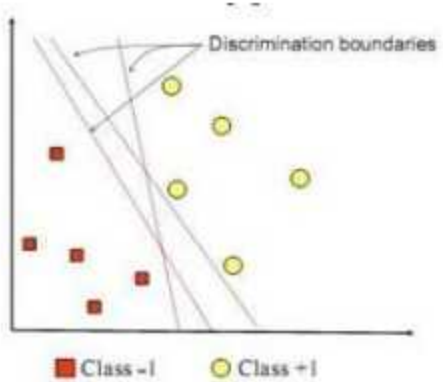
Nilai batas pada setiap bidang pembatas ke hyperplane adalah $\frac{1}{||w||}$, maka dapat ditentukan nilai batas gabungan adalah $\frac{1}{||w||} + \frac{1}{||w||} = \frac{2}{||w||}$.

Selanjutnya nilai batas ini akan dimaksimalkan, memaksimalkan nilai batas berarti meminimalkan nilai dari ||w|| sebagai penyebut. Jika kedua bidang pembatas sesuai dengan pertidaksamaan diatas, maka pencarian

hyperplane dengan batas terbesar dapat dirumuskan menjadi masalah optimasi konstrain, yaitu:

$$\frac{1}{2}|w|2 \quad (2.5)$$

$$yi\ (xi.w + b) - 1 \geq 0 \quad (2.6)$$



Gambar 2.6 Hyperplane terbentuk diantara class-1 dan +1 (Susilowati et al. 2015).

Hyperplane pemisah terbaik antara kedua class dapat ditemukan dengan mengukur margin hyperplane tersebut dan mencari titik maksimalnya. Margin adalah jarak antara hyperplane tersebut dengan pattern terdekat dari masing-masing kelas. Pattern yang paling dekat ini disebut sebagai support vector. Garis solid pada gambar menunjukkan hyperplane yang terbaik, yaitu yang terletak tepat pada tengah- tengah kedua kelas, sedangkan titik merah dan kuning yang berada dalam lingkaran hitam adalah support vector. Usaha untuk mencari lokasi hyperplane ini merupakan inti dari proses pembelajaran pada Support Vector Machine (Susilowati et al. 2015)

BAB III

METODE PENELITIAN

3.1. Jenis, Sifat dan Pendekatan Penelitian

Adapun jenis, sifat dan pendekatan penelitian yang akan dilakukan pada peneltian ini sebagai berikut:

a. Jenis Penelitian Eksperimen

Penelitian ini dilakukan untuk mengetahui metode yang menghasilkan tingkat akurasi tertinggi dalam deteksi emosi pada sosial media facebook.

b. Sifat Penelitian Deskriptif

Penelitian ini menjelaskan tahapan-tahapan untuk mengetahui metode yang menghasilkan tingkat akurasi tertinggi dalam deteksi emosi pada sosial media facebook.

c. Pendekatan Penelitian Kuantitatif

Penelitian ini menggunakan pendekatan kuantitatif yaitu hasil dari penerapan metode-metode untuk deteksi emosi berupa angka dan diagram yang menunjukkan tingkat akurasi dari penerapan metode-metode tersebut.

3.2. Metode Pengumpulan Data

Dataset yang digunakan untuk penelitian ini adalah dataset yang dikumpulkan oleh Arif Nur Rohman, Ema Utami, Suwanto Raharjo dalam penelitiannya yang berjudul *Deteksi Emosi Media Sosial Menggunakan Pendekatan Leksikon dan Natural Language Processing.* Dataset ini diambil dari sosial media facebook yang berupa data update status yang telah diberi beberapa label yaitu marah, bahagia, jijik, sedih, takut, terkejut, kaget. Adapun untuk proses pengumpulan data dapat dilihat dalam gambar 8 dibawah ini

Gambar 3.1 Diagram proses pengambilan data dari facebook.

Tahapan pengambilan data oleh (Rohman et al. 2019) meliputi scrapping timeline pada sosial facebook yang dipilih secara acak. Update status yang digunakan hanya yang berbahasa Indonesia saja. Pengumpulan dilakukan

dengan tool scrapping yang dibuat oleh peneliti sebelumnya (Rohman et al.2019). Tool yang telah dibuat ini bekerja dengan diberikan masukan username pengguna Facebook yang ingin diambil datanya. Keluaran dari tool ini berupa teks update status pengguna Facebook yang selanjutnya disimpan pada database MySQL untuk memudahkan pengelolaan. Dataset pada penelitian tersebut terdiri dari update status, *keyword*, label emosi.

3.3. Metode Analisis Data

Metode analisi data yang dilakukan untuk mengetahui metode yang menghasilkan nilai akurasi terbaik adalah

a. Preprocessing

Tahapan yang dilakukan pada preprocessing ada 4 langkah yaitu :

1) Case Folding

Tidak semua dokumen teks hanya menggunakan huruf kapital. Oleh karena itu, Case folding digunakan untuk mengubah semua huruf dalam dokumen menjadi huruf kecil. Hanya huruf 'a' sampai dengan 'z' yang diambil, karakter selain huruf dihilangkan. Tahap ini menggunakan phyton NLTK.

2) Tokenizing

Tokenizing adalah pengubahan kalimat menjadi kata. Tahap ini menggunakan phyton NLTK.

3) Stemming

Stemming yaitu mengubah suatu kata menjadi bentuk kata dasarnya untuk menghilangkan imbuhan-imbuhan baik itu berupa prefix, sufiks, maupun konfiks yang ada pada setiap kata. Stemming menggunakan phyton sastrawi.

4) Stopworld Removal

Pada tahap ini dilakukan proses penghapusan kata-kata yang terdapat pada stoplist. Stoplist berisi kosakata-kosakata yang bukan merupakan ciri dari dokumen, dalam hal ini adalah jenis emosi. Stopword removal menggunakan phyton sastrawi.

b. Ektraksi Fitur

Ekstraksi fitur yang akan diterapkan pada penelitian ini adalah pembobotan Count vector, TF-IDF word level, TF-IDF character level dan TF-IDF Ngram level. Selanjutnya akan di bandingkan keempat ekstraksi fitur tersebut.

c. Klasifikasi

Setelah ekstraksi fitur tentunnya akan masuk ke tahap selanjutnya yaitu tahap klasifikasi. Pada tahap klasifikasi ini akan dilakukan perhitungan dengan metode *K-Nearest Neighbour, Naive Bayes atau Suport Vector Machine (SVM).*

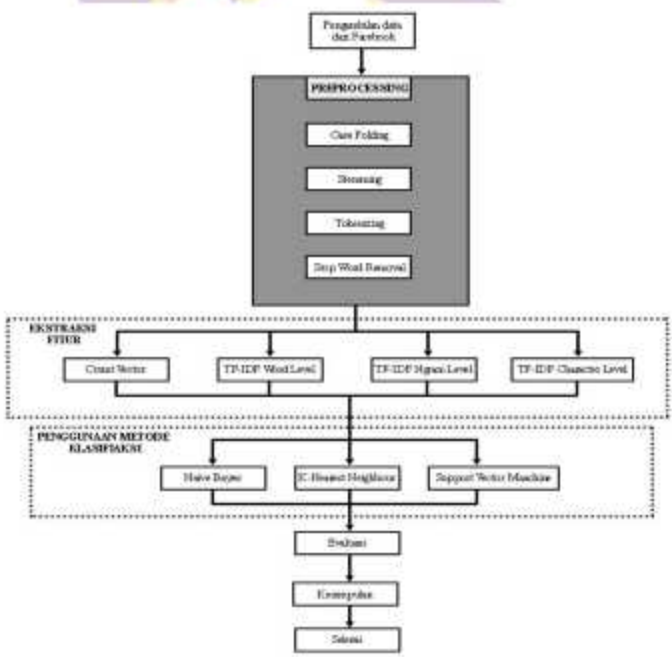
d. Evaluasi

Dari hasil klasifikasi akan di evaluasi untuk mengetahui nilai dari akurasi masing-masing metode yang digunakan.

Kemudian setelah mengetahui nilai akurasi masing-masing metode Langkah selanjutnya adalah membuat kesimpulan. Dalam kesimpulan ini nanti akan diketahui pasangan ekstraksi fitur dan metode klasifikasi apa yang paling baik dalam menghasilkan nilai akurasi tertinggi.

3.4. Alur Penelitian

Adapun untuk alur penelitiannya adalah sebagai berikut



Gambar 3.2 Alur Penelitian

Adapun alur penelitian ini dapat dilihat pada gambar 3.2. Penelitian ini dimulai dengan pengumpulan data update status pada sosial media facebook. Dan dalam penelitian ini data update status yang digunakan adalah data yang telah dikumpulkan oleh penelitian sebelumnya. Kemudian setelah mendapatkan data update status facebook peneliti melakukan teks *preprocessing*. Teks preprocessing yang dilakukan adalah tokenizing, stemming dan stopword removal. Selesai dilakukan preprocessing teks adalah proses pembobotan kata dengan TF-IDF dan Count Vector. Kemudian setelah itu baru diterapkan sebuah metode untuk mencari metode yang paling tepat dalam menyelesaikan penelitian ini dengan dilihat pada nilai akurasi yang paling tinggi.

