

Deen Virtuoso: Evaluasi Sentimen menggunakan Regex dan Lexicon-based Algorithm terhadap Dakwah Digital berbasis Teks

Cetta Reswara Parahita - 13521133

Program Studi Teknik Informatika
Sekolah Teknik Elektro dan Informatika
Institut Teknologi Bandung, Jalan Ganesha 10 Bandung
E-mail (gmail): 13521133@std.stei.itb.ac.id

Abstract— Dakwah dalam dunia digital memiliki hambatan susahnya konfirmasi komunikasi dua arah antara *muballigh* dan *mustami*¹. Analisis sentiment akan mempermudah *muballigh* untuk menyadari posisi penerimaan teks dakwahnya. Digunakan pemanfaatan regex dan lexicon-based algorithm untuk melakukan analisis sentimen ini. Metode analisis sentimen dilakukan dengan tahap tweet extraction, Islamic-lexicon creation, data preprocessing, sentiment classification, wight constant determination, dan general evaluation. Analisis sentimen ditampilkan pada GUI dan didapatkan hasil pengujian bahwa metode ini efektif untuk memprediksi sentimen masyarakat terhadap teks dakwah.

Keywords—dakwah, lexicon-based algorithm, sentiment analysis, regex

I. PENDAHULUAN

Transformasi dan invensi teknologi telah menjadi bagian besar dari kehidupan manusia saat ini. Teknologi yang semakin mutakhir ini hampir-hampir memenuhi seluruh kebutuhan dasar manusia sehingga mengakibatkan kehidupan digital dengan mudah mengambil peranan yang signifikan dalam menunjang bahkan menyubstitusi kehidupan nyata manusia.

Segala aspek kehidupan manusia, tak terkecuali aspek sosial dan religi, akhirnya memiliki porsinya sendiri dalam dunia digital. Salah satunya termasuk dakwah. Menurut M. Natsir dalam bukunya Fiqhud Dakwah (1978), dakwah adalah seruan atau ajakan dengan menyentuh hati manusia sesuai dengan fitrahnya, supaya mereka menjalani kehidupan sesuai dengan petunjuk Ilahi. Bagi seorang muslim, dakwah adalah bentuk kewajiban fitrah¹ manusia makhluk sosial (*makhluk ijtima'i*) yang kemudian ditegaskan kembali dalam Kitabullah dan Sunnah sebagaimana beberapa potongan ayat Alquran berikut:

وَلْتَكُنْ مِنْكُمْ أُمَّةٌ يَدْعُونَ إِلَى الْخَيْرِ وَيَأْمُرُونَ بِالْعُرْفِ وَيَنْهَوْنَ عَنِ الْمُنْكَرِ وَأُولَئِكَ هُمُ الْمُفْلِحُونَ

"dan hendaklah ada diantara kamu segolongan umat yang menyeru kepada kebajikan, menyuruh kepada yang ma'ruf dan mencegah dari yang munkar, merekalah orang-orang yang beruntung." (QS Ali- Imran : 104).

Dakwah sebagai sebuah fitrah¹ sekaligus kewajiban bagi manusia membuatnya tidak dapat lepas bahkan dalam kehidupan digital manusia. Digitalisasi teknologi ini bahkan dianggap sebagai salah satu hal yang mengkatalis dan mempermudah proses dakwah. Berbagai bentuk dakwah dalam dunia digital dengan mudah kita jumpai dewasa ini. Akan tetapi, kemudahan ini juga mendatangkan sebuah tantangan tersendiri, diantaranya dijelaskan oleh Islamiy (2021) adalah susahnya konfirmasi komunikasi dua arah antara *muballigh*² dan *mustami*³, yang menyebabkan *muballigh*² lebih sulit melihat secara langsung dampak dari dakwahnya pada perubahan *mustami*³. Padahal, dakwah itu sendiri dijelaskan oleh M. Natsir (1978) lebih dari sekadar menyampaikan konten, melainkan memerlukan beberapa syarat, yaitu mencari materi yang cocok, mengetahui keadaan subjek dakwah secara tepat, memilih metode yang representatif, menggunakan bahasa yang bijaksana.

Dakwah menggunakan bahasa yang bijaksana ini dijelaskan dalam Alquran sebagai *qaulan Layyina* atau kata-kata yang lemah lembut dan enak didengar, sikap yang bersahabat, serta perilaku yang menyenangkan dalam menyerukan agama Allah (Hani'ah, 2013). Kalimat dakwah yang baik ialah yang dapat menimbulkan sentimen positif bagi para penggunanya untuk dapat berlaku sesuai dengan fitrah dari Allah. Untuk membantu permasalahan *muballigh*² dalam melihat lebih objektif dampak kalimat dakwahnya terhadap sentimen *mustami*³ ketika membaca kalimat dakwah, penggunaan algoritma analisis sentiment sebagai perlu dimanfaatkan.

Analisis sentiment akan mempermudah *muballigh* untuk menyadari posisi penerimaan teks dakwahnya sehingga menghindari sebuah kalimat dakwah yang terkesan negatif menimbulkan persepsi tersendiri tanpa *mustami*³ mengonfirmasi secara langsung konteksnya kepada *muballigh*. Algoritma untuk analisis sentiment ini telah banyak dikembangkan, mulai dari yang sederhana yakni *string matching* hingga memanfaatkan pembelajaran komputer seperti CNN atau SVM. Salah satu algoritma yang sedang dikembangkan untuk analisis sentiment saat ini adalah algoritma berbasis regex dan lexicon. Algoritma ini memanfaatkan pendekatan semantik yang terhitung sederhana namun terbukti praktis dalam melakukan penilaian analisis sentiment.

Oleh karena itu, pemanfaatan regex dan lexicon-based algorithm akan dikembangkan dalam proyek *DeenVirtuoso* untuk melakukan evaluasi sentimen sebuah tek dalam konteks dakwah pada ruang digital pada makalah kali ini.

II. TEORI DASAR

A. Analisis Sentimen

Analisis sentimen adalah proses menentukan sentimen dan mengelompokkan polaritas teks dalam dokumen atau kalimat sehingga kategori dapat ditentukan sebagai sentimen positif, negatif, atau netral (Samsir, 2021). Analisis sentimen juga dapat disamakan dengan opinion mining, karena berfokus pada pendapat yang menyatakan positif atau negatif. Dalam analisis sentimen, penambangan data dilakukan untuk menganalisis, memproses, dan mengekstrak data tekstual dalam suatu entitas, seperti layanan, produk, individu, fenomena atau topik tertentu. Proses analisis dapat mencakup teks ulasan, forum, tweet, atau blog, dengan data *preprocessing* yang mencakup proses tokenization, stopwords, penghapusan, stemming, identifikasi sentimen, dan klasifikasi sentimen.

Analisis sentiment tergolong relative dekat dengan komputasional lingusitik, pemrosesan bahasa natural, dan text mining (Mejova, 2009). Analisis sentiman ini menawarkan penelaahan aspek afektif psikologis dan penilaian rasional judgmental dengan bantuan data mining dan komputasional linguistik. Analisis sentiment ini merupakan bidang yang memperelajari elemen subjektif yang dijelaskan Wiebe dkk. (2004) sebagai ekspresi linguistik mengenai hal-hal privat yang tercantum di dalam konteks linguistik.

