# Analisis Sentimen Komentar Beracun pad Media Sosial Menggunakan Word2Vec dan *Support Vectore Machine* (SVM)

**Nurul Dian Kusumawati[1], Said Al Faraby[2], Mahendra Dwifebri P[3]**

[1,2,3] Universitas Telkom, Bandung
**[1]nuruldiankusumawati@student.telkomuniversity.ac.id, [2]saidalfaraby@telkomuniversity.ac.id, [3]mahendradp@telkomuniversity.ac.id**

**Abstrak**

**Media sosial berkembang secara eksponensial sejak tahun 2004 sebagai sebuah wadah untuk berdiskusi dan bertukar pendapat. Perkembangan ini mendapatkan antusiasme yang baik oleh warga negara Indonesia. Di mana dari 170 juta jiwa penduduk negara Indonesia menjadi pengguna aktif sosial media. Namun tujuan dari pembuatan media sosial sering sekali disalahgunakan untuk menyebarkan komentar beracun, seperti menyebarkan kebencian, pornografi, radikalisme, SARA, dan masih banyak lagi. Sehingga, akhir- akhir ini analisis sentimen terhadap komentar beracun pada media sosial sedang marak dilakukan. Maka dari itu pada penelitian ini akan dilakukan analisis sentimen terhadap komentar beracun pada media sosial untuk memilah komentar yang beracun atau tidak beracun berdasarkan label atau *multilabel*. Pada penelitian ini, digunakan metode Word2Vec sebagai ekstraksi fitur. Word2Vec sebagai ekstraksi fitur telah banyak diterapkan pada penelitian *Natural Language Processing* (NLP) dan menunjukkan dampak potensial yang tinggi pada kinerja analisis sentimen. Kemudian dilakukan klasifikasi menggunakan metode *Support Vector Machine* (SVM) untuk menemukan hasil terbaik. Hasil optimum yang dihasilkan dari beberapa pengujian yang dilakukan menunjukkan nilai F1-Score tertinggi sebesar 73,69% data terklasifikasi benar menggunakan Word2Vec sebagai ekstraksi fitur dan tanpa menggunakan *stemming* pada tahap *preprocessing*.**

**Kata kunci : Komentar beracun, Analisis sentimen, *Multilabel*, Word2Vec, *Support Vector Machine* (SVM).**

**Abstract**

**Social media has grown exponentially since 2004 as a forum for discussion and exchanging opinions. This development received good enthusiasm by Indonesian citizens. Where out of 170 million people in Indonesia are active users of social media. However, the purpose of making social media is often misused to spread toxic comments, such as spreading hate, pornography, radicalism, SARA, and many more. So, lately, sentiment analysis of toxic comments on social media is being carried out. Therefore, in this study, sentiment analysis will be carried out on toxic comments on social media to sort out toxic or non-toxic comments based on labels or *multilabels*. In this research, Word2Vec method is used as feature extraction. Word2Vec as feature extraction has been widely applied in *Natural Language Processing* (NLP) research and shows a high potential impact on sentiment analysis performance. Then the classification is carried out using the method *Support Vector Machine* (SVM)to find the best results. The optimum results obtained from several tests carried out showed the highest F1-Score value of 73.69% of the data classified correctly using Word2Vec as feature extraction and without using *stemming* at the *preprocessing*.**

**Keywords: Toxic Comment, Sentiment Analysis, *Multilabel*, Word2Vec, *Support Vector Machine* (SVM).**

## 1. Pendahuluan

### Latar Belakang

Pada era digital saat ini, perkembangan dunia teknologi informasi dan komunikasi sangat pesat, salah satunya adalah media sosial. Tanpa kita sadari, media sosial telah berkembang secara eksponensial sejak tahun 2004 sebagai wadah untuk berdiskusi dan mengungkapkan pendapat [1]. Perkembangan ini diimbangi dengan antusiasme masyarakat sehingga mempengaruhi budaya berpendapat yang biasanya dilakukan secara langsung namun kini dapat dilakukan secara virtual, melalui fitur komentar yang ada pada media sosial. Berdasarkan artikel "*Digital 2021 : The Latest Insights Into The State Of Digital*", Indonesia memiliki jumlah penduduk sekitar 274,9 juta jiwa dengan pengguna aktif media sosial mencapai 170 juta pada tahun 2021. Dengan demikian, angka presentasinya sekitar 61,8%. Di Indonesia aplikasi media sosial yang paling banyak digunakan adalah Youtube, WhatsApp, Instagram, Facebook, Twitter, dan lain sebagainya [2]. Namun tujuan pembuatan media sosial sering sekali disalahgunakan oleh pengguna, dengan menyebarkan opini atau komentar – komentar beracun, seperti menyebarkan kebencian, pencemaran nama baik, pornografi, radikalisme, SARA (Suku, Ras, dan Agama), dan lain sebagainya. Kini opini publik telah banyak digunakan untuk memahami sentimen terhadap suatu topik, seseorang, atau suatu produk [3]. Proses ini disebut sebagai analisis sentimen.

Analisis sentimen atau *opinion mining* merupakan bagian dari *text mining*, dimana pada proses ini akan dilakukan ekstrak, mengolah, dan memahami data yang berbentuk tekstual untuk mendapatkan informasi sentimen yang terkandung dalam suatu kalimat. Sentimen dapat diklasifikasi menjadi tiga kelas, yaitu kelas positif, kelas

negatif, dan kelas netral. Untuk mengklasifikasikan analisis sentimen berupa data teks diperlukan sebuah proses ekstraksi fitur. Ada beberapa metode ekstraksi fitur yang sering digunakan, yaitu Glove, TF-IDF, *Bag-Of-Word* (BOW), dan Word2Vec. Metode tersebut memiliki keunggulan dan kelemahan masing-masing. Seperti metode TF-IDF selain sering digunakan karena memiliki keunggulan mudah untuk diimplementasikan, namun juga memiliki kelemahan yaitu tidak dapat memproses relasi semantik antar kata sehingga menganggap setiap kata memiliki konteks yang berbeda [4]. Pada analisis sentimen, relasi semantik memberikan dampak berupa hasil klasifikasi yang lebih baik [5]. Ekstraksi fitur dengan relasi semantik dapat dilakukan dengan metode *word embedding*. Word2Vec merupakan salah satu metode *word embedding*, di mana word2vec mempelajari representasi kata pada ruang vektor dengan dimensi tinggi. Word2vec akan menghitung relasi semantik antar kata dengan merepresentasikan kata menjadi vektor – vektor angka [6]. Vektor - vektor yang dihasilkan pada Word2vec akan digunakan sebagai ekstraksi fitur. Pada klasifikasi analisis sentimen ada beberapa metode yang dapat digunakan, salah satunya adalah algoritma *support vector machine* (SVM). SVM merupakan metode yang paling akurat untuk klasifikasi dibandingkan dengan metode lain seperti Naive Bayes, dan Decision Tree [7].