BAB IV

HASIL PENELITIAN DAN PEMBAHASAN

4.1 Gambaran Umum Penelitian

Dalam penelitian ini data yang akan diteliti adalah data teks yang diambil dari media sosial facebook. Data tersebut berjumlah 1000 dataset yang dibagi menjadi dua bagian dengan prosentase 80:20 dimana 80% merupakan dataste latih dan 20% merupakan dataset uji. Data yang dijadikan untuk datasset tersebut merupakan data dengan Bahasa Indonesia. Kemudian setelah data terkumpul data tersebut akan di lakukan tahap preprocessing yang meliputi *case folding, tokenizing, stemming dan stopword removal*. Hasil dari tahap tersebut dilanjutkan menuju tahap pembobotan kata. Pada pembobotan kata, ekstraksi fitur yang digunakan adalah TF-IDF dan Count Vector. Setelah selesai baru dilakukan tahap pengklasifikasian. Di dalam tahap pengklasifikasian ini terdapat tiga metode yang akan diujikan yaitu metode *Naïve Bayes, K-Nearest Neigbour, dan Support Vector Machine*. Sehingga hasilnya akan diketahui bahwa pasangan rekayasa fitur dan metode apa yang menghasilkan nilai akurasi tertinggi.

4.2 Pengumpulan Data

Pada penelitian ini data yang digunakan untuk penelitian adalah data update status facebook yang telah dikumpulkan oleh peneliti sebelumnya. Data yang terkumpul berjumlah 1000 data yang selanjutnya di bagi menjadi dataset uji dan dataset latih oleh peneliti. Dataset tersebut kemudian dipindahkan ke microsoft excel untuk selanjutnya diubah menjadi CSV. Khusus untuk dataset latih penulisan jenisnya emosinya menggunakan tanda pemisah default dari

computer masing-masing, dikarenakan jika berbeda bisa saja menjadi pemicu program tidak mau berjalan. Untuk dataset latih dan dataset uji dapat dilihat pada gambar dibawah ini .

Gambar 4.1 Contoh Dataset Latih

Gambar 4.2 Contoh Dataset Uji

4.3 Preprocessing teks

Proses selanjutnya yang perlu dilakukan adalah preprocessing teks. Preprocessing teks ini berfungsi untuk mengubah data mentah menjadi data yang lebih baik sehingga siap untuk diproses selanjutnya.

a. Case Folding

Proses awal dari tahap preprocessing teks adalah case folding. Dimana di dalam case folding ini terdapat beberapa tahapan. Dari yang pertama adalah tahap pengubahan huruf pada kata menjadi lowercase kemudian tahap penghapusan spasi kosong dan yang terakhir adalah

menghapus tanda baca. Code phyton yang digunakan untuk proses lower case, dapat dilihat pada gambar 4.3.

```
#lowercase
train_X = train_csv['0'].str.lower()
train_y = train_csv['1'].str.lower()
test_X = test_csv['0'].str.lower()
test_y = test_csv['1'].str.lower()
```

Gambar 4.3 Code Phyton untuk Lower Case

Code diatas menerangkan bahwa dataset yang dibagi menjadi dua menjadi x dan y diberlakukan code lower() yang berarti adalah lowercase atau membuat huruf menjadi kecil semua. Contoh implementasi dari penerapan lowercase dalam suatu kalimat dapat dilihat pada gambar 4.4 dan 4.5.

```
sentence = "Masih ada beberapa Provinsi di Negara Indonesia yang menerapkan PSBB untuk menangani virus Corona."
```

Gambar 4.4 Contoh kalimat sebelum di terapkan lowercase

```
In [5]: runfile('C:/Users/user/untitled0.py', wdir='C:/Users/
user')
masih ada beberapa provinsi di negara indonesia yang
menerapkan psbb untuk menangani virus corona.
```

Gambar 4.5 Contoh kalimat setelah diterapkan lowercase

Setelah semua huruf pada kata yang ada berubah menjadi kecil semua maka akan dilakukan proses penghapusan spasi kosong. Berikut adalah code phyton untuk menghapus spasi kosong.

```
#menghapus white space
for x in range(len(train_X)):
    train_X[x] = re.sub(r'[^A-Za-z0-9 -]', ' ', train_X[x], flags = re.IGNORECAS
    train_X[x] = train_X[x].strip()

for x in range(len(train_y)):
    train_y[x] = re.sub(r'[^A-Za-z0-9 -]', ' ', train_y[x], flags = re.IGNORECAS
    train_y[x] = train_y[x].strip()

for x in range(len(test_X)):
    test_X[x] = re.sub(r'[^A-Za-z0-9 -]', ' ', test_X[x], flags = re.IGNORECASE|
    test_X[x] = test_X[x].strip()

for x in range(len(test_y)):
    test_y[x] = re.sub(r'[^A-Za-z0-9 -]', ' ', test_y[x], flags = re.IGNORECASE|
    test_y[x] = test_y[x].strip()
```

Gambar 4.6 Code Phyton untuk Menghapus Spasi Kosong

Dalam code diatas dapat dilihat bahwa terdapat dua fungsi yang dijalankan, yaitu yang pertama adalah mengganti karakter yang tidak sesuai dengan semua huruf dan semua digit dengan spasi. Kedua adalah menghilangkan spasi di awal kalimat dan diakhir kalimat dengan perintah strip(). Contoh penerapan penghapusan spasi diawal dan akhir kalimat dapat dilihat pada gambar 4.7 dan 4.8.

```
sentence = "  /Masih ada beberapa Provinsi di Negara Indonesia yang menerapkan PSBB untuk menangani virus Corona.
```

Gambar 4.7 Contoh kalimat sebelum di terapkan strip()

Gambar diatas memperlihatkan kalimat yang di awal dan di akhir kalimat terdapat spasi kosong yang nantinya apabila dibiarkan bisa berpengaruh pada perhitungan TF-IDF.

```
In [13]: runfile('C:/Users/user/untitled0.py', wdir='C:/
Users/user')
/Masih ada beberapa Provinsi di Negara Indonesia yang
menerapkan PSBB untuk menangani virus Corona.
```

Gambar 4.8 Contoh kalimat setelah diterapkan code strip()

Pada gambar diatas dapat dilihat bahwa spasi diawal kalimat dan diakhir kalimat telah hilang setelah di terapkan code strip(). Kemudian

setelah spasi kosong terhapus masih terdapat tanda baca yang bukan merupakan komponen kata yang akan diujikan. Maka masih terdapat proses lanjutan yaitu menghapus tanda baca. Berikut adalah code phyton untuk menghapus tanda baca.

```
#menghilangkan tanda baca
train_X = train_X.str.translate(str.maketrans("","",string.punctuation))
train_y = train_y.str.translate(str.maketrans("","",string.punctuation))
test_X = test_X.str.translate(str.maketrans("","",string.punctuation))
test_y = test_y.str.translate(str.maketrans("","",string.punctuation))
```

Gambar 4.9 Code Phyton untuk Menghapus Tanda Baca

Code phyton diatas menunjukkan baliwa dalam menghapus tanda baca pada phyton menggunakan fungsi .translate() yang akan memetakan teks kedalam karakter yang terdapat pada tabel .maketrans() untuk mengganti string.punctuation() yang berupa tanda baca menjadi "". Contoh penerapan penghapusan tanda baca pada kalimat dapat dilihat pada gambar 4.10 dan 4.11.

```
import string, re

sentence = "Masih ada (beberapa) Provinsi di Negara Indonesia, yang menerapkan PSBB untuk menangani virus Corona.!!"
```

Gambar 4.10 Contoh kalimat yang masih terdapat tanda baca

```
In [9]: runfile('C:/Users/user/untitled0.py', wdir='C:/
Users/user')
Masih ada beberapa Provinsi di Negara Indonesia yang
menerapkan PSBB untuk menangani virus Corona
```

Gambar 4.11 Hasil penghapusan tanda baca

b. **Tokenizing**

Tahap selanjutnya setelah selesai semua preprocessing pada *case folding* adalah tokenizing. Tokenizing bertujuan untuk memisakan teks menjadi potongan-potongan yang disebut token agar dapat di analisa. Semua yang terkandung di dalam suatu kalimat dapat disebut

token baik berupa kata, angka, tanda baca, symbol dan entitas. Di dalam NLP, token berarti "kata" akan tetapi meskipun demikian fungsi tokenizing dalam phyton ini juga dapat digunakan untuk memisakan paragraph menjadi kalimat. Tokenizing ini merupakan salah satu proses preprocessing yang penting untuk diterapkan, karena apabila tidak berhasil menerapkan nantinya akan berbeda hasil akurasinya. Yang akan digunakan oleh peneliti disini adalah tokenizing pada kata dengan menggunakan modul NLTK. Berikut code phyton untuk tokenizing.

```
#Tokenizing
for k in range(len(train_X)):
    train_X[k] = nltk.tokenize.word_tokenize(train_X[k])

for k in range (len(test_X)):
    test_X[k] = nltk.tokenize.word_tokenize(test_X[k])

# create stemmer
```