Sentimen sendiri merupakan refleksifitas opini yang menggambarkan kecenderungan perasaan seseorang (Pang dan Lee, 2008). Sentimen dapat tergambarkan secara eksplisit seperti pada pesan “Hari ini sangat menyenangkan” yang secara langsung menggambarkan perasaan dalam opininya atau secara implisit seperti pada pesan “Earphone ini pasti akan rusak dalam dua hari” yang secara tersirat mengimplikasikan kecenderungan perasaan tertentu yakni perasaan negative ketidakpercayaan (Liu, 2006). Polaritas sentiment sendiri adalah sebuah fitur sebuah teks yang menjelaskan dikotomi atau rentang yang menunjukkan kecenderungan positif atau negative dari suatu teks (Mejova, 2009).

Analisis sentiment secara umum dilakukan dengan deteksi opini atau sentiment. Deteksi ini biasa dilakukan dengan meninjau klasifikasi kemunculan konteks yang dapat dilakukan dengan berbagai teknik yang melakukan tinjauan dari berbagai sudut pandang, seperti TF-IDF yang berfokus pada tinjauan frekuensi kemunculan istilah, n-grams yang berfokus pada tinjauan penempatan istilah, part-of-speech yang berfokus pada tinjauan adjektif dari sebuah kalimat, syntax yang berfokus pada algoritma subpohon dari kalimat, serta negasi yang befokus pada kemunculan negasi dalam kalimat (Mejova, 2009).

Deteksi opini dan sentiment ini kemudian perlu diolah untuk menentukan orientasi semantik polaritas sentimen menggunakan pendekatan skala multipoin. Beberapa metode yang biasa digunakan adalah algoritma berbasis lexicon,

algoritma berbasis teknik statistika, dan algoritma yang memanfaatkan *machine training* dari beberapa dokumen.

Metode analisis sentimen berbasis algoritma lexicon memiliki beberapa kelebihan dibandingkan metode lain. Dijelaskan oleh Kaity dan Balakrishnan (2020) bahwa metode ini berbasis kemanusiaan, teks alami, dan sistem kamus yang menjadikannya lebih akurat, mudah diaplikasikan dalam berbagai bahasa dan variasinya, serta fleksibel untuk diterapkan secara general meskipun memerlukan lebih banyak usaha, waktu, *pre-processing*, dan sedikit memerlukan usaha lebih dalam tinjauan berbasis perbedaan kultur di berbagai bahasa.

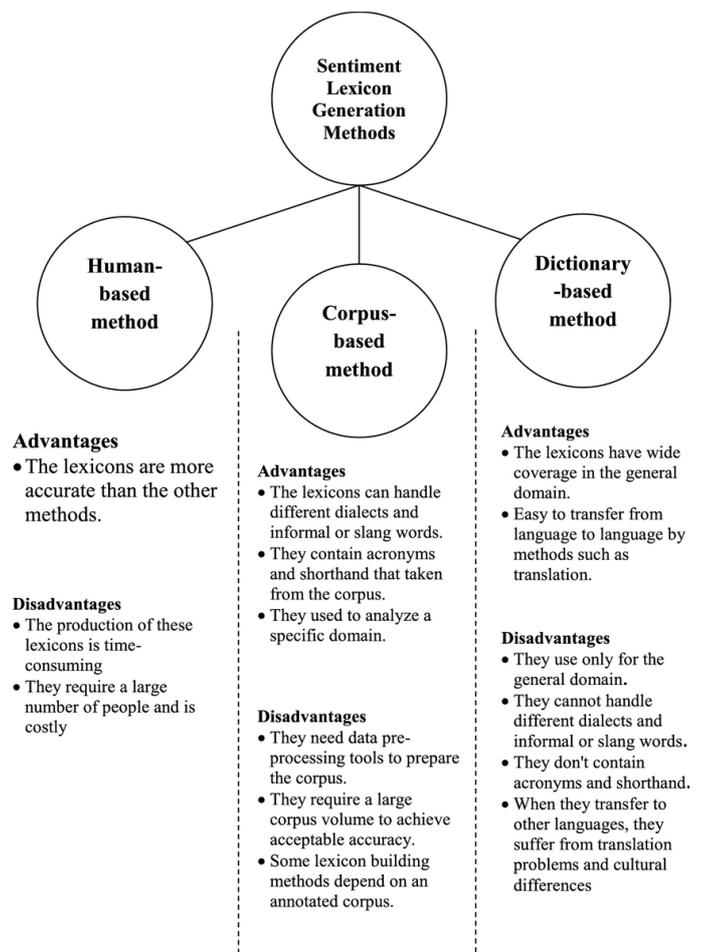


Fig. 1. The main advantages and disadvantages of sentiment lexicons built by each method (Source: *An integrated semi-automated framework for domain-based polarity words extraction from an unannotated non-English corpus from The Journal of Supercomputing on page 76 by Kaity & Balakrishnan, 2020*)

Dengan berbagai kelebihan dan kekurangan diatas, metode lexicon dipilih sebagai metode analisis sentimen pada teks dakwah dengan *pre-processing* berbasis regex mempertimbangkan latar belakang permasalahan yang berfokus kepada pembuatan alat untuk menyadari posisi penerimaan teks dakwah *muballigh* terhadap *mustami*¹ yang

¹Fitrah ialah sesuatu yang mendasar dalam sifat manusia sejak penciptaan manusia

²Orang yang melakukan dakwah, orang yang menyampaikan kebaikan dan mencegah dari kemunkaran

³Pendengar atau orang yang mendengarkan dakwah

memerlukan keakuratan konteks yang lebih berfokus pada pendekatan berbasis kemanusiaan, teks alami, dan sistem kamus sebagaimana fitur yang ditawarkan oleh *lexicon-based algorithm* dalam analisis sentimen.

B. Lexicon-based Algorithm

Pendekatan algoritma berbasis lexicon menggunakan orientasi semantik kontekstual dari setiap kata dalam teks untuk mengidentifikasi sentimen dari keseluruhan teks. Kata-kata dalam teks dibandingkan dengan kata-kata dalam daftar yang dipelihara yang diberi label dengan orientasi semantik mereka. Berdasarkan orientasi semantik masing-masing kata, sentimen keseluruhan teks dihitung, dan teks diklasifikasikan sebagai positif, negatif, atau netral. Algoritma analisis sentimen berbasis lexicon berbeda dari model lain dalam cara menggabungkan nilai sentimen dari kata-kata positif dan negatif dalam sebuah pesan. Dalam pendekatan ini, diperlukan kamus kata-kata positif dan negatif, yang disebut sebagai "*bag of words*" (Syed, 2021).