**Topik dan Batasannya**

Pada penelitian ini data yang digunakan berasal dari [8], berisi kumpulan komentar beracun (*toxic*) dari Twitter, Kaskus, dan Twitter, *list stopword removal, list emojis translation,* dan *list slangword translation*. Metode yang akan digunakan pada penelitian ini adalah Word2vec dan *Support Vector Machine* (SVM). Metode diimplementasikan dengan menggunakan bahasa pemrograman *python*.

**Tujuan**

Tujuan dari penelitian ini adalah menganalisis faktor-faktor yang dapat mempengaruhi hasil performansi dari implementasi analisis sentimen komentar beracun pada media sosial menggunakan metode Word2Vec dan *Support Vector Machine* (SVM), yaitu seperti pengaruh penggunaan *stemming* sastrawi, *stopword removal*, *slangword translation*, pelatihan pada model Word2Vec, dan perbandingan metode klasifikasi sesuai dengan studi literatur rujukan.

**Organisasi Tulisan**

Penelitian yang penulis buat akan ditulis dan disusun sebagai berikut, bagian dua menjelaskan studi terkait. Pada bagian tiga akan dijelaskan *dataset* yang akan digunakan, dan rancangan sistem yang dibangun. Bagian empat akan menjelaskan evaluasi hasil pengujian dan analisis hasil pengujian. Pada bagian terakhir yaitu lima, akan menjelaskan kesimpulan dari penelitian ini serta saran untuk penelitian terkait selanjutnya.

## 2. Studi Terkait

Penelitian mengenai klasifikasi komentar beracun telah banyak di lakukan, seperti pada [9], [10]. Pada [9] menggunakan data berbahasa Urdu Romawi dengan TF-IDF dan *word embedding* sebagai metode ekstraksi fitur. Metode klasifikasi yang di gunakan adalah Multi Layer Perceptron (MLP) dan Random Forest (RF). Nilai F1 Score terbaiknya sebesar 96,6%. Kemudian pada [10] mengambil data dari Kaggle dengan membandingkan hasil performansi dari metode klasifikasi Long Short Term Memory (LSTM) dan Naive Bayes. Dimana hasil menunjukkan bahwa penggunaan metode LSTM lebih unggul daripada metode Naive Bayes, yaitu sebesar 64% dan 73%. Peneliti juga memberikan saran untuk penelitian selanjutnya agar menggunakan data multilabel dan mengoptimalkan metode klasifikasi menggunakan *Support Vector Clustering* (SVC) atau *Conventional Neural Networks* (CNN).

Namun pada penelitian-penelitian yang dijabarkan di atas masih menggunakan *single* label yaitu hanya membedakan antara komentar beracun dan tidak beracun. Berdasarkan data yang digunakan, penelitian ini akan di bangun menggunakan data multilabel. Sedangkan penelitian komentar beracun menggunakan data multilabel dan berbahasa Indonesia masih jarang dilakukan. Pada studi literatur [11] menggunakan data dari [8] dengan metode yang digunakan adalah TF-IDF dan Chi-Square sebagai seleksi fitur serta SVM sebagai metode klasifikasi. Penelitian tersebut menghasilkan F1 Score sebesar 76,57%. Peneliti juga memberikan saran agar penelitian selanjutnya menggunakan metode yang dapat menangani data *imbalance* seperti SMOTE atau metode yang lainnya, kemudian penggunaan model yang dapat memperkaya pengetahuan seperti Word2Vec, FastText, dan lain sebagainya. Pada studi literatur pada [12] juga dikatakan bahwa Word2Vec merupakan salah satu model umum yang dapat digunakan untuk *embedding* kata. Di mana algoritma berbasis prediksi yang digunakan untuk mempresentasikan sebuah kata sebagai vektor dengan relasi semantik. Word2Vec banyak diterapkan pada *Natural Language Processing* (NLP) dan menunjukkan dampak potensial yang tinggi pada kinerja analisis sentimen. Adapun penelitian analisis sentimen menggunakan metode

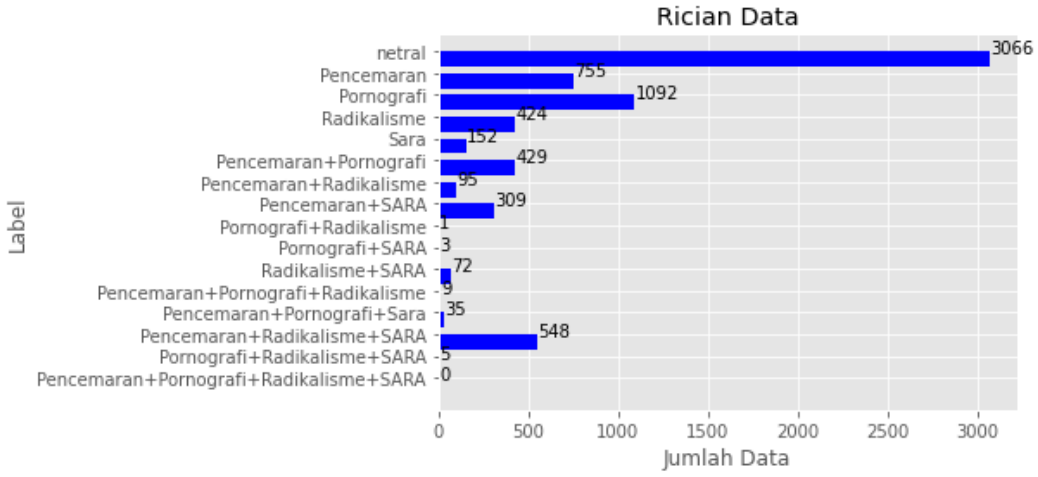
Word2Vec dan SVM, seperti [13]. Penelitian – penelitian tersebut menunjukkan hasil Akurasi yang tinggi pada penggunaan Skip-gram dan SVM yaitu sebesar 72%.

## 3. Sistem yang Dibangun

Penelitian ini dilakukan dengan membangun sistem yang dapat mengklasifikasi komentar beracun pada media sosial menggunakan metode Word2Vec dan *SVM*. Berikut merupakan penjelasan mengenai *dataset* yang digunakan dan skema sistem yang dibangun.

### 3.1 Dataset

Data ini dikumpulkan oleh Ahmad Izzan, Christian Wibisono dan Ilham Firdausi Putra [8] berjumlah 6995. Data tersebut merupakan kumpulan komentar beracun *multilabel* yang diambil dari beberapa sosial media, yaitu : Twitter, Instagram dan Kaskus. Data ini terdiri dari empat label berbeda, yaitu pornografi, SARA, radikalisme dan pencemaran nama baik, selain itu data ini mempunyai dua kelas, yaitu kelas 1 (positif) dan kelas 0 (negatif). Adapun pelabelan data dilakukan secara manual. Berikut merupakan rincian jumlah data pada *dataset*.