Gambar 4.12 Code phyton untuk tokenizing

Dari code diatas dapat kita lihat bahwa penulis menggunakan module nltk dalam menerapkan metode tokenizing. Sebenarnya terdapat metode lain dalam menerapkan fungsi tokenizing ini yaitu dengan code split() akan tetapi apabila terdapat 2 kata yang sama nantinya akan diartikan dengan 2 entitas yang berbeda, maka itulah penulis menggunakan module nltk. Contoh implementasi fungsi tokenizing dalam suatu kalimat dapat dilihat pada gambar 4.13 dan 4.14.

```
import string, re
import nltk

from nltk.tokenize import word_tokenize

sentence = "Warih ada beberapa Provinsi di Negara Indonesia yang menerapkan PSBB untuk menangani virus Corona"
```

Gambar 4.13 Contoh kalimat sebelum diterapkan tokenizing

```
In [15]: runfile('C:/Users/user/untitled0.py', wdir='C:/
Users/user')
['Masih', 'ada', 'beberapa', 'Provinsi', 'di', 'Negara',
'Indonesia', 'yang', 'menerapkan', 'PSBB', 'untuk',
'menangani', 'virus', 'Corona']
```

Gambar 4.14 Contoh kalimat hasil tokenizing

Dalam contoh kalimat hasil dari tokenizing diatas masih terdapat beberapa huruf yang besar karena fungsi tokenizing hanya memecah kalimat menjadi kata saja sehingga akan lebih baik apabila penggunaan tokenizing ini diterapkan setelah proses case folding agar tanda baca dan huruf yang masih besar bisa dikecilkan sehingga tidak mengganggu pada proses perhitungan dengan algoritma.

c. **Stemming**

Setelah melalui proses lower case dan tokenizing adalah proses stemming. Proses stemming bertujuan untuk mengubah kata menjadi kata dasarnya. Jadi dalam proses stemming ini terdapat beberapa imbuhan yang harus dihilangkan baik itu imbuhan prefix, sufiks, maupun konfiks. Dalam penelitian ini proses stemming menggunakan phyton sastrawi. Berikut adalah code stemming dengan menggunakan phyton.

```
# create stemmer
factoryStemm = StemmerFactory()
stemmer = factoryStemm.create_stemmer()
for y in range(len(train_X)):
    train_X[y] = stemmer.stem(train_X[y])

for y in range(len(test_X)):
    test_X[y] = stemmer.stem(test_X[y])
```

Gambar 4.15 Code Phyton untuk Stemming

Dalam hal stemming bahasa Indonesia ini peneliti menggunakan library phyton sastrawi yang sudah disiapkan pada awal code. Berikut contoh implementasi penggunaan stemming bahasa Indonesia.

```
sentence = "Masih ada beberapa Provinsi di Negara Indonesia yang menerapkan PSBB untuk menangani virus Corona"
```

Gambar 4.16 Contoh kalimat sebelum diterapkan stemming

```
In [21]: runfile('C:/Users/user/untitled0.py', wdir='C:/
Users/user')
masih ada beberapa provinsi di negara indonesia yang
terap psbb untuk tangan virus corona
```

Gambar 4.17 Contoh kalimat hasil dari stemming

Dalam hasil stemming diatas dapat dilihat bahwa terdapat imbuhan yang dihilangkan pada kata menerapkan menjadi terap dan pada kata menangani menjadi tangan.

d. Stop Word Removal

Selanjutnya proses terakhir adalah penghapusan stopword pada data update status yang tersisa. Jika terdapat kata yang termasuk dalam list stopword maka akan otomatis dibuang. Daftar kata yang ada pada stopword sastrawi bisa ditambah maupun di kurangi sesuai dengan kebutuhan penelitian masing-masing. Berikut adalah contoh kata yang masuk dalam daftar stopword pada phyton sastrawi.

```
def get_stop_words(self):
    return ['yang', 'untuk', 'pada', 'ke', 'para', 'namun', 'menurut', 'antara', 'dia', 'dua',
        'ia', 'seperti', 'jika', 'jika', 'sehingga', 'kembali', 'dan', 'tidak', 'ini', 'karena',
        'kepada', 'oleh', 'saat', 'harus', 'sementara', 'setelah', 'belum', 'kami', 'sekitar',
        'bagi', 'serta', 'di', 'dari', 'telah', 'sebagai', 'masih', 'hal', 'ketika', 'adalah',
        'itu', 'dalam', 'bisa', 'bahwa', 'atau', 'hanya', 'kita', 'dengan', 'akan', 'juga',
        'ada', 'mereka', 'sudah', 'saya', 'terhadap', 'secara', 'agar', 'lain', 'anda',
        'begitu', 'mengapa', 'kenapa', 'yaitu', 'yakni', 'daripada', 'itulah', 'lagi', 'maka',
        'tentang', 'dari', 'dimana', 'kemana', 'pula', 'sambil', 'sebelum', 'sesudah', 'supaya',
        'guna', 'kah', 'pun', 'sampai', 'sedangkan', 'selagi', 'sementara', 'tetapi', 'apakah',
        'kecuali', 'sebab', 'selain', 'seolah', 'seraya', 'seterusnya', 'tanpa', 'agak', 'boleh',
        'dapat', 'dsb', 'dst', 'dll', 'dahulu', 'dulunya', 'anu', 'demikian', 'tapi', 'ingin',
        'juga', 'nggak', 'mari', 'nanti', 'melainkan', 'oh', 'ok', 'seharusnya', 'sebetulnya',
        'setiap', 'setidaknya', 'sesuatu', 'pasti', 'saja', 'toh', 'ya', 'walau', 'tolong',
        'tentu', 'amat', 'apalagi', 'bagaimanapun']
```

Gambar 4.18 Daftar Kata Pada Phyton Sastrawi

Setelah daftar tersebut sesuai kebutuhan penelitian, code stopwordnya bisa langsung dijalankan. Berikut code untuk stopword pada phyton.

```
#stop Removal Sastrawi
factoryStopRemove = StopWordRemoverFactory()
stopword = factoryStopRemove.create_stop_word_remover()
for z in range(len(train_X)):
    train_X[z] = stopword.remove(train_X[z])

for z in range(len(test_X)):
    test_X[z] = stopword.remove(test_X[z])
```

Gambar 4.19 Code Stopword pada Phyton

Code diatas menampilkan proses stopword dengan menggunakan library phyton sastrawi. Library phyton sastrawi tersebut selain digunakan untuk stemming pada tahap preproessing sebelumnya juga dapat digunakan untuk *stopword removal* atau *filtering*. Berikut adalah contoh implementasi filtering dengan menggunakan library phyton sastrawi.

```
sentence = "Masih ada beberapa Provinsi di Negara Indonesia yang menerapkan PSBB untuk menangani virus Corona"
```

Gambar 4.20 Contoh kalimat sebelum diterapkan filtering

```
In [19]: runfile('C:/Users/user/untitled0.py', wdir='C:/
Users/user')
Masih beberapa Provinsi Negara Indonesia menerapkan PSBB
menangani virus Corona
```

Gambar 4.21 Contoh kalimat hasil filtering

Dapat dilihat pada contoh kalimat hasil dari penerapan stopword removal atau biasa disebut filtering ini bahwa terdapat beberapa kata yang dianggap mempunyai arti yang rendah dan masuk ke dalam stoplist sehingga tidak muncul kembali, kata tersebut yang hilang adalah ada, di, yang dan untuk

4.4 Ekstraksi Fitur

Setelah proses preprocessing selesai, proses selanjutnya adalah ektraksi fitur. Pada penelitian ini ekstraksi fitur yang digunakan adalah *Term Frequency – Invers Document Frequency* (TF_IDF) dan Count Vector. Jenis pembobotan yang dilakukan adalah Count vector, TF-IDF word level , TF-IDF character level, TF IDF N-gram level. Dalam penelitian ini *feature* TF-IDF dan count vector yang digunakan adalah *library scikit-learn* atau biasa disebut dengan sklearn. Library sklearn merupakan library yang mendukung *machine learning* dan mempunyai berbagai feature untuk mendukung algoritma *classification*, *regression*, dan *clustering*. Berikut code phyton untuk Count Vector.

```
# ----------------------------------------------
print('---------------Time Extraction Count Vector---------------')
t = time()  # not compulsory

# loading CountVectorizer
tf_vectorizer = CountVectorizer() # or term frequency

X_train_tf = tf_vectorizer.fit_transform(train_X)

duration = time() - t
print("Time taken to extract features from training data : %f seconds" % (duration))
print("n_samples: %d, n_features: %d" % X_train_tf.shape)

t = time()
X_test_tf = tf_vectorizer.transform(test_X)

duration = time() - t
print("Time taken to extract features from test data : %f seconds" % (duration))
print("n_samples: %d, n_features: %d" % X_test_tf.shape)
```