Dijelaskan oleh Ortony, Clore, dan Collins pada buku mereka yang berjudul *The Cognitive Structure of Emotions* (1988) tentang beberapa teori tentang emosi "fundamental" atau "dasar" (lihat *fig. 2*). Metode yang paling terkini ditemukan terkait *lexicon-based algorithm* ini sekaligus pengetahuan yang termasuk paling canggih adalah analisis basis pengetahuan *common sense* yang telah dikembangkan oleh peneliti dalam kecerdasan buatan. Bagian ini menjelaskan hasil pekerjaan terbaru yang dilakukan pada lexicon dengan anotasi sentimen.

Theorist	Fundamental emotions	Basis for selection	Reference
Arnold, M. B.	anger aversion courage dejection desire despair fear hate hope love sadness	relation to action tendencies	Arnold (196)
Ekman, P.	anger disgust fear joy sadness surprise	universal facial expressions	Ekman, Friesen & Ellsworth (1982)
Frijda, N.	desire joy pride surprise distress anger aversion contempt fear shame	forms of action readiness	Frijda (1987, and personal communication)
Gray, J.	rage/terror anxiety joy	hardwired	Gray (1982)
Izard, C. E.	anger contempt disgust distress fear guilt interest joy shame surprise	hardwired	Izard (1972)
James, W.	fear grief love rage	bodily involvement	James (1884)
McDougall, W.	anger disgust elation fear subjection tender-emotion wonder	relation to instincts	McDougall (1926)
Mowrer, O. H.	pain pleasure	unlearned emotional states	Mowrer (1960)
Oatley, K., and Johnson-Laird, P. N.	anger disgust fear happiness sadness	do not require propositional content	Oatley & Johnson-Laird (1987)
Panksepp, J.	expectancy fear rage panic	hardwired	Panksepp (1982)
Plutchik, R.	acceptance anger anticipation disgust joy fear sadness surprise	relation to adaptive biological processes	Plutchik (1980)
Tomkins, S. S.	anger interest contempt disgust distress fear joy shame surprise	density of neural firing	Tomkins (1984)
Watson, J. B.	fear love rage	hardwired	Watson (1930)
Weiner, B.	happiness sadness	attribution-independent	Weiner & Graham (1984)

Note. Not all the theorists represented in this table are equally strong advocates of the idea of basic emotions. For some it is a crucial notion (e.g., Izard, 1977; Panksepp, 1982; Plutchik, 1980; Tomkins, 1984), while for others it is of peripheral interest only and their discussions of basic emotions are hedged (e.g., Mowrer, 1960; Weiner & Graham, 1984).

Fig. 2. A selection of lists of "fundamental" or "basic" emotions (Source: *The Cognitive Structure of Emotions* by Ortony, Clore, and Collin, 1988)

Secara umum, ada dua cara untuk membuat lexicon, yaitu secara manual (Taboada dkk., 2011; Tong dkk., 2001) atau secara otomatis dengan memperluas dari sekumpulan kata (Kanayama dkk., 2006; Kaji dan Kitsuregawa, 2007; Turney, 2002; Turney dan Littman, 2003). Secara sederhana, konsep untuk membuat lexicon yang secara manual memerlukan beberapa jenis tinjauan yang perlu diperhatikan sebagai berikut:

1. **Common Lexicon:** Lexicon ini berisi data yang memiliki makna atau arti semantik yang sama di berbagai domain dan kategori.

- **Kata-Kata Sentimen Default:** Kata-kata sentimen positif dan negatif yang memiliki nilai sentimen atau arti yang sama di berbagai domain. Misalnya, kata sentimen "baik" selalu menggambarkan sentimen positif dan tidak tergantung pada kategori apa pun. Kata-kata sentimen positif atau negatif memiliki skor sentimen +1 atau -1 untuk menunjukkan polaritas yang sesuai.
- **Kata-Kata Negasi:** Kata-kata negasi adalah kata-kata yang membalikkan polaritas sentimen. Misalnya, "Umur baterai tidak baik" memiliki sentimen negatif.
- **Kata-Kata Negasi Tersembunyi:** Dalam kalimat, "TV membutuhkan remote yang lebih baik", kata "membutuhkan" merupakan kata negasi tertutup. Kata-kata negasi tertutup beroperasi pada tingkat kalimat dan menunjukkan ketiadaan atau kehadiran suatu makna yang tidak diinginkan dalam fitur produk.
- **Kata Hubung:** Kata hubung digunakan untuk memisahkan kalimat menjadi klausa. Daftar kata pisah terdiri dari konjungsi dan tanda baca. Sebagai contoh, kalimat kompleks, "Kamera bagus tapi baterai buruk" dipisah menjadi dua klausa, yaitu "Kamera bagus" dan "Baterai buruk".

2. **Category Specific Lexicon:** Lexicon ini berisi:

- **Katalog Produk:** Mengidentifikasi semua produk yang kita minati.
- **Katalog Fitur:** Daftar atribut yang dimiliki produk. Hal ini memungkinkan mesin Serendio untuk melakukan analisis pada tingkat fitur.
- **Kata-Kata Sentimen (Positif dan Negatif) yang Khusus untuk Kategori Tertentu:** Misalnya, untuk kategori Televisi, sebuah produk dapat menjadi Samsung TV. Fitur dapat berupa layar LCD, dan kata "silau" akan menjadi kata sentimen negatif yang khusus untuk kategori tersebut.

Kemudian konsep manual ini dikembangkan menjadi cara otomatis. Salah satu cara otomatis untuk melakukan tinjauan lexicon adalah dengan menggunakan WordNet. WordNet adalah basis data besar untuk leksikal yang awalnya dibuat khusus untuk bahasa Inggris, yang dikembangkan di bawah arahan George A. Miller. Database ini kemudian dikembangkan untuk beberapa negara lain. Kata benda, kata kerja, kata sifat, dan kata keterangan dikelompokkan ke dalam kumpulan sinonim kognitif (synset), masing-masing

mengekspresikan konsep yang berbeda. Synset saling terhubung melalui hubungan konseptual-semantik dan leksikal.

Esuli dan Sebastiani (2006) memperluas WordNet dengan menambahkan label polaritas (Positif-Negatif) dan objektivitas (Subjektif-Objektif) untuk setiap kata. Untuk memberi label pada setiap kata, mereka mengklasifikasikan synset (sekelompok sinonim) yang termasuk dalam kata tersebut menggunakan seperangkat pengklasifikasi ternary (alat yang melekat pada setiap objek dengan satu dari tiga label), masing-masing mampu menentukan apakah suatu synset adalah Positif, Negatif, atau Objektif. Skor yang dihasilkan berkisar antara 0.0 hingga 1.0, memberikan evaluasi bertingkat terhadap properti terkait opini dari kata-kata tersebut. Ini dapat dijelaskan secara visual seperti pada *fig. 3*. Garis-garis segitiga mewakili salah satu dari tiga klasifikasi (positif, negatif, dan objektif). Sebuah kata dapat ditempatkan dalam ruang ini sebagai titik, mewakili sejauh mana kata tersebut termasuk dalam masing-masing klasifikasi.