**Gambar 1. Jumlah Data Komentar Beracun**

Berikut contoh data komentar beracun dapat dilihat pada tabel 1.

**Tabel 1 Contoh Data Komentar Beracun**

| Komentar Toxic | P | S | R | N |
|---|---|---|---|---|
| Jihad tidak selalu harus dengan BOM. Pengertian Jihad bukanlah "bawa tas, bunuh diri biar org kafir ikut mati" "bawa bom, lempar kesono" ! | 0 | 1 | 1 | 1 |
| Meski hanya Show Of Force, preman kafir tetap lempari pemuda Islam Solo dengan batu dan bom molotov via @arrahmah http://j.mp/KteNOKÂ | 0 | 1 | 1 | 1 |
| ajil maaf ya ak gk bsa dtg di mng kmu krna ak skrng lgi di kota malang di rmh nenek ku maaf ya jil gk bsa dtg aku | 0 | 0 | 0 | 0 |
| Mending pro cina daripada pro arab. Arab antek Amerika. Dukung aja sono Islam arab, gw mah Islam liberal dan nusantaraðŸ˜Ž,instagram | 0 | 1 | 0 | 0 |

Keterangan :
N (Pencemaran Nama Baik) : komentar yang mengandung pencemaran nama baik seseorang, golongan, instansi dan sejenisnya.
P (Pornografi) : komentar yang mengandung hal – hal yang berhubungan dengan seksualitas.
R (Radikalisme) : komentar yang mengandung paham ekstrim terkait keinginan perubahan sosial dan politik.
S (SARA) : komentar yang di dalamnya mengandung ujaran yang menyinggung dengan suku, ras, agama maupun golongan.

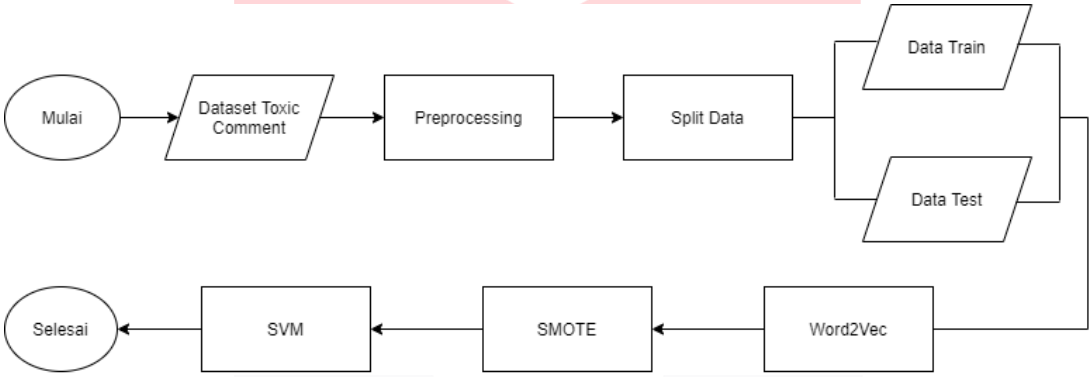
Adapun kata – kata yang sering muncul pada setiap label, dapat dilihat pada tabel 2.

**Tabel 2 Kata yang sering muncul**

| Label | Kata - Kata |
|---|---|
| Pencemaran | “kafir”, “cina”, “tolol” |
| Pornografi | “kont*l”, ‘m*m*k”, “t*t*k” |
| Radikalisme | “khalifah”, “negara”, “Indonesia” |
| SARA | “pribumi”, “rasulullah”, “teroris” |

3.2 Rancangan Sistem

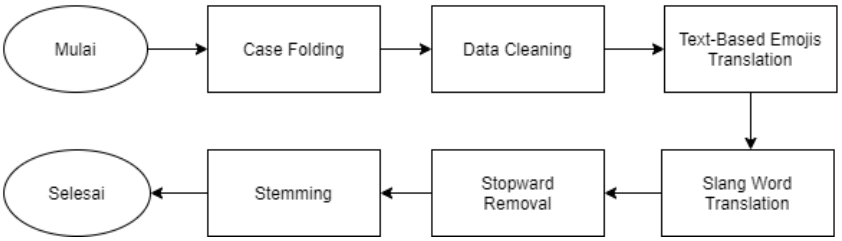
Dalam penelitian ini akad membangun suatu sistem yang dapat melakukan analisis sentimen terhadap komentar beracun pada media sosial menggunakan metode Word2vec sebagai ekstraksi fitur dan *Support Vectore Mahine* (SVM) sebagai metode klasifikasi. Alur perancangan sistem pada penelitian ini dapat dilihat diilustrasikan pada Gambar 1 :




**Gambar 2. Alur Perancangan Sistem**

3.3 *Preprocessing*

*Preprocessing* merupakan tahap awal sebelum melakukan klasifikasi data. *Preprocessing* merupakan suatu proses yang dilakukan untuk mengubah data yang berbentuk tekstual menjadi data yang lebih terstruktur, sehingga data – data tersebut dapat diolah pada tahapan selanjutnya [8]. Adapun beberapa proses *preprocessing* yang akan digunakan pada penelitian ini, yaitu *case foalding*, data *cleaning*, *text-based emojis translation*, *slangword translation*, *stopword removal*, dan *stemming*. Berikut merupakan ilustrasi tahapan dari proses *preprocessing*.




**Gambar 3. Alur Preprocessing**

- *Case Folding*

*Case Folding* merupakan suatu proses untuk mengubah semua karakter yang ada di dalam dokumen menjadi huruf kecil [14]. Karakter yang dapat diubah adalah huruf “a” hingga “z”, contohnya “A” menjadi “a” . Karakter selain huruf dihilangkan dan dianggap sebagai *delimiter*.

- Data *Cleaning*

Data *cleaning* merupakan proses untuk menghapus format bawaan pada setiap media sosial dan simbol – simbol yang ada pada *dataset*. Berikut atribut yang di hapus pada proses ini : format Instagram, format Twitter, format Kaskus, simbol, url, \n (new line), \\n, *whitecase*.

- *Text-Based Emojis Translation*

*Text-Based Emojis Translation* merupakan proses untuk menerjemahkan *emoticon* dalam suatu kalimat. Pada penelitian ini akan digunakan *list* emoticon dari [8]. Berikut merupakan contoh dari *text-based emojis translation* yang digunakan pada penelitian ini:

- :* artinya Cium.
- :-) artinya Bahagia.
- :| artinya Heran.