Gambar 4.22 Code Phyton untuk Count Vector

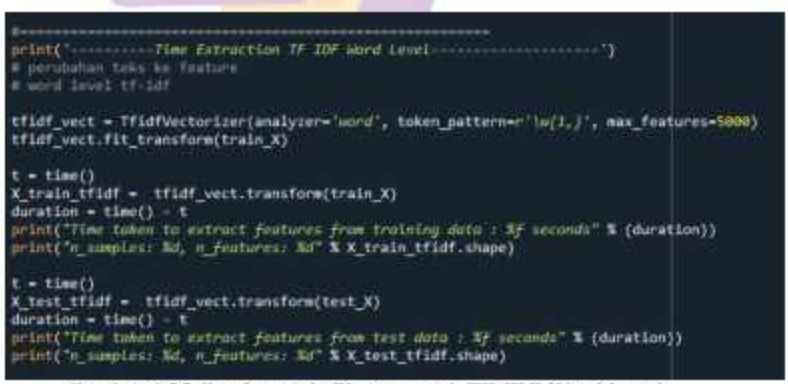
Count Vector adalah notasi matrik dimana setiap baris mewakili dokumen dari korpus (dataset latih), setiap kolom mewakili istilah dari korpus (dataset latih) dan setiap sel mewakili jumlah frekuensi dari istilah tertentu dalam dokumen tertentu. Selanjutnya adalah code phyton untuk TF-IDF Ngram level. Berikut code phyton untuk TF-IDF N-gram level.

Gambar 4.23 Code Phyton untuk TF-IDF N-gram Level

Skor TF-IDF mewakili kepentingan relative suatu istilah dalam dokumen dan seluruh korpus (dataset latih). Skor TF-IDF terdiri dari dua term : yang pertama menghitung Normalized Term Frequency (TF), yang kedua adalah Inverse Documen Frequency (IDF), dihitung sebagai logaritma jumlah dokumen dalam korpus dibagi dengan angka dokumen dimana istilah spesifik muncul. Vektor TF-IDF dapat dibuat pada berbagai level token input kata, karakter dan n-gram. Yang pertama yang akan kita bahas adalah N-gram. N-gram adalah kombinasi dari N suku Bersama. Matriks ini mereprentasikan skor TF-IDF dari N-gram. Selanjutnya adalah code untuk TF-IDF Character Level. Berikut adalah code phytonnya.

Gambar 4.24 Gambar Code Phyton untuk TF-IDF Char Level.

Dari gambar diatas dapat dilihat bahwa TF-IDF yang sedang digunakan adalah char. Hal tersebut dapat dilihat dari code jenis analyzernya. Kemudian *n-gram_range* digunakan untuk mengontrol token apa yang dihasilkan , dalam code tersebut tertulis (2,3) yang artinya pada TF-IDF char level ini akan menghasilkan token dua kata dan tiga kata. Selanjutnya terdapat *max_features = 5000* berarti bahwa batas kosakata untuk fitur yang paling sering terjadi sebesar 5000 yang mengasumsikan bahwa istilah yang paling sering adalah yang paling penting. Ekstraksi fitur yang selanjutnya adalah TF-IDF Word Level. Berikut adalah code phytonnya.



Gambar 4.25 Gambar code Phyton untuk TF-IDF Word Level.

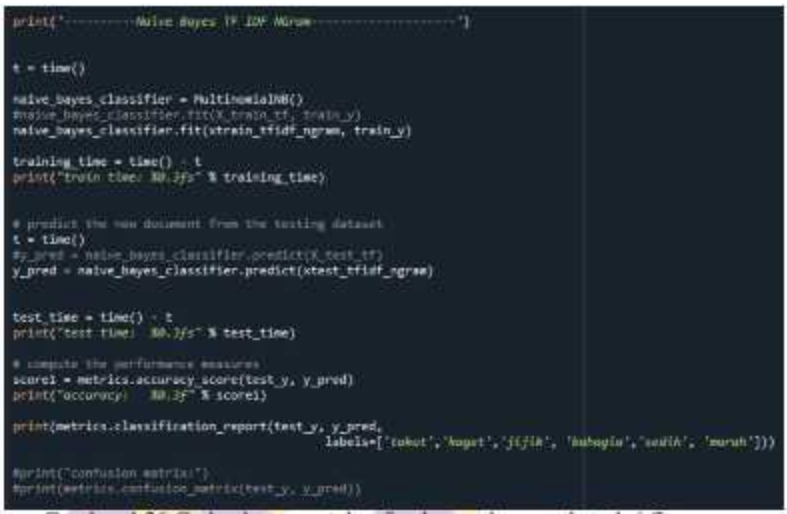
Apabila diperhatikan dengan seksama code phyton pada TF-IDF word level hamper sama dengan code TF-IDF N-gram level hanya saja pada Word level tidak menggunakan range untuk mengontrol token yang dihasilkan. TF-IDF Word Level adalah matriks yang merepresentasikan skor tf-idf pada setiap istilah dalam dokumen yang berbeda.

4.5 Klasifikasi

Setelah selesai pada tahap ekstraksi fitur , tahap selanjutnya adalah tahap klasifikasi. Penelitian ini menggunakan klasifikasi naive bayes, support vector machine, dan k-nearest neigbour. Selanjutnya ketiga metode klasfikasi tersebut akan disandingkan satu per satu dengan ekstraksi fitur count vector, tf-idf n-gram level, tf-idf character level dan tf-idf word level. Kemudian akan dijabarkan code phyton dari masing-masing pemasangan fitur ekstraski dan metode klasifikasi tersebut.

a. Naïve Bayes

Yang pertama adalah Naïve Bayes. Jenis algoritma naïve bayes yang digunakan adalah multinominal naïve bayes. Berikut adalah code phyton untuk metode naïve bayes dengan ekstraksi fitur n-gram level.



```
print('----------Naive Bayes TF IDF NGram--------------------')


t = time()

naive_bayes_classifier = MultinomialNB()
#naive_bayes_classifier.fit(X_train_tf, train_y)
naive_bayes_classifier.fit(xtrain_tfidf_ngram, train_y)

training_time = time() - t
print("train time: %0.3fs" % training_time)


# predict the new document from the testing dataset
t = time()
#y_pred = naive_bayes_classifier.predict(X_test_tf)
y_pred = naive_bayes_classifier.predict(xtest_tfidf_ngram)


test_time = time() - t
print("test time:  %0.3fs" % test_time)

# compute the performance measures
score1 = metrics.accuracy_score(test_y, y_pred)
print("accuracy:   %0.3f" % score1)

print(metrics.classification_report(test_y, y_pred,
                                    labels=['sukut','bagat','jijik', 'bahagia','sedih', 'marah']))

#print("confusion matrix:")
#print(metrics.confusion_matrix(test_y, y_pred))
```

Gambar 4.26 Code phyton untuk naïve bayes dengan ekstraksi fitur tf-idf n-gram level

Selanjutnya adalah code phyton untuk naïve bayes dengan ekstraksi fitur tf-idf character level.

```
print('----------Naive Bayes TF IDF Char Level--------------------')


t = time()

naive_bayes_classifier = MultinomialNB()
#naive_bayes_classifier.fit(X_train_tf, train_y)
naive_bayes_classifier.fit(xtrain_tfidf_ngram_chars, train_y)

training_time = time() - t
print("train time: %0.3fs" % training_time)


# predict the new document from the testing dataset
t = time()
#y_pred = naive_bayes_classifier.predict(X_test_tf)
y_pred = naive_bayes_classifier.predict(xtest_tfidf_ngram_chars)


test_time = time() - t
print("test time:  %0.3fs" % test_time)

# compute the performance measures
score1 = metrics.accuracy_score(test_y, y_pred)
print("accuracy:   %0.3f" % score1)

print(metrics.classification_report(test_y, y_pred,
                                    labels=['sukut','bagat','jijik', 'bahagia','sedih', 'marah']))

#print("confusion matrix:")
#print(metrics.confusion_matrix(test_y, y_pred))
```

Gambar 4.27 Code phyton untuk naïve bayes dengan ekstraksi fitur Tf-idf character level

Yang ketiga adalah code phyton untuk naïve bayes dengan ekstraksi fitur tf-idf word level.

```
print('----------Naive Bayes TF IDF Word Level--------------------')


t = time()

naive_bayes_classifier = MultinomialNB()
#naive_bayes_classifier.fit(X_train_tf, train_y)
naive_bayes_classifier.fit(X_train_tfidf, train_y)

training_time = time() - t
print("Train time: %0.3fs" % training_time)


# predict the new document from the testing dataset
t = time()
#y_pred = naive_bayes_classifier.predict(X_test_tf)
y_pred = naive_bayes_classifier.predict(X_test_tfidf)


test_time = time() - t
print("Test time:  %0.3fs" % test_time)

# compute the performance measures
score1 = metrics.accuracy_score(test_y, y_pred)
print("accuracy:   %0.3f" % score1)

print(metrics.classification_report(test_y, y_pred,
                                    labels=['tunur','kuper','jijik', 'bahagia','sedih', 'marah']))

#print("confusion matrix:")
#print(metrics.confusion_matrix(test_y, y_pred))
```

Gambar 4.28 Code phyton untuk naïve bayes dengan ekstraksi fitur tf-idf word level.