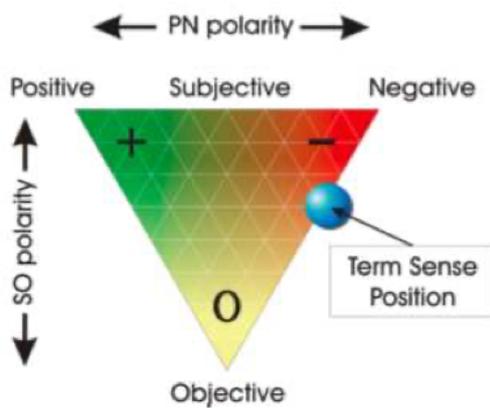


Fig. 3. Graphical representation of opinion-related properties of a term (Source: *Sentiment Analysis: An Overview by Mejova, 2009*)

WordNet juga telah langsung digunakan dalam Analisis Sentimen. misalnya, Kim dan Hovy (2004) dan Hu dan Liu (2005) menghasilkan leksikon kata-kata positif dan negatif dengan memulai dari daftar kecil kata-kata "seed" yang memiliki polaritas yang diketahui (misalnya, cinta, suka, bagus, dll.) dan kemudian menggunakan sifat antonimi dan sinonimi dari kata-kata untuk mengelompokkannya ke dalam salah satu dari kategori polaritas tersebut.

Pada implementasi yang akan dilakukan kali ini, lexicon akan diterapkan dengan menggabungkan implementasi otomatis dan manual. Implementasi otomatis akan memanfaatkan WordNet untuk melakukan pemrosesan data yang diambil dari twitter. Kemudian data diolah menjadi sebuah kamus lexicon terkait islam yang akan menjadi poin tambahan dalam penilaian lexicon.

Untuk menambahkan nilai hitung dengan lexicon ini, dibuat skala dinamis dengan menerapkan tinjauan apakah konstanta perhitungan terdistribusi normal atau tidak menggunakan pembuktian hipotesis P-value dengan Z-score. Setelah itu data akan diolah sesuai ketentuan awal, apakah tergolong positif atau negative yang kemudian akan menjadi

faktor evaluasi utama dalam evaluasi sentimen teks dakwah DeenVirtuoso.

C. Regex

Dalam buku Regular Expressions: The Complete Tutorial oleh Jan Goyvaerts (2007), dijelaskan *regular expression* atau ekspresi reguler adalah pola yang menggambarkan sejumlah teks tertentu. Nama ini berasal dari teori matematika yang biasa disingkat menjadi regex atau regexp.

Dengan pola ekspresi reguler, dapat dilakukan pencarian alamat email dalam file teks, atau pemverifikasian apakah string tertentu mirip dengan alamat email. Selain itu ekspresi reguler juga dapat digunakan dengan data apa pun yang dapat diakses menggunakan aplikasi atau bahasa pemrograman yang sedang digunakan.

Ekspresi reguler digunakan dalam banyak aplikasi praktis, misalnya Perl, Awk, Python, egrep, vi, dan emacs. Diketahui bahwa ekspresi reguler praktis berbeda dari yang teoritis. Ekspresi reguler praktis sering disebut regex. Regex dikembangkan di bawah pengaruh ekspresi reguler teoritis (Campeanu, dkk., 2002). Dalam regular expression teoritis terdapat bentuk basis dari regex yang dijelaskan dalam *fig. 4*.

operation	regular expression	matches	does not match
concatenation	<code>aabaab</code>	aabaab	every other string
wildcard	<code>.u.u.u.</code>	cumulus jugulum	succubus tumultuous
union	<code>aa baab</code>	aa baab	every other string
closure	<code>ab*a</code>	aa abbba	ab ababa
parentheses	<code>a(a b)aab</code>	aaaab abaab	every other string
	<code>(ab)*a</code>	a ababababa	aa abbba

regular expression	matches	does not match
<code>*spb.*</code> <i>contains the trigraph spb</i>	raspberry crispbread	subspace subspecies
<code>a* (a*ba*ba*ba*)*</code> <i>multiple of three b's</i>	bbb aaa bbbaababaa	b bb baabbbba
<code>.*0....</code> <i>fifth to last digit is 0</i>	1000234 98701234	111111111 403982772
<code>gcg(cgg agg)*ctg</code> <i>fragile X syndrome indicator</i>	gcgctg gcgaggctg gcgaggctg	gcgcgg cggcggcggctg gcgcaggctg

Fig. 4. Regular Expressions: Basic Operations Notation to specify a set of strings. (Source: Introduction to Theoretical CS by Princeton University, 2012)

Dalam implementasinya, regex digunakan menyesuaikan dengan bahasa pemrograman yang digunakan. Penggunaan ekspresi reguler dalam bahasa Python dapat diimplementasikan dengan memanfaatkan modul `re` dengan ekspresi yang mengikuti pada *fig. 5*.



Fig. 5. Regular Expression on Python Cheat Sheet (Source: Dataquest, 2019)

Pada makalah ini, penggunaan regular expression dimanfaatkan pada *pre-processing*. Sebuah kalimat cenderung berisi banyak variasi huruf, tanda baca, atau kata-kata slang yang perlu untuk diproses sebelum diolah lebih lanjut dalam algoritma. Tujuan *pre-processing* adalah untuk menormalisasi kalimat sehingga menjadi bentuk yang standar yang dapat dengan lebih mudah dilakukan ekstraksi sentimen terhadapnya (Palanisamy, dkk., 2013). Tahap-tahap *pro-processing* ini mencakup:

1) *Stemming*: Stemmer memberikan kata dasar. Lexicon Serendio hanya berisi kata dasar. Jadi, kata-kata non kata dasar distem dan digantikan dengan kata dasar. Misalnya, kata-kata seperti 'loved', 'loves', 'loving', 'love' digantikan dengan 'lov'. Ini akan membantu mesin mencocokkan kata dari teks dengan lexicon. Stemming dilakukan menggunakan modul NLTK pada Python atau dengan menggunakan regular expression tersendiri.

2) *Penyederhanaan kata berlebihan*: Kata-kata yang memiliki huruf yang sama lebih dari dua kali dan tidak ada dalam lexicon disederhanakan menjadi kata dengan huruf berulang hanya satu kali (Kouloumpis dkk., 2011). Misalnya, kata berlebihan "NOOOOOO" disederhanakan menjadi "NO".

3) *Deteksi emotikon atau tanda baca berlebih*: Emotikon atau tanda baca berlebih memiliki sentimen yang terkait

dengan itu (Ritter dkk., 2011; Ritter dkk., 2012). Oleh karena itu perlu digunakan sebagai tinjauan tersendiri.

D. Kaidah Diksi dalam Penyampaian Dakwah

Diksi atau pemilihan kata bukan sekadar kegiatan memilih kata yang tepat, melainkan juga memilih kata yang sesuai dengan situasi dan konteks pemakaiannya (Hani'ah, 2001). Selain itu, maknanya tidak bertentangan dengan nilai rasa masyarakat pemakainya. Oleh karena itu, dalam memilih kata diperlukan pertimbangan tertentu. Dalam penyampaian dakwah, diksi yang perlu dipilih telah diatur dalam Alquran. Penyampaian ini antara lain dirangkum oleh Hani'ah (2001) sebagai berikut:

1) Memilih kata-kata yang kariema

Memilih kata-kata yang kariema sebagai salah satu formula kaidah diksi dalam berkomunikasi, terdapat dalam surat Al-Isra' ayat 23. Dilihat dari segi bahasa, kata kariema berarti perkataan mulia. Perkataan yang mulia adalah perkataan yang memberi penghargaan dan penghormatan kepada orang yang diajak bicara. Sehubungan dengan diksi atau pilihan kata, maka kata yang termasuk kategori kariema adalah kata-kata yang mencerminkan kemuliaan, atau menggambarkan sikap memuliakan orang yang diajak bicara.