- *Slang Word Translation*

*Slang Word Translation* merupakan suatu proses mengubah kata tidak baku menjadi kata baku, seperti singkatan, kesalahan penulisan, bahasa tidak resmi dan penambahan unsur visual seperti angka. *Slang word transformation* yang digunakan pada penelitian ini berdasarkan kamus yang diambil dari [8]. Berikut merupakan contoh dari *slang word translation* pada penelitian ini:

- 7an menjadi Tujuan
- Asbun menjadi Asal bunyi
- Aing menjadi Saya

- *Stopword Removal*

*Stopward removal* merupakan proses yang digunakan untuk mengurangi *noise term* pada suatu kalimat. Dimana proses ini akan menghilangkan sekumpulan kata umum yang digunakan dan memiliki fungsi dalam teks namun tidak memiliki makna [15]. Pada penelitian ini *list stopword* yang digunakan berasal dari [8].

- *Stemming*

*Stemming* merupakan suatu proses bertujuan untuk membersihkan data dari kata imbuhan, awalan, sapaan, akhiran, dan kombinasi. Dengan demikian setiap kata pada dokumen hanya mengandung kata dasar saja. Proses ini berguna untuk memperkecil ragam kata yang ada di dalam data. Karena data yang digunakan berbahasa Indonesia maka *stemmer* yang digunakan pada penelitian ini adalah sastrawi.

Adapun ilustrasi *preprocessing* pada penelitian ini, dapat dilihat pada tabel 3

**Tabel 3 Ilustrasi Preprocessing**

| No. | Proses | Input | Output |
|---|---|---|---|
| 1. | *Case Folding* | @syukronamin justru dgn khilafah potensi korup lebih besar menurut saya. Karena harus taat dengan pimpinan (ulil amri) tdk boleh bantah :D | @syukronamin justru dgn khilafah potensi korup lebih besar menurut saya. karena harus taat dengan pimpinan (ulil amri) tdk boleh bantah :D |
| 2. | Data *Cleaning* | @syukronamin justru dgn khilafah potensi korup lebih besar menurut saya. karena harus taat dengan pimpinan (ulil amri) tdk boleh bantah :D | syukronamin justru dgn khilafah potensi korup lebih besar menurut saya. karena harus taat dengan pimpinan ulil amri tdk boleh bantah :D |
| 3. | *Text-Based Emojis Translation* | syukronamin justru dgn khilafah potensi korup lebih besar menurut saya. karena harus taat dengan pimpinan ulil amri tdk boleh bantah :D | syukronamin justru dgn khilafah potensi korup lebih besar menurut saya. karena harus taat dengan pimpinan ulil amri tdk boleh bantah tertawa |
| 4. | *Slang Word Translation* | syukronamin justru dgn khilafah potensi korup lebih besar menurut saya. karena harus taat dengan pimpinan ulil amri tdk boleh bantah tertawa | syukronamin justru dengan khilafah potensi korupsi lebih besar menurut saya. karena harus taat dengan pimpinan ulil amri tidak boleh bantah tertawa |
| 6. | *Stopword removal* | syukronamin justru dengan khilafah potensi korupsi lebih besar menurut saya. karena harus taat dengan pimpinan ulil amri tidak boleh bantah tertawa | syukronamin khalifah potensi korupsi taat pimpinan uli amri bantah tertawa |
| 7. | *Stemming* | syukronamin khalifah potensi korupsi taat pimpinan uli amri bantah tertawa | syukronamin khalifah potensi korupsi taat pimpinan uli amri bantah tawa |

3.4 *Split* Data

*Split* data merupakan metode yang dapat digunakan untuk mengevaluasi performa model. Metode ini akan membagi *dataset* menjadi dua bagian, yaitu data *training* dan data *testing* dengan proporsi yang ditentukan. Data train digunakan untuk *fit* model, sedangkan data test digunakan untuk mengevaluasi hasil *fit* model tersebut.

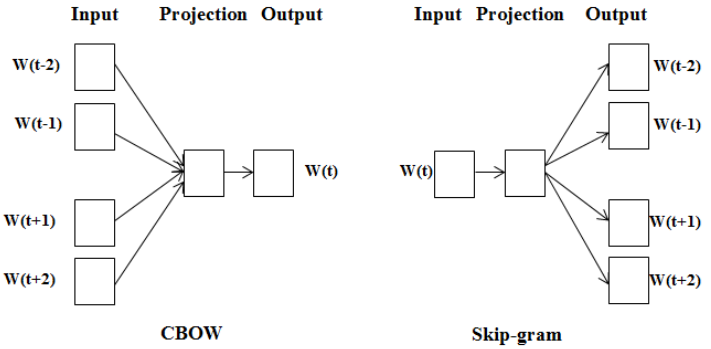
Proporsi yang digunakan dalam penelitian ini adalah 70% sebagai data *train* dan 30% sebagai data *test*. Berikut merupakan rincian jumlah data setelah proses split data, dapat dilihat pada tabel 4.

**Tabel 4 Persebaran Data Latih dan Data Uji**

| Label | Data *Train* | | Data *Test* | |
|---|---|---|---|---|
| | Positif | Negatif | Positif | Negatif |
| Pencemaran Nama Baik | 2143 | 1526 | 919 | 654 |
| Pornografi | 2143 | 1102 | 919 | 472 |
| Radikalisme | 2143 | 808 | 919 | 346 |
| SARA | 2143 | 787 | 919 | 337 |

3.5 Word2Vec

Word2vec adalah salah satu teknik terpopuler dan telah banyak digunakan [16]. Metode ini digunakan untuk merepresentasikan sebuah *teks* menjadi *vektor*, yang berbentuk seperti jaringan neural yang dangkal. Metode Word2Vec memiliki dua model, yaitu Skip-Gram dan *Continuous Bag of Word* (CBOW) [16]. Model Skip-Gram akan memprediksi sekumpulan konteks kata dari *input* satu kata lalu akan belajar untuk merepresentasikan vektor kata dari konteks kata yang saling berkaitan dan berdekatan [17], sedangkan model CBOW kebalikan dari model Skip Gram. Berikut merupakan ilustrasi dari kedua model dapat dilihat pada Gambar 4.




**Gambar 4. Arsitektur Word2Vec [16]**

Sesuai dengan studi literatur [13], pada penelitian ini akan digunakan Skip-gram sebagai arsitektur dari model. Berikut merupakan alur pembuatan model Word2Vec :

- Membangun konteks pasangan kata dari data korpus berdasarkan jumlah dimensi. Untuk penelitian ini akan menggunakan *window* sebesar 7 dan dimensi sebesar 100 dan 300 untuk mengetahui nilai kesamaan semantik apabila dimensi yang digunakan semakin besar.

**Tabel 5 Model Word2Vec**

| Model | Dimensi | Window |
|---|---|---|
| 1 | 100 | 7 |
| 2 | 300 | 7 |

- Kemudian dilakukan *training* untuk mengubah data tekstual menjadi bentuk *binary vector* melalui proses *one-hot-encoder*.
- Kemudian sistem akan memprediksi vektor *input* kata dengan melatih model berdasarkan koteks kata disekitarnya dengan *hidden layer* dan dimensi vektor.
- *Hidden layer* dihasilkan matriks *output*, kemudian matriks tersebut akan diubah dengan *Softmax function* untuk mendapatkan *Word Vector* yang akan diproses pada klasifikasi.