Keempat adalah code phyton untuk naïve bayes dengan ekstraksi fitur count vector

```
print('----------------Naive Bayes Count Vector----------------')

t = time()

naive_bayes_classifier = MultinomialNB()
naive_bayes_classifier.fit(X_train_tf, train_y)
#naive_bayes_classifier.fit(X_train_tfidf, train_y)

training_time = time() - t
print("Train time: %0.3fs" % training_time)


# predict the new document from the testing dataset
t = time()
y_pred = naive_bayes_classifier.predict(X_test_tf)
#y_pred = naive_bayes_classifier.predict(X_test_tfidf)


test_time = time() - t
print("Test time:  %0.3fs" % test_time)

# compute the performance measures
score1 = metrics.accuracy_score(test_y, y_pred)
print("accuracy:   %0.3f" % score1)

print(metrics.classification_report(test_y, y_pred,
                                    labels=['tunur','kuper','jijik', 'bahagia','sedih', 'marah']))

#print("confusion matrix:")
#print(metrics.confusion_matrix(test_y, y_pred))
```

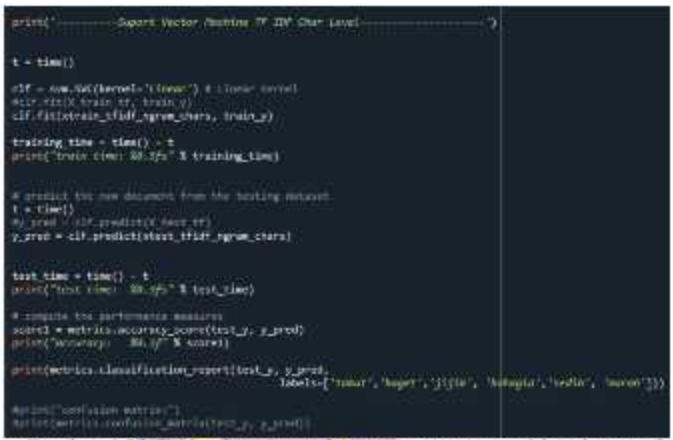
Gambar 4.29. Code phyton untuk naïve bayes dengan ekstraksi fitur count vector.

b. Support Vector Machine

Selanjutnya untuk klasifikasi kedua yang digunakan dalam penelitian ini adalah suport vector machine. Berikut code untuk klasifikasi suport vector machine dengan ekstraksi fitur tf-idf n-gram level.

Gambar 4.30 Code phyton untuk support vector machine dengan fitur ekstraksi tf-idf n-gram level.

Selanjutnya adalah code phyton untuk klasifikasi support vector machine dengan ekstraksi fitur tf-idf character level.



Gambar 4.31 Code phyton untuk support vector machine dengan fitur ekstraksi tf-idf character level.

Yang ketiga adalah code phyton untuk support vector machine dengan ekstraksi fitur tf-idf word level.

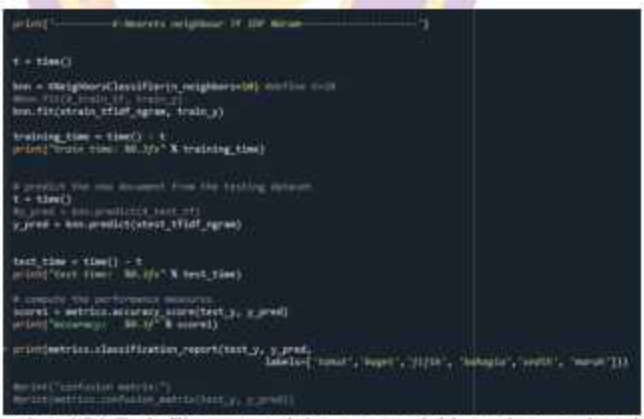
Gambar 4.32 Code phyton untuk support vector machine dengan fitur ekstraksi tf-idf word level.

Yang keempat adalah code phyton untuk klasifikasi support vector machine dengan ekstraksi fitur count vector.

Gambar 4.33 Code phyton untuk support vector machine dengan fitur ekstraksi count vector

c. K-Nearest Neigbour

Kemudian yang terakhir klasifikasi yang digunakan adalah k-nearest neighbour. Berikut code phyton untuk klasifikasi k-nearest neigbour dengan ekstraksi fitur tf-idf n-gram level.

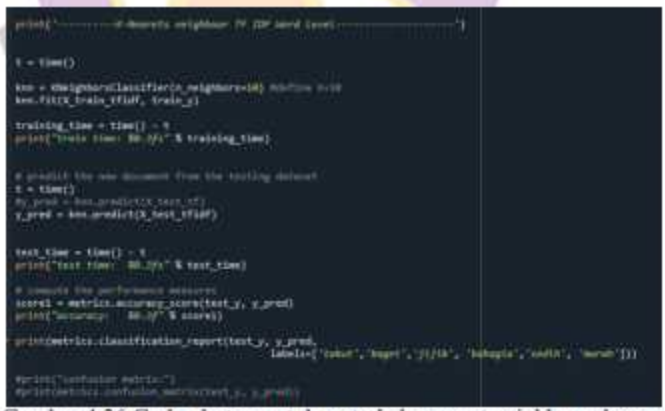


Gambar 4.34 Code Phyton untuk k-nearest neighbour dengan ekstraksi fitur tf-idf n-gram level.

Selanjutnya adalah code phyton untuk metode k-nearest neighbour dengan fitur tf-idf character level.

Gambar 4.35 Code phyton untuk metode k-nearest neighbour dengan fitur tf-idf char level.

Kemudian yang ketiga adalah code phyton untuk metode k-nearest neighbour dengan ekstraksi fitur tf-idf word level.



Gambar 4.36 Code phyton untuk metode k-nearest neighbour dengan ekstraksi fitur tf-idf word level.

Dan yang terakhir dari metode klasifikasi ini adalah code phyton untuk metode k-nearest neighbour dengan ekstraksi fitur count vector.

Gambar 4.37 Code phyton untuk metode k-nearest neighbour dengan ekstraksi fitur count vector.

4.6 Uji Coba dan Evaluasi

a. Lingkungan Uji Coba

Lingkungan uji coba dalam penelitian ini berupa perangkat keras dan perangkat lunak yang digunakan untuk melakukan uji coba text mining pada social media facebook untuk mendeteksi emosi pengguna. Lingkungam uji coba ini berupa komputer dengan spesifikasi sebagai berikut :

Perangkat	Spesifikasi
Perangkat Keras	Processor AMD A9-9425 APU (Up to 3.1GHz) Memory 4GB DDR4-2133, HDD 1TB
Perangkat Lunak	Sistem Operasi : Microsoft Windows 10 Pro Perangkat Pengembang : Anaconda 3 Perangkat Pembantu : Spyder, Microsoft Excell 2016 Microsoft Word 2016

b. Skenario Uji Coba

Pada subbab ini akan dijelaskan scenario uji coba yang telah dilakukan oleh peneliti. Terdapat beberapa scenario uji coba yang berhasil dilakukan diantaranya :

1. Perhitungan hasil akurasi, recall, precision dan fscore tertinggi pada metode naïve bayes dengan keempat ekstraksi fitur yang digunakan.
2. Perhitungan hasil akurasi, recall, precision dan fscore tertinggi pada metode support vector machine dengan keempat ekstraksi fitur yang digunakan.
3. Perhitungan hasil akurasi, recall, precision dan fscore tertinggi pada metode k-nearest neigbour dengan keempat ekstraksi fitur yang digunakan.

c. Pembagian Data Klasifikasi

Pada penelitian ini data yang didapat dari peneliti sebelumnya berjumlah 1000 data update status bahasa Indonesia. Kemudian data tersebut dibagi menjadi dua bagian yaitu data latih dan data uji dengan masing-masing terdapat variasi prosentasi yang bertujuan untuk mengetahui nilai prosesntase yang menghasilkan nilai akurasi tertinggi. Berikut jumlah klasifikasi data yang didapatkan oleh peneliti berdasarkan kelas emosi.