2) Memilih kata-kata yang ma'rufa

Kata *Qaulan Ma'rufa* disebutkan Allah dalam Al-Quran sebanyak lima kali. Dalam QS An-Nissa: 5, QS An-Nissa: 8, QS Al Baqarah: 235, QS Al Baqarah: 263, QS Al-Ahzab: 32. Kata ma'rufa dari kelima ayat tersebut besinonim dengan kata al-khair atau al-ikhshan yang berarti baik. Ucapan yang baik adalah ucapan yang diterima sebagai sesuatu yang baik dalam pandangan masyarakat lingkungan penutur. Dengan demikian, hubungan harmonis antarwarga akan terus dipelihara, dan karena itulah ayat ini ditetapkan dengan perintah ucapkanlah kepada mereka kata-kata yang baik (Shihab, 2000).

Perkataan yang baik maksudnya perkataan yang menimbulkan rasa tentram dan damai bagi orang yang mendengarkannya baik *interpersonal communication*, *group communication* dan *mass communication*. *Qaulan Ma'rufa* berarti kata-kata yang bermanfaat, memberikan pengetahuan, mencerahkan pemikiran, dan menunjukkan pemecahan kesulitan. *Qaulan Ma'rufa* lebih banyak ditunjukkan kepada wanita atau orang yang kurang beruntung kehidupannya. Hal ini dimaksudkan agar setiap orang dituntut untuk dapat berkomunikasi dengan pantas kepada orang lain, baik dengan siapapun, dimanapun, dan kapanpun, dengan syarat pembicaraan itu dapat mendatangkan manfaat dan pahala, baik untuk komunikator dan juga komunikan.

3) Memilih kata-kata yang layyina

Kata *Qaulan Layyina* disebutkan dalam QS Thaahaa: 44. Menurut Al-Maraghi, *Qaulan Layyinan* berarti pembicaraan yang lemah lembut agar lebih dapat

menyentuh hati dan menariknya untuk menerima dakwah. Sedangkan menurut Ibnu Kasir, yang dimaksud *layyinan* ialah kata kata sindiran/ bukan dengan kata-kata terus terang.

Berdasarkan pendapat di atas dapat disimpulkan *Qaulan Layyina* memiliki makna kata-kata yang lemah lembut, suara yang enak didengar, sikap yang bersahabat, dan perilaku yang menyenangkan dalam menyerukan agama Allah. Dengan kata-kata *Qaulan Layyinan*, orang yang diajak berkomunikasi akan merasa tersentuh hatinya, tergerak jiwannya dan tenang batinnya, sehingga ia akan patuh.

4) Memilih kata-kata yang baligha

Qaulan Baligha diartikan sebagai pembicaraan yang fasih atau tepat, jelas maknanya, terang, serta tepat mengungkapkan apa yang dikehendakinya atau juga dapat diartikan sebagai ucapan yang benar dari segi kata. Dan apabila dilihat dari segi sasaran atau ranah yang disentuhnya dapat diartikan sebagai ucapan yang efektif.

Ucapan yang artikulatif atau fasih lebih menarik dan dapat meminimalisir kesalahpahaman antara komunikan dan komunikator. Sebaliknya, ucapan yang tidak artikulatif, akan membosankan sehingga kurang mendapat perhatian dari komunikan atau pendengar.

5) Memilih kata-kata yang maysura

Secara etimologis, kata *Maysura* berasal dari kata *Yasara* yang artinya mudah atau gampang, berarti berkata dengan mudah dan gampang. Berkata dengan mudah maksudnya kata-kata yang digunakan mudah dicerna, mudah dimengerti, dan dipahami oleh komunikan. Dalam tafsir Al-Maraghi kata *Qaulan Masyuran* berarti kata-kata yang mudah dan lunak agar pesan dapat diterima oleh pendengar, maka pesan tersebut hendaknya disampaikan dengan cara yang mudah.

Salah satu prinsip komunikasi dalam Islam adalah setiap berkomunikasi harus bertujuan mendekatkan manusia dengan Tuhannya dan hamba-hamba-Nya yang lain. Seorang komunikator yang baik adalah komunikator yang mampu menampilkan dirinya sehingga disukai dan disenangi orang lain. Untuk dapat disenangi orang lain, ia harus memiliki sikap simpati (menempatkan diri secara imajinaif dalam posisi orang lain) dan empati (berada pada posisi orang lain).

6) Memilih kata-kata yang Sadieda

Perkataan qaulan sadieda berarti perkataan yang benar. Namun, tidak sederhana itu. Perkataan qaulan sadida diungkapkan Al-Quran dalam QS. An-Nisa: 9 dengan konteks pembicaraan mengenai wasiat. Menurut beberapa ahli tafsir seperti Hamka, At-Thabari, Al-Baghawi, Al-Maraghi dan Al-Buruswi bahwa *qaulan sadida* dari segi konteks ayat mengandung makna kekuatiran dan kecemasan seorang para pemberi wasiat

terhadap anak-anaknya yang digambarkan dalam bentuk ucapan-ucapan yang lemah lembut, jelas, jujur, tepat, baik, dan adil.

Peranan pemilihan diksi dalam berdakwah sangatlah penting, sebagaimana disampaikan oleh Ustadz Felix Siaw (2023) bahwa tiga pokok yang perlu dipahami oleh seorang *muballigh* adalah kaidah benar, baik, dan bagus. Diksi akan membawa penyampaian *muballigh* hingga pada tahapan bagus sehingga mudah diterima oleh target dakwahnya. Oleh karena itu perlu sekali diperhatikan bagaimana diksi ini dipilih oleh seorang *muballigh*.

III. IMPLEMENTASI

Dalam implementasi DeenVirtuoso, digunakan bahasa Python dengan detail repository pada laman <https://github.com/CettaReswara/DeenVirtuoso.git>. Implementasi dilakukan dengan proses sesuai dengan diagram berikut:

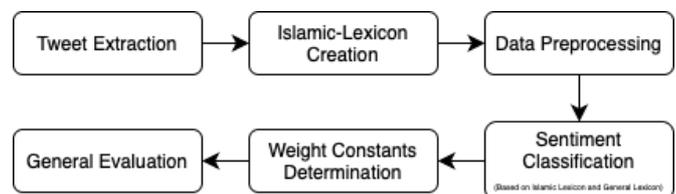


Fig. 6. Overall System Architectur of DeenVirtuoso: Evaluasi Sentimen menggunakan Regex dan Lexicon-based Algorithm terhadap Dakwah Digital berbasis Tek

A. Tweet Extraction

Fase awal yang dilakukan dalam implementasi adalah ekstraksi data yang akan digunakan sebagai sumber lexicon islami. Ekstraksi data adalah proses mengumpulkan *tweet* sebagai portal utama media sosial berbasis teks. Kata kunci yang digunakan antara lain adalah "islam", "muslim", "dakwah", "quran", dan "hadith". Selain itu juga dilakukan ekstraksi untuk *tweet* yang mengandung tagar "islamic", "muslims", "dawah", dan "islamicquotes". Ekstraksi ini dilakukan menggunakan modul *tweepy* pada Python dan disimpan pada dataset berbasis csv dengan menggunakan modul csv.