**3.6 SMOTE (*Synthetic Minority Oversampling Technique*)**

Pada penelitian ini *dataset* yang digunakan tidak setimbang antara kelas 1 dan 0, dimana data pada kelas 0 memiliki jumlah data lebih banyak daripada jumlah data pada kelas 1. Sehingga diperlukan SMOTE untuk menangani permasalahan ini. SMOTE adalah metode *over-sampling* yang bekerja dengan menambah jumlah data pada kelas minor agar setara dengan kelas mayor dengan cara membangkitkan data buatan. Data bantuan dibangun

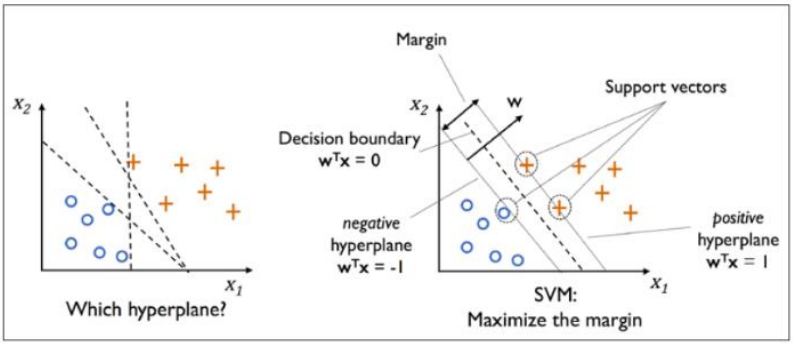
berdasarkan *k*-tetangga terdekat. Sehingga akan menghindarkan dari permasalahan *overfitting*. Berikut merupakan persamaan yang bekerja pada algoritma SMOTE :

$$Y_{new} = Y_i + (\hat{Y}_i - Y_i) \times \delta \quad (1)$$

Dimana $Y_i$ merupakan vektor dari fitur pada kelas minoritas, $\hat{Y}_i$ merupakan k-tetangga $Y_i$ , dan δ merupakan angka acak 0 sampai 1.

### 3.7 *Support Vector Machine* (SVM)

*Support Vector Machine* (SVM) merupakan salah satu metode klasifikasi *supervised*. Jika dilihat secara matematis, SVM memiliki konsep yang lebih baik dan jelas dibandingkan dengan teknik klasifikasi lainnya. Salah satu keunggulannya teknik ini dapat mengatasi permasalahan pada klasifikasi dan regresi baik secara *linear* maupun *non linear*. SVM memiliki empat *kernel*, seperti *kernel* linear, *kernel polynomial*, *kernel sigmoid*, dan *kernel* radial bias *function* (RBS) [18]. Berikut alur pembelajaran SVM yaitu dengan mencoba memberikan garis batas kepada dua buah kategori, atau yang biasa disebut *hyperplane* [7]. Pada SVM objek terluar yang paling dekat dengan *hyperplane* disebut sebagai *support vector*. Berikut merupakan ilustrasi dari *hyperplane* bekerja :




**Gambar 5. Cara Kerja Hyperplane**

Pada penelitian ini akan digunakan SVM *kernel* linear. Pemilihan *kernel* tersebut berdasarkan pada [11] yang memperoleh hasil terbaik pada penggunaan *kernel* linear. Berikut merupakan persamaan *kernel* linear sebagai berikut.

$$w^T.x + b = 0 \quad (2)$$

Di mana w merupakan vektor *weight* dan x adalah vektor dari *dataset*, sedangkan b adalah nilai bias. SVM menentukan *hyperplane* untuk memaksimalkan margin dan juga mengurangi *misklasifikasi* [7].

### 3.8 Evaluasi Akurasi

Analisis sentimen merupakan salah satu bentuk dari klasifikasi, sehingga hasil dari proses klasifikasi harus dievaluasi. Penelitian ini menggunakan data *multilabel*, proses klasifikasi yang dilakukan yaitu dengan menghitung nilai F1 Score pada setiap label kemudian dibagi dengan jumlah label. Cara yang biasa dilakukan untuk mengevaluasi yaitu dengan evaluasi matriks [19]. Evaluasi matriks terdiri dari *presisi*, *recall*, dan *f-score*. Hasil dari klasifikasi dibagi menjadi empat jenis, yaitu *True Positive* (TP), *True Negative* (TN), *False Positive* (FP), dan *False Negative* (FN). Keempat jenis hasil klasifikasi digunakan untuk menentukan *presisi*, *recall*, dan *f-score* seperti pada persamaan berikut.

$$precision = \frac{TP}{TP + FP} \quad (3)$$

Nilai *precision* digunakan sebagai penentu nilai prediksi yang bernilai positif terhadap semua prediksi positif.

$$recall = \frac{TP}{TP + FN} \quad (4)$$

Nilai *recall* digunakan sebagai penentu nilai prediksi benar bernilai positif terhadap semua data yang bernilai positif.

$$F1 = 2 \times \frac{recall \times precision}{recall + precision} \quad (5)$$

Nilai F1-Score digunakan untuk menentukan nilai perbandingan dari rata – rata *precision* dan *recall* yang di bobotkan.

## 4. Evaluasi

Pada penelitian ini dilakukan evaluasi dengan menggunakan lima skenario pengujian untuk mengetahui pengaruh dan hasil F1-Score terbaik. Skenario 1 dilakukan pengujian terhadap proses *preprocessing* dengan membandingkan pengaruh penggunaan *stopword removal* dan *stemming*, dimana akan dilakukan kombinasi proses *preprocessing* pada data, yaitu : *stopword removal* tanpa *stemming*, *stemming* tanpa *stopwords removal*, menggunakan *stemming* dan *stopword removal*. Kemudian pada skenario 2, dilakukan pengujian terhadap proses *preprocessing* untuk melihat pengaruh penggunaan *slang word removal*. Kemudian pada skenario 3 dilakukan pengujian terhadap model Wor2Vec (Skip-gram), dimana akan dilakukan perbandingan penggunaan korpus Wikipedia (besar) dan korpus data komentar beracun (kecil dan spesifik). Kemudian pada skenario 4 dilakukan perbandingan metode klasifikasi berdasarkan studi literatur rujukan [11] dan [13]. Kemudian yang terakhir akan dilakukan analisis terhadap data *missklasifikasi*.