Tabel 4.1 Jumlah klasifikasi data keseluruhan

Kelas Emosi	Jumlah
Marah	65
Bahagia	406
Sedih	211
Kaget	32
Takut	46
Jijik	40

d. Skenario 1 Perhitungan Hasil Akurasi, Recall, Precision, dan Fscore pada metode Naïve Bayes dengan empat ekstraksi fitur.

Skenario pertama dalam penelitian ini dilakukan untuk menghasilkan nilai akurasi, recall, precision dan fscore pada metode naïve bayes dengan di padukan dengan empat ekstraksi fitur secara bergantian. Dengan demikian dalam skenario ini akan dilakukan sebanyak 4 kali perhitungan. Dari keempat perhitungan tersebut nantinya akan dievaluasi mana yang menghasilkan nilai akurasi, recall, precision dan fscore. Berikut adalah tabel hasil perhitungan yang telah dilakukan.

Tabel 4.2 Hasil perhitungan akurasi, recall, precision, fscore dari metode naïve bayes dengan 4 ekstraksi fitur.

Nama	Akurasi	Emosi	precision	Recall	F1 score
count vector	60%	Takut	86%	32%	46%
		Kaget	100%	67%	80%
		Jijik	100%	83%	91%
		Bahagia	61%	84%	70%
		Sedih	50%	41%	45%
		Marah	44%	18%	26%
TF-IDF Word Level	51%	Takut	0%	0%	0%
		Kaget	0%	0%	0%
		Jijik	0%	0%	0%
		Bahagia	49%	98%	65%
		Sedih	80%	16%	26%
		Marah	0%	0%	0%

Tabel 4.2 Hasil perhitungan akurasi, recall, precision, fscore dari metode naïve bayes dengan 4 ekstraksi fitur (Lanjutan)

Nama	Akurasi	Emosi	precision	Recall	F1 score
TF-IDF Ngram level	50%	Takut	0%	0%	0%
		Kaget	0%	0%	0%
		Jijik	0%	0%	0%
		Bahagia	49%	100%	66%
		Sedih	100%	6%	11%
		Marah	100%	5%	9%
TF-IDF Char level	48%	Takut	0%	0%	0%
		Kaget	0%	0%	0%
		Jijik	0%	0%	0%
		Bahagia	48%	100%	65%
		Sedih	0%	0%	0%
		Marah	0%	0%	0%

Dari tabel diatas dapat dilihat bahwa pada skenario pertama ini nilai akurasi tertinggi terdapat pada percobaan antara metode klasifikasi naïve bayes dengan ekstraksi fitur count vector dengan nilai 60%. Dan untuk nilai terendah terdapat pada perhitungan metode naïve bayes dengan ekstraksi fitur tf-idf character level dengan nilai 48%.

e. Skenario 2 Perhitungan Hasil Akurasi, Recall, Precision dan Fscore pada metode Suport Vector Machine dengan empat ekstrasksi fitur.

Selanjutnya untuk skenario yang kedua adalah perhitungan untuk menghasilkan nilai akurasi, recall, precision dan fscore pada metode support vector machine dengan keempat ekstraksi fitur yang digunakan. Sama halnya dengan skenario penelitian yang pertama, pada skenario ini pun dilakukan perhitungan sebanyak empat kali terhadapat dataset dan

data latih yang sama. Akan tetapi yang berbeda adalah model ekstraksi fitur yang digunakan. Berikut adalah tabel hasil perhitungan yang telah dilakukan.

Tabel 4.3 Hasil perhitungan akurasi, recall, precision, fscore dari metode Support vector machine dengan 4 ekstraksi fitur.

Nama	Akurasi	Emosi	precision	Recall	F1 score
count vector	57%	Takut	62%	42%	50%
		Kaget	67%	67%	67%
		Jijik	55%	100%	71%
		Bahagia	59%	79%	68%
		Sedih	48%	31%	38%
		Marah	50%	18%	27%
TF-IDF Word Level	56%	Takut	100%	21%	35%
		Kaget	100%	17%	29%
		Jijik	100%	33%	50%
		Bahagia	55%	91%	68%
		Sedih	52%	31%	39%
		Marah	75%	14%	23%
TF-IDF Ngram level	57%	Takut	100%	32%	48%
		Kaget	100%	50%	67%
		Jijik	100%	83%	91%
		Bahagia	53%	100%	69%
		Sedih	100%	6%	11%
		Marah	100%	9%	17%
TF-IDF Char level	56%	Takut	100%	5%	10%
		Kaget	100%	17%	29%
		Jijik	100%	33%	50%
		Bahagia	55%	93%	69%
		Sedih	51%	35%	42%
		Marah	100%	5%	9%

Pada skenario kedua ini nilai akurasi tertinggi dihasilkan oleh pasangan metode support vector machine dengan ektraksi fitur count vector dan tf-idf ngram level dengan nilai 57%. Dan dua pasang yang lain yaitu

anatar metode support vector machine dengan tf-idf char level dan tf-idf word level menduduki nilai akurasi terendah dengan nilai 56%.

f. Skenario 3 Perhitungan Hasil Akurasi, Recall, Precision dan Fscore pada metode K-Nearest Neloghbour dengan empat ekstraksi fitur.

Sama halnya dengan skenario pertama dan kedua. Pada skenario ketiga ini juga menghasilkan nilai akurasi , recall , precision dan fscore pada metode k-nearest neighbour dengan menggunakan empat ekstraksi fitur. Berikut adalah perhitungan yang telah dilakukan.

Tabel 4.4 Hasil perhitungan akurasi, recall, precision, fscore dari metode k-nearest neighbour dengan 4 ekstraksi fitur.

Nama	Akurasi	Emosi	precision	Recall	F1 score
count vector	48%	Takut	0%	0%	0%
		Kaget	0%	0%	0%
		Jijik	0%	0%	0%
		Bahagia	49%	93%	64%
		Sedih	39%	14%	20%
		Marah	0%	0%	0%
TF-IDF Word Level	44%	Takut	0%	0%	0%
		Kaget	0%	0%	0%
		Jijik	0%	0%	0%
		Bahagia	53%	81%	64%
		Sedih	30%	22%	25%
		Marah	0%	0%	0%
TF-IDF Ngram level	46%	Takut	0%	0%	0%
		Kaget	0%	0%	0%
		Jijik	0%	0%	0%
		Bahagia	47%	97%	64%
		Sedih	0%	0%	0%
		Marah	0%	0%	0%

Tabel 4.4 Hasil perhitungan akurasi, recall, precision, fscore dari metode k-nearest neighbour dengan 4 ekstraksi fitur (Lanjutan)

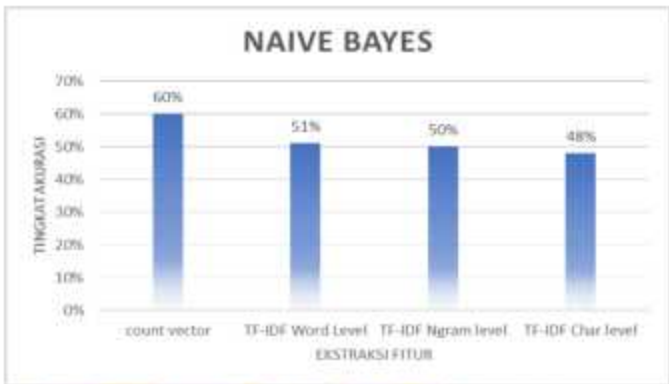
Nama	Akurasi	Emosi	precision	Recall	F1 score
TF-IDF Char level	47%	Takut	0%	0%	0%
		Kaget	0%	0%	0%
		Jijik	0%	0%	0%
		Bahagia	51%	84%	63%
		Sedih	35%	25%	30%
		Marah	0%	0%	0%

Dari skenario ketiga ini dapat kita lihat bahwa nilai akurasi terdapat pada percobaan ke pertama dengan metode k-nearest neighbour dan ekstraksi fitur count vector yang menghasilkan nilai akurasi sebesar 48%. Dan untuk nilai akurasi terendah terdapat pada percobaan ke dua yaitu metode k-nearest neighbour dengan ekstraksi fitur tf-idf word level dengan nilai 44%.

g. Evaluasi Skenario 1

Pada skenario 1 akan membahas tentang perhitungan yang telah dilakukan dengan metode klasifikasi naïve bayes. Dalam skenario yang pertama telah dilakukan perhitungan sebanyak 4 kali dimana dalam setiap perhitungan tersebut metode klasifikasi dipasangkan dengan metode fitur ekstraksi yang berbeda-beda. Dapat dilihat kembali pada tabel 4.2, dalam tabel tersebut sudah memeparkan hasil dari keempat perhitungan yang telah dilakukan. Pada nilai precision tertinggi terdapat dikelas emosi kaget dan jijik dengan nilai 100%. Sedangkan untuk nilai recall nilai tertinggi terdapat pada kelas emosi bahagia dengan nilai 84%. Kemudian nilai tertinggi pada fscore terdapat pada kelas emosi jijik dengan nilai 91%. Dan untuk hasil akurasinya juga cukup

berfariatif. Selanjutnya untuk nilai akurasi akan ditampilkan dalam bentuk diagram agar lebih mudah dipahami.