Langkah-langkah yang dilakukan untuk melakukan ekstraksi dari Twitter adalah sebagai berikut:

1. Menyambungkan kode pada Twitter API dengan membuat Akun Pengembang Twitter pada laman <https://developer.twitter.com>. Kemudian perlu dibuat sebuah aplikasi twitter yang dapat disambungkan pada laman asli atau pada laman github untuk mendapatkan akses pada data-data di twitter. Data ini kemudian disambungkan melalui *keys* dan *keys secret* yang kemudian perlu disimpan pada sebuah variabel dalam kode Python kita.
2. Menggunakan Library Python Tweepy. Lakukan instalasi library Tweepy menggunakan pip dengan menjalankan perintah berikut di terminal atau command prompt:

```
pip install tweepy
```

Import library Tweepy dan gunakan Consumer Key, Consumer Secret, Access Token, dan Access Token Secret yang telah Anda dapatkan sebelumnya. Kemudian buatlah objek API dan ekstrak cuitan yang terkait dengan kata kunci, tagar, atau akun yang telah ditentukan sebelumnya.

Implementasi pada poin ekstraksi ini dilakukan pada file `./src/tweet_mining.py`. Kemudian tiap *tweet* yang didapatkan dari ekstraksi sebelumnya disimpan dalam `./data/twitter_dataset.csv` dengan atribut Waktu, ID, Username, dan Teks. Bentuk implementasi kode secara umum adalah sebagai berikut:

```
import csv
import tweepy

# Mengatur Consumer Key dan Consumer Secret
consumer_key = "QLQf3dUALjDbEDnRl9Amsnfe"
consumer_secret = "KXW4anAYqxE7FDtaPDPVu0PBqS5Bx9putyouTouei9tEwIRFm3"

# Mengatur Access Token dan Access Token Secret
access_token = "1292359577292271618-28LRmx2NeieEX3BPpa5YGLJNgzMdkPE"
access_token_secret = "nbUssIhiSifvTS8xAtpJR4fbXcji68RwAWbAs4c6bd0Dl"

# Membuat objek OAuthHandler
auth = tweepy.OAuthHandler(consumer_key, consumer_secret)

# Mengatur Access Token dan Access Token Secret
auth.set_access_token(access_token, access_token_secret)

# Membuat objek API
api = tweepy.API(auth)

# Faktor-faktor pencarian
query_keywords = ["islam", "muslim", "dakwah", "quran", "hadith"]
query_hashtags = ["islamic", "muslims", "dawah", "islamicquotes"]

# Jumlah tweet yang akan diambil
tweet_count = 1000

# Mengumpulkan tweet yang sesuai dengan faktor-faktor pencarian
tweets = []

# Pencarian berdasarkan kata kunci
for keyword in query_keywords:
    query = f"{keyword} -filter:retweets"
    for tweet in tweepy.Cursor(api.search_tweets, q=query, lang="id").items(tweet_count):
        tweets.append(tweet.text)

# Pencarian berdasarkan hashtag
for hashtag in query_hashtags:
    query = f"#{hashtag} -filter:retweets"
    for tweet in tweepy.Cursor(api.search_tweets, q=query, lang="id").items(tweet_count):
        tweets.append(tweet.text)

# Menyimpan tweet ke dalam file CSV
filename = "./data/twitter_dataset.csv"

with open(filename, "w", newline="", encoding="utf-8") as file:
```

```
writer = csv.writer(file)
writer.writerow(["Tweet"])
for tweet in tweets:
    writer.writerow([tweet])

print(f"Dataset berhasil disimpan dalam file {filename}")
```

Fig. 7. Source code `tweets_mining.py`

B. Islamic Lexicon Creation dan Data Preprocessing

Setelah memiliki data tweets, selanjutnya perlu dibuat *lexicon* yang terkait dengan kata-kata keislaman. *Lexicon* ini dibuat berdasarkan penilaian dari sentimen tiap tweet dengan bantuan database *lexicon* yang pernah dibuat sebelumnya yakni WordNet pada modul NLTK. Langkah-langkah pembuatan *lexicon* islam ini mencakup preprocessing dan penambahan Islamic *lexicon* pada dataset. Detail proses yang dilakukan adalah sebagai berikut:

- Preprocessing Teks:
 - Melakukan proses preprocessing teks pada tweet yang telah dikumpulkan, seperti menghapus karakter khusus, tanda baca, dan mengubah teks menjadi huruf kecil. Pre-processing ini dilakukan dengan menggunakan implementasi regex untuk normalisasi teks dengan detail perubahan pada bagian source code.
 - Melakukan proses *stemming* atau *lemmatisasi* untuk mengubah kata-kata menjadi bentuk dasar.
 - Pre-processing ini juga akan digunakan pada bagian-bagian selanjutnya untuk analisis sentimen.
- Analisis Sentimen:
 - Melakukan analisis sentimen pada setiap tweet menggunakan algoritma berbasis *lexicon* yang tersimpan dalam modul NLTK
 - Menentukan label sentimen untuk setiap tweet, misalnya 'positive', 'negative', atau 'neutral'.
- Pembuatan Kamus:
 - Melakukan eksplorasi pada hasil analisis sentimen dan identifikasi kata-kata yang memiliki pengaruh kuat terhadap sentimen positif atau negatif.
 - Mengumpulkan kata-kata tersebut ke dalam kamus *lexicon* pada `islamic_lexicon.csv` dengan atribut teks, sentimen dengan menambahkan label sentimen ('positive' atau 'negative') untuk tiap kata dalam *lexicon*.
- Evaluasi dan Pemutakhiran:
 - Evaluasi kamus yang telah dibuat dengan melakukan pengujian pada data sentimen yang berbeda dari Twitter atau sumber data lainnya.