### 4.1 Analisis Terhadap Pengujian Skenario 1

Pada skenario 1 pengujian dilakukan pada tahap *preprocessing* untuk mengetahui pengaruh penggunaan *stopword removal* dan *stemming*. Pada skenario ini akan dilakukannya perlakuan yang berbeda untuk *dataset* dengan mengombinasikan *stopword removal* dan *stemming*. Diantarainya yaitu *dataset* yang menggunakan *stopwords removal* dan *stemming*, *stopwords removal* tanpa *stemming*, *stemming* tanpa *stopwords removal*. Penggunaan *stemming* maupun *stopword* removal merupakan faktor yang penting dalam proses *preprocessing*. Dimana *stemming* berfungsi untuk menghilangkan imbuhan kata. Sedangkan *stopword removal* berfungsi untuk menghilangkan kata yang tidak bermakna yang tidak mewakilkan sebuah sentimen seperti “oleh”, “ke”, “adalah”, ”di” dan lain sebagainya. Jika proses *preprocessing* tanpa menggunakan proses *stopwords removal* maka kata yang tidak bermakna tersebut dapat dijadikan sebagai fitur dan mengakibatkan kesalahan dalam proses klasifikasi sehingga hasil performansi menjadi rendah. Berikut merupakan hasil pengujian 1 dapat dilihat pada tabel di bawah ini.

**Tabel 6 Hasil Terhadap Pengaruh Stemming**

| ***Preprocessing*** | **F1-Score** | **F1-Score *Max*** |
|---|---|---|
| *Stemming + Stopword Removal* | 73,01% | **73,69%** |
| Tanpa *Stemming* | 73,69% | |
| Tanpa *Stopword Removal* | 73,62% | |

Berdasarkan hasil pengujian skenario 1 pada tabel 6, menunjukkan bahwa hasil nilai F1-Score menggunakan data tanpa *stemming* lebih unggul dibandingkan dengan kombinasi yang lain. Dengan nilai F1-Score sebesar 73,69. Hal ini dapat terjadi karena *stemming* dapat mengubah makna kalimat, seperti “Penista agama menjadi santri kehormatan” menjadi “nista agama santri hormat”. Pada contoh tersebut kata yang dapat menjadi kata kunci label yaitu “kehormatan” di rubah menjadi “hormat” sehingga data tersebut dianggap *missklasifikasi* karena kata kunci telah berubah. Oleh sebab itu, proses *stemming* pada penelitian ini dapat menyebabkan *missklasifikasi* data dan nilai F1-Score menurun.

### 4.2 Analisis Terhadap Pengujian Skenario 2

Pada skenario 2 pengujian dilakukan pada tahap *preprocessing* untuk mengetahui pengaruh penggunaan *slangword translation*. Dimana *slangword translation* berfungsi sebagai pengubah kata tidak baku menjadi kata yang baku, seperti “ga”, “ngga”, “gak” yang berarti tidak, “aje” yang berarti saja. Selain itu *Slangword transation* juga dapat mengartikan kata – kata salah ketik, seperti “ngomng” yang berarti bicara. Pada penelitian ini digunakan *list salngword translation* dari [8] yang berjumlah 2.878. Berikut merupakan hasil pengujian 2 dapat dilihat pada tabel di bawah ini.

**Tabel 7 Hasil Penegaruh Terhadap Slangword Translation**

| ***Preprocessing*** | **F1-Score** | **F1-Score *Max*** |
|---|---|---|
| *Slangword Translation* | 73,69% | **73,69%** |
| Tanpa *Slangword Translation* | 72,25% | |

Berdasarkan hasil pengujian 2 pada tabel 7, menunjukkan penggunaan *slangword translation* memiliki pengaruh dalam proses *preprocessing*. Namun perbedaan hasil yang didapatkan masih kurang signifikan, yaitu 1,44%. Hal tersebut terjadi karena *list stopword removal* yang digunakan belum bisa mencakup semua kata-kata yang tidak baku pada *dataset*. Alangkah lebih baik jika *list* tersebut dapat dilengkapi agar performansi yang didapatkan lebih optimal.

4.3 Analisis Terhadap Pengujian Skenario 3

Pada skenario 3 pengujian dilakukan pada parameter dimensi Word2Vec. Berdasarkan hasil pengujian 1 dan 2, pada skenario ini data yang digunakan adalah data tanpa proses *stemming* namun menggunakan *slangword translation*. Parameter dimensi yang digunakan pada penelitian ini adalah 100 dan 300. Tujuan dari pengujian parameter dimensi pada Word2vec adalah untuk mengetahui nilai kesamaan semantik yang lebih unggul. Selain itu akan dilakukan perbandingan penggunaan data sebagai korpus, yaitu yang pertama data berasal dari Wikipedia dan yang kedua berasal dari data komentar beracun yang telah melalui proses *preprocessing*. Berikut merupakan hasil dari pengujian 3 dapat dilihat pada tabel di bawah ini.

**Tabel 8 Hasil Terhadap Pengujian Word2Vec**

| **Data** | ***Word2Vec*** | **F1-Score** | **F1-Score *Max*** |
|---|---|---|---|
| Wikipedia | 100 Dimensi | 73,04% | **73,69%** |
| | 300 Dimensi | 73,69% | |
| Komentar Beracun | 100 Dimensi | 57,41% | |
| | 300 Dimensi | 57,47% | |

Berdasarkan hasil pengujian skenario 3 pada tabel 8, penggunaan korpus yang lebih besar dan nilai parameter dimensi yang besar menghasilkan nilai performansi F1-Score yang tinggi, yaitu sebesar 73,69%. Hal ini dapat disebabkan karena perbedaan korpus dan model dapat menghasilkan nilai kedekatan kata yang berbeda-beda sehingga terdapat perbedaan nilai vektor untuk kata yang sama. Nilai vektor yang berbeda mempengaruhi data *input* untuk proses klasifikasi.

4.4 Analisis Terhadap Pengujian Skenario 4

Pada skenario 4 pengujian dilakukan dengan perbandingan metode sesuai pada [13]. Salah satu metode pembanding yang digunakan pada studi literatur tersebut adalah metode Naive Bayes. Berikut merupakan hasil perbandingan metode klasifikasi dapat dilihat pada tabel di bawah ini.

**Tabel 9 Pengujian Terhadap Metode Klasifikasi**

| **Metode** | ***Precision*** | ***Recall*** | **F1-Score** | **F1-Score *Max*** |
|---|---|---|---|---|
| SVM (linear) | 68% | 76,5% | 73,69% | **73,69%** |
| Naive Bayes (Gaussian) | 59,25% | 72% | 57,47% | |

Berdasarkan hasil pengujian skenario 4 pada tabel 9, penggunaan metode SVM memiliki nilai F1-Score paling tinggi yaitu sebesar 73,69%. Rentang nilai F1-Score yang dihasilkan memiliki oleh kedua metode tersebut sebesar 16,22%. Hal ini dapat dilihat dari tingkat keakuratan pencarian atau *precision* pada metode SVM yaitu sebesar 76,5%, dimana hasil tersebut lebih tinggi 4,5% daripada metode Naive Bayes. Sehingga pada *dataset* ini penggunaan metode klasifikasi SVM juga terbukti merupakan metode klasifikasi yang akurat seperti yang dipaparkan pada studi literatur [7].