Gambar 4.38 Diagram hasil akurasi metode naïve bayes

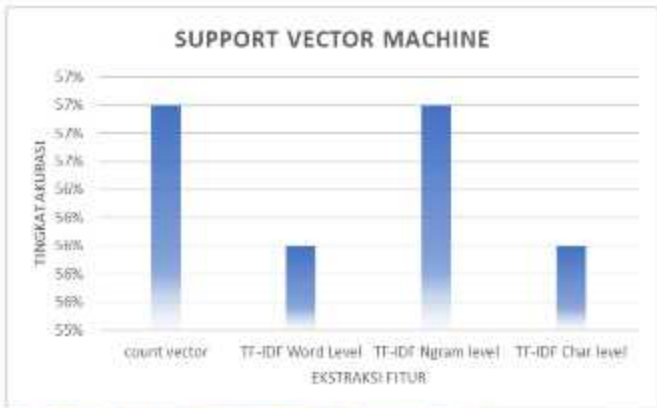
Dalam diagram diatas dapat dilihat bahwa tingkat akurasi cukup bervariatif dari 48%-60%. Tingkat akurasi tertinggi pada skenario pertama ini terdapat pada ekstraksi fitur count vector dengan nilai 60% yang berarti metode klasifikasi naïve bayes dapat menghasilkan nilai akurasi maksimal apabila menggunakan ekstraksi fitur count vector. Kemudian untuk nilai akurasi terendah adalah ekstraksi fitur tf-idf char level dengan nilai 48%. Yang berarti bahwa metode klasifikasi naïve bayes akan akan menghasilkan nilai akurasi terendah apabila ekstraksi fitur yang digunakan adalah tf-idf char level

h. Evaluasi Skenario 2

Selanjutnya untuk skenario yang kedua sama dengan skenario pertama yang berbeda hanyalah metode klasifikasi yang digunakan, yaitu metode support vector machine. Dalam skenario ini juga dilakukan perhitungan

sebanyak 4 kali dengan menggunakan metode ekstraksi fitur yang berbeda disetiap perhitungannya. Hasil dari perhitungan sudah ditampilkan di tabel 4.3. Dapat dilihat dalam tabel tersebut bahwa tidak ad akelas emosi yang memiliki nilai 0%, ini sedikit berbeda dengan hasil perhitungan pada skenario pertama. Pada perhitungan yang pertama yaitu ektraksi fitur count vector dengan klasifikasi support vector machine nilai precision paling tinggi terdapat pada kelas emosi kaget dengan nilai 67%. Sedangkan untuk nilai recall tertinggi terdapat pada kelas emosi jijik dengan nilai 100%. Kemudian untuk nilai fscore tertinggi terdapat pada kelas emosi jijik dengan nilai 71%. Perhitungan kedua menggunakan ekstraksi fitur tf-idf word level dengan metode klasifikasi support vector machine menghasilkan nilai precision tertinggi pada kelas emosi takut, kaget dan jijik dengan nilai 100%. Dan untuk nilai recall tertinggi pada kelas emosi bahagia dengan nilai 91%. Nilai fscore tertinggi terdapat pada kelas emosi bahagia dengan nilai 68%. Perhitungan ketiga menggunakan ekstraksi fitur tf-idf ngram level. Pada perhitungan ini nilai precision hampir semua kelas emosi memiliki nilai sempurna yaitu 100% hanya kelas emosi bahagia yang memiliki nilai 53%. Selanjutnya untuk recall nilai tertinggi terdapat pada kelas emosi bahagia dengan nilai 100%. Dan untuk fscore nilai tertinggi terdapat pada kelas emosi jijik dengan nilai 91%. Yang terakhir untuk perhitungan keempat dengan ekstraksi fitur tf-idf char level dengan metode klasifikasi support vector machine. Pada perhitungan ini nilai

precision tertinggi terdapat pada kelas emosi marah, takut, kaget, jijik. Kemudian untuk nilai recall tertinggi terdapat pada bahagia dengan nilai 93%. Dan untuk nilai fscore tertinggi terdapat pada kelas emosi bahagia dengan nilai 69%. Selanjutnya untuk nilai akurasi disajikan dalam bentuk diagram pada gambar 4.40.



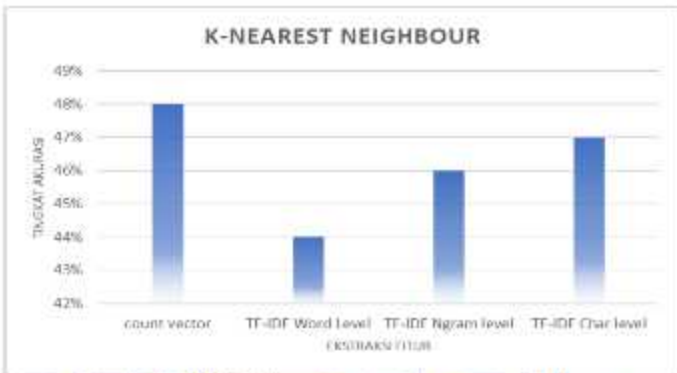
Gambar 4.39 Diagram hasil akurasi metode support vector machine

Dalam diagram tersebut dapat dilihat bahwa nilai akurasi tertinggi terdapat pada dua perhitungan yaitu dengan menggunakan ekstraksi fitur count vector dan tf-idf ngram level dipadukan dengan metode klasifikasi support vector machine dengan nilai 57%. Kemudian selanjutnya yang lain menjadi yang terendah yaitu ekstraksi fitur tf-idf word level dan tf-idf char level dengan metode klasifikasi support vector machine dengan nilai 56%.

l. Evaluasi Skenario 3

Memasuki evaluasi untuk skenario yang ketiga yaitu dengan metode klasifikasi k-nearest neighbour. Pada skenario yang ketiga ini juga sama dengan skenario yang pertama dan kedua dimana dilakukan perhitungan sebanyak 4 kali dengan menggunakan ekstraksi fitur yang berbeda. Hasil dari percobaan ini telah disajikan pada tabel 4.4. Dapat dilihat pada tabel 4.4 pada percobaan pertama dengan menggunakan ekstraksi fitur count vector nilai precision tertinggi terdapat pada kelas emosi bahagia dengan nilai 49%. Selanjutnya untuk nilai tertinggi pada recall terdapat pada kelas emosi bahagia 93%. Dan untuk nilai tertinggi pada fscore terdapat pada kelas emosi bahagia juga dengan nilai 64%. Perhitungan yang kedua dengan menggunakan ekstraksi fitur tf-idf word level dan metode klasifikasi k-nearest neighbour menghasilkan nilai precision tertinggi terdapat pada kelas emosi bahagia dengan nilai 53%. Untuk nilai recall tertinggi terdapat pada kelas emosi bahagia dengan nilai 81%. Dan untuk nilai fscore tertinggi terdapat pada kelas emosi bahagia dengan nilai 64%. Perhitungan ketiga dengan metode ekstraksi fitur ngram level dan metode klasifikasi k-nearest neighbour menghasilkan nilai precision , recall dan fscore tertinggi adalah kelas emosi bahagia. Dalam percobaan ini selain kelas emosi bahagia memiliki nilai 0% pada precision, recall maupun fscore. Kemudian perhitungan keempat yaitu ekstraksi fitur tf-idf char level dengan metode klasifikasi k-nearest neighbour menghasilkan nilai precision

terdapat pada kelas emosi bahagia dengan nilai 51%. Dan untuk nilai recall tertinggi terdapat pada kelas emosi bahagia juga dengan nilai 84%. Begitupun dengan nilai fscore tertinggi juga terdapat pada kelas emosi bahagia dengan nilai 63%. Selanjutnya untuk nilai akurasi akan ditampilkan dalam bentuk diagram pada gambar 4.41.



Gambar 4.40 Hasil akurasi metode k-nearest neighbour.

Dapat dilihat dalam diagram tersebut nilai akurasi tertinggi terdapat pada perhitungan menggunakan ekstraksi fitur count vector dan metode klasifikasi k-nearest neighbour dengan nilai 48%. Kemudian untuk nilai terendah terdapat pada perhitungan menggunakan ekstraksi fitur tf-idf word level dan metode klasifikasi k-nearest neighbour dengan nilai 44%.