Pembuatan Islamic *lexicon* ini dilakukan pada file `./src/islamic_lexicon.py` kemudian set kata yang didapatkan disimpan pada `./data/Islamic_lexicon.csv`. Bentuk implementasi kode secara umum adalah sebagai berikut:

```
import csv
import re
import text_analyzer_algo as taa

def preprocess_text(text):
    # Menghapus karakter non-alfanumerik dan tanda baca
    text = re.sub(r'[\W\s-]', '', text)

    # Menghapus angka
    text = re.sub(r'\d', '', text)
```

```

# Mengubah teks menjadi huruf kecil
text = text.lower()

# Menghapus karakter berulang (contoh:
"haaaappppyyyy" menjadi "happy")
text = re.sub(r'(\w)\1+', r'\1', text)

# Menghapus spasi berlebihan
text = re.sub(r'\s+', ' ', text)

# Menghapus kata-kata pendek (kurang dari 3
karakter)
text = re.sub(r'\b\w{1,2}\b', '', text)

# Menghapus stopwords (kata umum yang tidak
berkontribusi pada sentimen)
stopwords = ['dan', 'atau', 'juga', 'dari',
'adalah'] # contoh stopwords, sesuaikan dengan
kebutuhan
text = ' '.join(word for word in text.split() if
word not in stopwords)

return text

def add_islamic_lexicon(filename):
    islamic_lexicon_current =
load_islamic_lexicon("./data/islamic_lexicon.csv")
    islamic_lexicon_new = {}

    with open(filename, "r", encoding="utf-8") as file:
        reader = csv.reader(file)
        next(reader) # Skip header row
        for row in reader:
            waktu, ID, username, teks = row
            words = teks.split()
            for word in words:
                if word not in islamic_lexicon_current:
                    islamic_lexicon_new[word.lower()] =
'positive'

```

Fig. 8. Source code *islamic_lexicon.py*

C. Sentiment Classification, Wight Constant Determination, dan General Evaluation

Setelah dibuat *islamic lexicon*, maka selanjutnya dilakukan *sentiment classification*. Secara umum proses klasifikasi dilakukan dengan menerima input dari pengguna pada GUI. Selanjutnya dilakukan data *pre-processing* seperti pada keterangan yang dijelaskan pada bagian sebelumnya. Selanjutnya data yang telah diproses akan dianalisis menggunakan dua pendekatan lexicon, yakni lexicon yang berlaku pada kalimat umum dari database WordNet dan lexicon khusus untuk kosakata islami dengan Islamic Lexicon. Secara umum, algoritma yang dilakukan sama dengan bagian Analisis Sentimen pada pembuatan Islamic Lexicon. Algoritma umum untuk pemrosesan data ini adalah sebagai berikut

```

def load_islamic_lexicon(filename):
    islamic_lexicon = {}

    with open(filename, "r", encoding="utf-8") as file:
        reader = csv.reader(file)
        next(reader) # Skip header row
        for row in reader:
            word, sentiment = row
            islamic_lexicon[word.lower()] =
sentiment.lower()

    return islamic_lexicon

def sentiment_analysis(text, islamic_lexicon):
    sentiment_score = 0

```

```

pos_word = 0
neg_word = 0

# Menghitung skor sentimen berdasarkan kamus
leksikon agama Islam
for word in text.lower().split():
    if word in islamic_lexicon:
        if islamic_lexicon[word] == 'positive':
            pos_word+=1
        else:
            neg_word+=1

if neg_word > pos_word:
    sentiment_score = -neg_word
else:
    sentiment_score = pos_word - neg_word

for word in text.lower().split():
    if word == 'tidak' or '-' in word:
        sentiment_score *= -1

# Menentukan label sentimen berdasarkan skor
sentimen
return sentiment_score

```

Fig. 9. Source code *islamic_lexicon.py*

```

import nltk

nltk.download('wordnet')
nltk.download('omw-1.4')

from nltk.corpus import wordnet as wn
from nltk.sentiment import SentimentIntensityAnalyzer

def get_sentiment_score(word):
    synsets = wn.synsets(word, lang='ind')
    if synsets:
        # Ambil definisi pertama dari synset
        definition = synsets[0].definition()
        # Gunakan SentimentIntensityAnalyzer dari NLTK
        sia = SentimentIntensityAnalyzer()
        sentiment_scores =
sia.polarity_scores(definition)
        # Ambil skor komposit
        sentiment_score = sentiment_scores['compound']
        return sentiment_score
    else:
        return 0

def analyze_sentiment(text):
    sentiment_score = 0
    positive_words = 0
    negative_words = 0
    words = text.lower().split()

    for word in words:
        if get_sentiment_score(word) == 0:
            words.remove(word)
        elif get_sentiment_score(word) < 0:
            negative_words += get_sentiment_score(word)
        else:
            positive_words += get_sentiment_score(word)

    if negative_words > positive_words:
        sentiment_score = -negative_words
    else:
        sentiment_score = positive_words -
negative_words

    for word in words:
        if word == 'tidak' or '-' in word:
            sentiment_score *= -1

    return sentiment_score

```

Fig. 10. Source code `text_analyzer_algo.py`

Dua *sentiment score* yang didapatkan kemudian dibobot secara dinamis berdasarkan tinjauan bobot yang memperhatikan distribusi normal data *tweet* yang sebelumnya telah dibuat menggunakan fungsi `calculate_wight` yang ada pada file `./src/weighting.py`

```
def calculate_weight(factor_1_ratings,
factor_2_ratings):
    # Calculate mean and standard deviation of ratings
    for each factor
    mean_rating_factor_1 = np.mean(factor_1_ratings)
    std_rating_factor_1 = np.std(factor_1_ratings)
    mean_rating_factor_2 = np.mean(factor_2_ratings)
    std_rating_factor_2 = np.std(factor_2_ratings)

    # Define the weight calculation function for each
    factor
    def calculate_weight_factor_1(rating):
        z_score = (rating - mean_rating_factor_1) /
std_rating_factor_1
        weight = stats.norm.pdf(z_score)
        return weight

    def calculate_weight_factor_2(rating):
        z_score = (rating - mean_rating_factor_2) /
std_rating_factor_2
        weight = stats.norm.pdf(z_score)
        return weight

    # Calculate weights for each factor
    weights_factor_1 =
[calculate_weight_factor_1(rating) for rating in
factor_1_ratings]
    weights_factor_2 =
[calculate_weight_factor_2(rating) for rating in
factor_2_ratings]

    # Combine the weights using a linear combination
    stopping_criteria = False
    weights_constant_1 = 0
    weights_constant_2 = 1
    while not stopping_criteria:
        combined_weights = weights_factor_1 *
np.array(weights_factor_1) + weights_constant_2 *
np.array(weights_factor_2)

        # Normalize the combined weights
        normalized_weights = combined_weights /
np.sum(combined_weights)

        # Verify that the weights follow a normal
        distribution
        _, p_value =
stats.normaltest(normalized_weights)

        if p_value > 0.05:
            stopping_criteria = True
        else:
            weights_constant_1 += 0.1
            weights_constant_2 -= 0.1
    return weights_constant_1, weights_constant_2
```

Fig. 11. Source code `weighting.py`

Weighting ini kemudian digunakan untuk *general analysis* dengan pendekatan sebagai berikut. Hasil kemudian akan ditampilkan pada GUI dengan detail:

```
def full_analysis(text):
    constfull, constislam =
wt.calculate_weight([ilm.sentiment_analysis(text,
ilm.load_islamic_lexicon("./data/islamic_lexicon.csv"))],
```

```
[analyze_sentiment(text))]
    sentimentnewestscore = constfull *
ilm.sentiment_analysis(text,
ilm.load_islamic_lexicon("./data/islamic_lexicon.csv")) +
constislam * analyze_sentiment(text)
    return sentimentnewestscore
def process_text(text):
    # Lakukan pemrosesan atau operasi lain pada teks
    processed_text = taa.full_analysis(text)
    if processed_text > 0:
        return "Positif"
    elif processed_text < 0:
        return "Negatif"
    else:
        return "Netral"
```