4.5 Analisis Terhadap Data *Missklasifikasi*

Setelah melakukan pengujian terhadap beberapa skenario dari proses *preprocessing* hingga perbandingan metode klasifikasi, pada penelitian ini menunjukkan nilai F1-Score tertinggi adalah sebesar 73,69%. Maka selanjutnya akan dilakukan evaluasi terhadap data *missklasifikasi* untuk mengetahui persebaran dan permasalahan data *misklasifikasi* pada setiap label.

**Tabel 10 Pengujian Terhadap Train dan Test**

| Pengujian | F1-Score |
|---|---|
| Terhadap *Train* | **79,89%** |
| Terhadap *Test* | 73,69% |

**Tabel 11 Hasil Model Terbaik**

| Label | *Precision* | *Recall* | F1-Score | Jumlah Data *Missklasifikasi* | | |
|---|---|---|---|---|---|---|
| | | | | FP | FN | Jumlah |
| **Pencemaran Nama Baik** | **67%** | **73%** | **70%** | **229** | **179** | **408** |
| Pornografi | 77% | 68% | 81,1% | 122 | 67 | 189 |
| Radikalisme | 69% | 88% | 77,3% | 136 | 42 | 178 |
| SARA | 59% | 77% | 66,4% | 182 | 79 | 261 |

Berdasarkan pada tabel 10, selisih nilai F1-Score pada data *train* dan data *test* hanya sebesar 6,2%, hal tersebut menunjukkan bahwa perbedaan tidak terlalu jauh. Maka dari itu diperlukan analisis terhadap data *missklasikasi.* Berdasarkan tabel 11 dapat dilihat bahwa data *missklasifikasi* terbanyak ditemukan pada label pencemaran nama baik, hal ini disebabkan oleh banyaknya kata - kata pada label pencemaran yang kurang spesifik dengan label – label lainnya sehingga menyebabkan salah klasifikasi, seperti kata "cina" yang dapat masuk pada label SARA dan lain sebagainya. Lain halnya pada label pornografi yang memiliki kata – kata yang lebih spesifik dalam klasifikasi seperti "meremas", "susu", dan lain sebagainya. Penyebab lain dari *missklasifikasi* ini karena banyaknya penulisan menggunakan singkatan atau salah ketik yang dilakukan oleh pengguna sehingga sistem yang dibangun belum dapat mengklasifikasi dengan benar seperti kata "ngmong" yang seharusnya "ngomong". Meskipun pada penelitian ini sudah menggunakan *slangword translation*, namun kamus *slangword translation* yang digunakan masih belum bisa menangani masalah tersebut. Selain itu beberapa data juga ditemukan menggunakan bahasa daerah, seperti bahasa Jawa, Sunda, dan lain - lain sehingga model yang dibangun belum dapat mengatasinya.

## 5. Kesimpulan

Berdasarkan hasil pengujian dan analisis yang telah dilakukan pada analisis sentimen terhadap komentar beracun menggunakan data *multiabel* yang memiliki 4 label, yaitu pencemaran nama baik, pornografi, radikalisme, dan SARA dengan jumlah data keseluruhan adalah 6995. Dapat disimpulkan bahwa tahap *preprocessing*, penggunaan korpus dan dimensi pada Word2Vec, dan metode klasifikasi dapat mempengaruhi hasil performansi pada analisis sentimen komentar beracun pada media sosial. Hasil skenario terbaik didapatkan saat menggunakan tahap *preprocessing* dengan *slangword transation* namun tanpa *stemming*, menggunakan korpus Wikipedia dengan dimensi sebesar 300, dan metode klasifikasi sesuai studi literatur rujukan yaitu SVM *kernel* linear dengan nilai F1-Score terbaik sebesar 73,69.

Saran untuk penelitian selanjutnya dapat dilakukan penambahan *list slangword translation*, karena kamus *slangword translation* yang digunakan masih kurang mencakup semua istilah atau singkatan kata yang ada pada *dataset*. Kemudian dapat dilakukan pengujian terhadap model Word2Vec yang lebih cocok seperti mengubah parameter *window* dan dimensi untuk mendapatkan performansi dapat lebih tinggi. Serta dapat dilakukan percobaan menggunakan korpus Word2Vec yang lebih besar lagi.

## REFERENSI


[1] Nuruzzaman, M. 2018. Terorisme dan Media Sosial Sisi Gelap Berkembangnya Teknologi Informasi Komunikasi. Jurnal Ilmiah Indonesia. *3*(9), 61-76.

[2] C. Stephanie. 2021. Riset Ungkap Lebih dari Separuh Penduduk Indonesia "Melek" Sosial Media. [Online] Available at: https://tekno.kompas.com/read/2021/02/24/08050027/riset-ungkap-lebih-dari-separuh-penduduk-indonesia-melek-media-sosial [Accessed 20 April 2021].

[3] Sohrabi, M. K., & Hemmatian. 2019. Hemmatian. An efficient preprocessing method for supervised sentiment analysis by converting sentences to numerical vectors: a twitter case study. *Multimedia tools and applications*, 78(17), 24863-24882.

[4] Ramos, J. 2003. Using tf-idf to determine word relevance in document queries. *Proceedings of the first instructional conference on machine learning*. Vol. 242. No. 1. PP. 29-48.

[5] Zhang, D., Xu, H., Su, Z., & Xu, Y. 2015. Chinese comments sentiment classification based on word2vec and SVMperf. *Expert Systems with Applications*. *42*(4). 1857-1863.

[6] Tian, W., Li, J., & Li, H. 2018. A method of feature selection based on Word2Vec in text categorization. In *2018 37th Chinese Control Conference (CCC)* (pp. 9452-9455). IEEE.

[7] Bhavsar, H., & Ganatra, A. 2012. A comparative study of training algorithms for supervised machine learning. *International Journal of Soft Computing and Engineering (IJSCE)*, *2*(4), 2231-2307.

[8] Izzan, A., Wibisono, C., & Putra, I. F. (t.thn.). *Indonesian Social Media Post Toxicity*. [Online] Available at: https://github.com/ahmadizzan/netifier [Accessd 1 Oktoer 2020].

[9] Abbas, W. 2019. *Toxic Comment Classification of Roman Urdu Text.* Text (Doctoral dissertation. Department of Computer Science, COMSATS University Islamabad, Lahore Campus).

[10] Zaheri, S., Leath, J., & Stroud, D. (2020). Toxic comment classification. *SMU Data Science Review*, *3*(1), 13.

[11] Azzahra, N., Murdiansyah, D., & Lhaksmana, K. 2021. Toxic Comment Classification on Social Media Using Support Vector Machine and Chi Square Feature Selection. *International Journal on Information and Communication Technology (IJoICT)*, *7*(1), 64-76.