BAB V

KESIMPULAN DAN SARAN

5.1 Kesimpulan

Kesimpulan yang dihasilkan pada uji coba dan evaluasi diatas diantaranya adalah

a. Dari ketiga metode klasisikasi yang digunakan yaitu K-Nearest Neighbour, Naïve Bayes, dan Support Vector Machine yang menghasilkan nilai tertinggi adalah metode klasifikasi naïve bayes dengan menggunakan ekstraksi fitur count vector dengan nilai 60%.

b. Dalam penelitian ini salah satu fitur yang mempengaruhi nilai akurasi adalah jenis metode ekstraksi fitur yang digunakan. Dalam artian apabila level token input yang digunakan berbeda maka hasil akurasinya juga akan berbeda. Ekstraksi fitur yang menggunakan karakter akan berbeda hasilnya dengan yang menggunakan kata dan ngram.

5.2 Saran

Terdapat beberapa saran untuk penelitian selanjutnya , diantaranya :

a. Perlu dilakukan text preprocessing lebih lanjut untuk bisa mendapatkan Bahasa Indonesia baku.

b. Perlu dilakukan pembersihan Bahasa daerah yang digunakan dalam update status facebook agar saat text preprocessing lebih mudah.

c. Perlu dilakkukan pembagian jumlah data pada masing-masing kelas dengan seimbang agar bisa menghasilkan nilai precsion, recall dan fscore yang merata.

d. Pengembangan penelitian selanjutnya dapat diperdalam lagi pada pengaruh dari preprocessing text. Selain itu juga terdapat lagi peluang penelitian pada akun facebook tokoh masyarakat, institusi, toko atau perusahaan.

DAFTAR PUSTAKA

PUSTAKA BUKU

David, Hume. "*Basic-Emotions*.Pdf."

Paul Ekman, Wallace V. Friesen, Phoebe Ellsworth. 2013. "*Emotion in the Human Face: Guidelines for Research and an Integration of Findings." In ed. Leonard Krasner Arnold P. Goldstein.* United States of America: Elsevier Ltd, 204.

Prawitasllri, Johana E. 1991. "*Mengenal Emosi Melalui Komunikasi Nonverbal*." (1990).

PUSTAKA MAJALAH, JURNAL ILMIAH ATAU PROSIDING

Abraham, R., Simha, J. B., & Iyengar, S. S. (2009). Effective Discretization and Hybrid feature selection using Naïve Bayesian classifier for Medical datamining. *International Journal of Computational Intelligence Research, 5(2), 116–129*. https://doi.org/10.5019/j.ijcir.2009.175

Ardiada, I. M. D., Sudarma, M., & Giriantari, D. (2019). Text Mining pada Sosial Media untuk Mendeteksi Emosi Pengguna Menggunakan Metode Support Vector Machine dan K-Nearest Neighbour. *Majalah Ilmiah Teknologi Elektro*, *18*(1), 55. https://doi.org/10.24843/mite.2019.v18i01.p08

Asiati, D. I., & Septadiyanto, S. (2019). Karakteristik Pengguna Media Sosial. *Mbia*, *17*(3), 25–36. https://doi.org/10.33557/10.33557/mbia.v17i3.158

Banjarsari, M. A., Budiman, I., & Farmadi, A. (2016). Penerapan K-Optimal Pada Algoritma Knn Untuk Prediksi Kelulusan Tepat Waktu Mahasiswa Program Studi Ilmu Komputer Fmipa Unlam Berdasarkan Ip Sampai Dengan Semester 4. *Klik - K Jurnal Ilmu Komputer*, *2*(2), 159–173. https://doi.org/10.20527/KLIK.V2I2.26

Bata, J., Suyoto, & Pranowo. (2015). Leksikon Untuk Deteksi Emosi Dari Teks Bahasa Indonesia. *Seminar Nasional Informatika 2015 (SemnasIF 2015)*, *2015*(November), 195–202.

Calvo, R. A., & D'Mello, S. (2010). Affect detection: An interdisciplinary review of models, methods, and their applications. *IEEE Transactions on Affective Computing, 1(1), 18–37*. https://doi.org/10.1109/T-AFFC.2010.1

Dandannavar, P. S., Mangalwede, S. R., & Kulkarni, P. M. (2018). Social Media Text - A Source for Personality Prediction. *Proceedings of the International Conference on Computational Techniques, Electronics and Mechanical Systems, CTEMS 2018, 62–65*. https://doi.org/10.1109/CTEMS.2018.8769304

David, H. (n.d.). *Basic-Emotions.pdf*.

Ferdinan, A. H., T, C. S. S., Elektro, F. T., Telkom, U., Mining, D., & Neighbor, K. (2018). Klasifikasi Emosi Pada Lirik Lagu Menggunakan Metode K-Nearest Neighbor Emotion Classification in Song Lyrics Using. 5(3), 6187–6194.

Ilmiah, J., Komputa, I., & Bandung, J. D. (n.d.). Klasifikasi Emosi Pada Teks Bahasa Indonesia Menggunakan Metode K-Nearest Neighbor Dengan Wildan Abdul Gani *Jurnal Ilmiah Komputer dan Informatika (KOMPUTA)*.

Islam, M. R., Kamal, A. R. M., Sultana, N., Islam, R., Moni, M. A., & Ulhaq, A. (2018). Detecting Depression Using K-Nearest Neighbors (KNN) Classification Technique. *International Conference on Computer, Communication, Chemical, Material and Electronic Engineering, IC4ME2 2018*, 1–4. https://doi.org/10.1109/IC4ME2.2018.8465641

Kabir, F., Mofizur Rahman, C., Hossain, A., & Dahal, K. (2011). Enhanced Classification Accuracy on Naive Bayes Data Mining Models. *International Journal of Computer Applications*, *28*(3), 9–16. https://doi.org/10.5120/3371-4657

Lopatovska, I., & Arapakis, I. (2011). Theories, methods and current research on emotions in library and information science, information retrieval and human-computer interaction. *Information Processing and Management*, *47*(4), 575–592. https://doi.org/10.1016/j.ipm.2010.09.001

Musa, O. R., & Alang, A. (2017). Analisis Penyakit Paru-Paru Menggunakan Algoritma K-Nearest Neighbors Pada Rumah Sakit Aloei Saboe Kota Gorontalo. *ILKOM Jurnal Ilmiah*, *9*(3), 348.

https://doi.org/10.33096/ilkom.v9i3.177.348-352

Nedwidek, E. (2010). Facebook for Bussines. In *Facebook for Bussines*. Lulu.com.

Novantirani, A., Sabariah, M. K., & Effendy, V. (2015). Analisis Sentimen pada Twitter untuk Mengenai Penggunaan Transportasi Umum Darat Dalam Kota dengan Metode Support Vector Machine. *E-Proceeeding of Engineering*, *2*(1), 1–7.

Paul Ekman, Wallace V. Friesen, P. E. (2013). Emotion in the human face: Guidelines for research and an integration of findings (L. K. Arnold P. Goldstein (ed.); p. 204). Elsevier Ltd.

Prawitasllri, J. E. (1991). *Mengenal Emosi Melalui Komunlkasi Nonverbal*. *1990*.

Rohman, A. N., Utami, E., & Raharjo, S. (2019). Deteksi Kondisi Emosi pada Media Sosial Menggunakan Pendekatan Leksikon dan Natural Language Processing. *Eksplora Informatika*, *9*(1), 70–76. https://doi.org/10.30864/eksplora.v9i1.277

Saputra, I., & Rosiyadi, D. (2019). Perbandingan Kinerja Algoritma K-Nearest Neighbor , Naïve Bayes Classifier dan Support Vector Machine dalam Klasifikasi Tingkah Laku Bully pada Aplikasi Whatsapp. *12*(2), 101–111.

Sofiyana, L., Abidin, Z., & Nurhayati, H. (2012). Klasifikasi Emosi Untuk Teks Berbahasa Indonesia Dengan Menggunakan K-Nearest Neighbor (Vol. 1, Issue January, pp. 194–299). http://www-scf.usc.edu/~saman/pubs/2007-MS-Thesis.pdf%5Cnpapers3://publication/uuid/4A52EBCD-41F0-482E-9029-4262C6AFA5FC

Susilowati, E., Sabariah, M. K., & Gozali, A. A. (2015). Implementasi Metode Support Vector Machine untuk Melakukan Klasifikasi Kemacetan Lalu Lintas Pada Twitter. *E-Proceeding of Engineering*, *2*(1), 1–7.

Ting, S. L., Ip, W. H., & Tsang, A. H. C. (2011). Is Naïve bayes a good classifier for document classification? *International Journal of Software Engineering and Its Applications*, *5*(3), 37–46.

Wu, J., Gao, Z., & Hu, C. (2009). An empirical study on several classification algorithms and their improvements. *Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, *5821 LNCS*, 276–286. https://doi.org/10.1007/978-3-642-04843-2_30