Fig. 12. Source code `text_analyzer_algo.py`

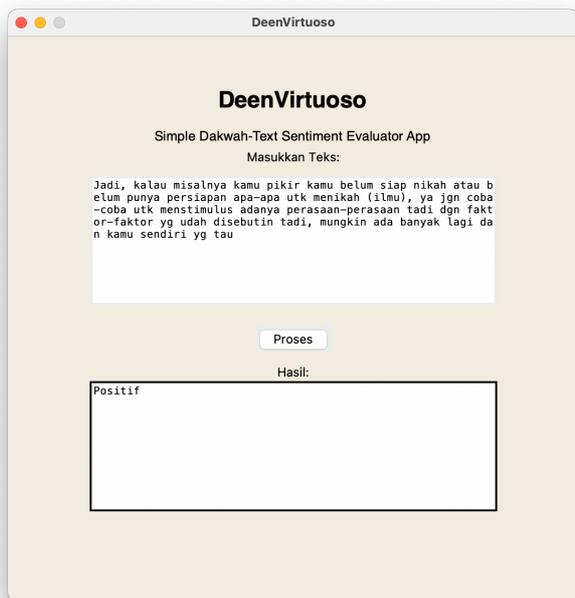
IV. HASIL PENGUJIAN

Berdasarkan program sederhana yang telah dibuat, dilakukan pengujian menggunakan dua teks dakwah yang memiliki tujuan sama, yakni mengingatkan terkait hukum berpacaran dalam islam, dengan menggunakan bahasa yang dan diksi yang berbeda, namun sama-sama berhasil memperoleh banyak *like* dan *comment* dengan sentimen yang berbeda pada twitter. Dua hal tersebut antara lain adalah sebagai berikut:

A. Teks dengan pilihan diksi negatif berhasil memperoleh penilaian negatif. Muballigh disarankan untuk memperbaiki pilihan diksi apabila memperoleh hasil seperti ini



B. Teks dengan pilihan diksi positif berhasil memperoleh penilaian positif. Tidak masalah untuk muballigh membagikan teks ini sebagai kalimat dakwah.



Dari dua hasil berikut, dapat dilakukan deduksi bahwa penilaian analisa sentimen telah terbukti akurat, sesuai dengan basis kemanusiaan sesuai dengan harapan penggunaan algoritma berbasis lexicon.

V. KESIMPULAN

DeenVirtuoso sebagai aplikasi sederhana yang menggenerasi leksicon baru berbasis islami sebagai sistem evaluasi sentimen menggunakan regex dan lexicon-based algorithm untuk teks dakwah pada platform digital menggunakan bantuan bahasa pemrograman Phyton berhasil melakukan analisis teks dakwah sesuai sentimen masyarakat sesungguhnya dengan skala keberhasilan 100%.

VIDEO LINK AT YOUTUBE

<https://bit.ly/DeenVirtuosoDemoVideo>

GITHUB LINK

<https://github.com/CettaReswara/DeenVirtuoso.git>

UCAPAN TERIMAKASIH

Penulis bersyukur kepada Tuhan Yang Maha Esa karena telah melimpahkan rahmat-Nya sehingga penulis mampu menyelesaikan makalah berjudul "DeenVirtuoso: Evaluasi Sentimen menggunakan Regex dan Lexicon-based Algorithm terhadap Dakwah Digital berbasis Teks". Penulis mengucapkan terima kasih kepada orang tua, bapak/ibu dosen, teman-teman, dan sahabat penulis yang selalu memberikan dukungan, motivasi, dan masukan sehingga penulis mampu menyelesaikan makalah dengan baik.

REFERENCES

- [1] Câmpeanu, C., Salomaa, K., & Yu, S. (2003). A FORMAL STUDY OF PRACTICAL REGULAR EXPRESSIONS. *International Journal of Foundations of Computer Science*, 14(06), 1007–1018. <https://doi.org/10.1142/s012905410300214x>
- [2] Contreras, J. L. G., Ballera, M. A., Lagman, A. C., & Raviz, J. (2018). *Lexicon-based Sentiment Analysis with Pattern Matching Application using Regular Expression in Automata*. <https://doi.org/10.1145/3301551.3301596>
- [3] Download citation of An integrated semi-automated framework for domain-based polarity words extraction from an unannotated non-English corpus. (n.d.). ResearchGate. https://www.researchgate.net/publication/339671931_An_integrated_semi-automated_framework_for_domain-based_polarity_words_extraction_from_an_unannotated_non-English_corpus/citation/download
- [4] Hadi, K. A., Lasri, R., & Abderrahmani, A. E. (2019). An Efficient Approach for Sentiment Analysis in a Big Data Environment. *ResearchGate*. https://www.researchgate.net/publication/342202276_An_Efficient_Approach_for_Sentiment_Analysis_in_a_Big_Data_Environment
- [5] M. Natsir, *dakwah dan pemikirannya*. (n.d.). Google Books. https://books.google.co.id/books?hl=en&lr=&id=22p8BWSShm_gC&oi=fnd&pg=PA5&dq=dakwah&ots=y3EPSEo2&sig=6D7KPV1HCc4XVNVkN7yT59y7j-I&redir_esc=y#v=onepage&q=dakwah&f=false
- [6] Mastori, M. (2021, July 27). *MENGAGAS ETIKA DAKWAH DI RUANG MEDIA SOSIAL*. <https://jurnal.lp2msasabel.ac.id/index.php/KPI/article/view/1810>
- [7] Samsir, S. (2021, January 22). *Analisis Sentimen Pembelajaran Daring Pada Twitter di Masa Pandemi COVID-19 Menggunakan Metode Naïve Bayes*. Samsir | JURNAL MEDIA INFORMATIKA BUDIDARMA. <http://stmik-budidarma.ac.id/ejurnal/index.php/mib/article/view/2580/1870>
- [8] Syed, M. (2022, January 6). Overview of Lexicon-based Approach in Identifying the Effects of Slang on Sentiment Analysis of Text. *Medium*. <https://medium.com/red-buffer/overview-of-lexicon-based-approach-in-identifying-the-effects-of-slangs-on-sentiment-analysis-of-75407ecc8ed5>
- [9] Taboada, M., Brooke, J., Tofiloski, M., Voll, K., & Stede, M. (2011). Lexicon-Based Methods for Sentiment Analysis. *Computational Linguistics*, 37(2), 267–307. https://doi.org/10.1162/coli_a_00049

PERNYATAAN

Dengan ini saya menyatakan bahwa makalah yang saya tulis ini adalah tulisan saya sendiri, bukan saduran, atau terjemahan dari makalah orang lain, dan bukan plagiasi.

Bandung, 22 Mei 2023

Cetta Reswara Parahita
13521133