[12] Al-Saqqa, S., & Awajan, A. 2019. The use of word2vec model in sentiment analysis: A survey. In *Proceedings of the 2019 International Conference on Artificial Intelligence, Robotics and Control* (pp. 39-43).

[13] Acosta, J., Lamaute, N., Luo, M., Finkelstein, E., & Andreea, C. 2017. Sentiment analysis of twitter messages using word2vec. *Proceedings of Student-Faculty Research Day, CSIS, Pace University*, *7*, 1-7.

[14] Fauzi, M. A., Arifin, A. Z., & Yuniarti, A. 2015. Term Weighting Berbasis Indeks Buku Dan Kelas Untuk Perangkingan Dokumen Berbahasa Arab. *Lontar Komput*, *5*(2), 110-117.

[15] R. Arthana, "Stop Word Bahasa Indonesia dan implementasi pada Apache Lucene," [Online] Available at: http://www.rey1024.com/2012/06/stop-word-bahasa-indonesia-dan-implementasi-pada-apache-lucene/ [Accessd 25 Mei 2021].

[16] Mikolov, T., Chen, K., Corrado, G., & Dean, J. 2013. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*.

[17] Sakti, I. 2014. Methodology of fuzzy logic with mamdani fuzzy models applied to the microcontroller. In *2014 The 1st International Conference on Information Technology, Computer, and Electrical Engineering* (pp. 93-98). IEEE.

[18] Hsu, C. W., Chang, C. C., & Lin, C. J. (2003). A practical guide to support vector classification.

[19] Giachanou, A., & Crestani, F. (2016). Like it or not: A survey of twitter sentiment analysis methods. *ACM Computing Surveys (CSUR)*, *49*(2), 1-41.

**Lampiran**

Berikut merupakan perhitungan menggunakan metode Word2Vec :

- Mengonversi *input* dan *output* layer (target) menjadi *one-hot* vektor berukuran [V×1], dimana V merupakan banyak kata hasil *tokenization* atau jumlah *vocabulary*.
- Pembobotan atau *weight* terhadap *input layer* dengan *hidden layer* (W) dan *hidden layer* dengan *output* (W'). Matriks pada *input hidden layer* berukuran [N×V], dimana N merupakan jumlah neuron.

$$W^{\mathrm{T}} = \frac{h}{x} \tag{6}$$

$$W'^{\mathrm{T}} = \frac{h}{\mu_j} \tag{7}$$

- Input akan dikalikan dengan *weight input-hidden layer*, hasil dari proses ini disebut sebagai *hidden activation*.
- *Hidden activation* akan dikalikan dengan *hidden-output weight* untuk memperoleh *output*. Kemudian *output* tersebut ditransformasikan dengan fungsi *softmax*.

$$y_j = \frac{exp\ (\mu_j)}{\sum_{j'}^{V} exp\ (\mu_j)} \tag{8}$$

- Menghitung *error* antara *output* dan target kata yang dihitung, kemudian akan dilakukan *backpropagation* untuk *re-adjust* nilai *weight*.

$$e = \sum_{1}^{c} (y_j - x_c) \tag{9}$$

$$dl_{dW'} = h \times e^{\mathrm{T}} \tag{10}$$

$$dl_{dW} = x \times (W' \times e)^{\mathrm{T}} \tag{11}$$

Berikut merupakan ilustrasi perhitungan menggunakan model Skip-gram :

$$\underset{(V\times 1)}{X_1} \quad \underset{(V\times N)}{W_{input}} \quad \underset{(N\times 1)}{h_{x1}}$$

$$\begin{bmatrix}1\\0\\0\\0\\0\end{bmatrix} \times \begin{bmatrix}1 & 2 & 3\\4 & 5 & 6\\7 & 8 & 9\\10 & 11 & 12\\13 & 14 & 15\end{bmatrix}^{\mathrm{T}} = \begin{bmatrix}1\\2\\3\end{bmatrix}$$

$$\underset{(V\times 1)}{X_1} \quad \underset{(V\times N)}{W_{input}} \quad \underset{(N\times 1)}{h_{x2}}$$

$$\begin{bmatrix}0\\0\\1\\0\\0\end{bmatrix} \times \begin{bmatrix}1 & 2 & 3\\4 & 5 & 6\\7 & 8 & 9\\10 & 11 & 12\\13 & 14 & 15\end{bmatrix}^{\mathrm{T}} = \begin{bmatrix}7\\8\\9\end{bmatrix}$$

$$\underset{(N\times V)}{W_{output}} \quad \underset{(N\times 1)}{h_{avg}} \quad \underset{(V\times 1)}{W^{\mathrm{T}}{}_{output}\, h_{avg}} \quad \underset{(V\times 1)}{y_{pred}}$$

$$\begin{bmatrix}0{,}11 & 0{,}12 & 0{,}13 & 0{,}14 & 0{,}15\\0{,}16 & 0{,}17 & 0{,}18 & 0{,}19 & 0{,}20\\0{,}21 & 0{,}22 & 0{,}23 & 0{,}24 & 0{,}25\end{bmatrix}\mathrm{T} \times \begin{bmatrix}7\\8\\9\end{bmatrix} = \begin{bmatrix}2{,}5\\2{,}65\\2{,}8\\2{,}95\\3{,}1\end{bmatrix} \rightarrow \begin{bmatrix}0{,}145\\0{,}168\\0{,}196\\0{,}227\\0{,}264\end{bmatrix}$$

$$\underset{(V\times 1)}{y_{pred}} \quad \underset{(V\times 1)}{y_{target}} \quad \underset{(V\times 1)}{loss}$$

$$\begin{bmatrix}0\\1\\0\\0\\0\end{bmatrix} - \begin{bmatrix}0{,}145\\0{,}168\\0{,}196\\0{,}227\\0{,}264\end{bmatrix} = \begin{bmatrix}-0{,}145\\0{,}832\\-0{,}196\\-0{,}227\\-0{,}264\end{bmatrix}$$

Pada contoh pelatihan diatas digunakan satu kalimat toxic "Pak Andy Cindo yang pelit". Kalimat tersebut direpresentasikan terlebih dahulu dalam bentuk one-hot encoding. Kata diubah menjadi kumpulan angka di dalam

matriks. Untuk kata ”Cindo” diubah menjadi [1,0,0,0,0]T , dan untuk kata ”yang” diubah menjadi [0,1,0,0,0]T . Pada contoh diatas akan diprediksi kata ”pelit” dengan *input* kata ”Cindo” dan ”peit”. Menggunakan arsitektur Skip-gram dengan *window context* C=1, jumlah kata V=5 dan jumlah dimensi N=3. Nilai matriks *input* dan *output* diasumsikan untuk contoh kasus ini.