BAB II

LANDASAN TEORI

2.1 Cyberbullying

Cyberbullying didefinisikan sebagai sikap perundungan atau penindasan yang menggunakan teknologi digital sebagai *platform* perantara (Unicef, 2020). Hal ini dapat terjadi di media sosial, *platform chatting*, *platform online game*, dan sebagainya. Adapun menurut Think Before Text, *cyberbullying* adalah perilaku agresif dan bertujuan yang dilakukan suatu kelompok atau individu, menggunakan media elektronik, secara berulang-ulang dari waktu ke waktu, terhadap seseorang yang dianggap tidak mudah melakukan perlawanan atas tindakan tersebut (Unicef, 2020). Jadi, terdapat perbedaan kekuatan antara pelaku dan korban. Perbedaan kekuatan dalam hal ini merujuk pada sebuah persepsi kapasitas fisik dan mental. DSLA (2021) menjabarkan perilaku yang bersifat *cyberbullying* di Indonesia menjadi 6 jenis sebagai berikut.

a. *Flaming* – Tindakan seseorang mengirimkan pesan teks yang berisi kata-kata frontal dan penuh amarah. Secara umum, tindakan *flaming* berupa provokasi, penghinaan, mengejek, sehingga menyinggung orang lain.
b. *Harassment* – Tindakan seseorang mengirim pesan-pesan berisi gangguan melalui sms, e-mail, teks jejaring sosial dengan intensitas terus-menerus sehingga menyebabkan tekanan emosional bagi korban.
c. *Denigration* – Tindakan yang dilakukan sengaja dan sadar mengumbar keburukan orang lain melalui internet hingga dapat merusak nama baik dan reputasi orang yang dibicarakan pada jejaring sosial tersebut.

d. *Cyberstalking* – Tindakan memata-matai, mengganggu, dan pencemaran nama baik terhadap seseorang yang dilakukan secara intens. Dampaknya, orang yang menjadi korban merasakan ketakutan besar dan depresi.
e. *Impersonation* – Tindakan berpura-pura atau menyamar menjadi orang lain untuk melancarkan aksinya mengirimkan pesan-pesan dan status tidak baik. Biasanya terjadi pada jejaring sosial seperti Instagram dan Twitter dengan menggunakan akun palsu.
f. *Outing & Trickery* – *Outing* merupakan tindakan menyebarkan rahasia orang lain. *Outing* berupa foto-foto pribadi seseorang yang setelah disebarkan menimbulkan rasa malu atau depresi. Sementara itu, *Trickery* merupakan tindakan tipu daya yang dilakukan dengan membujuk orang lain untuk memperoleh rahasia maupun foto pribadi dari calon korban. Dalam banyak kasus, pelaku *Outing* biasanya juga melakukan *Trickery*.

Salah satu *platform* media sosial yang paling popular di Indonesia adalah Instagram. Pengguna yang dimulai dari anak-anak, remaja hingga orang dewasa turut mendongkrak popularitas Instagram. Namun, media sosial ini tidak lepas dari bahaya *hatespeech* penyebab perilaku *cyberbullying* yang sering dilakukan pada kolom komentar di unggahan seseorang yang diincarnya. Bukan hanya akun grup atau sekelompok saja, tak jarang pengguna Instagram berasal dari individu yang rela menggunakan akun palsu untuk mem-*bully* orang yang dituju. Berdasarkan survei dari 10.000 sampel dengan rentang usia 12 hingga 20 tahun, sebesar 42% remaja mengaku pernah menjadi korban *cyberbullying* di Instagram (Bohang, 2017). Bahaya perilaku *cyberbullying* tentunya meresahkan banyak orang dikarenakan dampak yang sangat berpengaruh bagi korban yang mengalaminya.

Maka dari itu, penting dilakukan suatu analisis sentimen pada komentar di media sosial yang berupaya untuk membantu menangani penyebaran konten di Instagram yang mengandung unsur *cyberbullying.* Think Before Text (2020) menjabarkan dampak dari perilaku *cyberbullying* dapat mempengaruhi seseorang dengan berbagai cara antara lain (Unicef, 2020):

a. Dampak bagi Korban:
    - Dampak psikologis: mudah depresi, marah, timbul perasaan gelisah, cemas, menyakiti diri sendiri, dan percobaan bunuh diri
    - Dampak sosial: menarik diri, kehilangan kepercayaan diri, lebih agresif kepada teman dan keluarga
    - Dampak akademik: penurunan prestasi akademik, rendahnya tingkat kehadiran, perilaku bermasalah di sekolah.

b. Dampak bagi Pelaku:

    Cenderung bersifat agresif, berwatak keras, mudah marah, impulsif, lebih ingin mendominasi orang lain, kurang berempati, dan dapat dijauhi oleh orang lain.

c. Dampak bagi yang menyaksikan (*bystander*):

    Jika *cyberbullying* dibiarkan tanpa tindak lanjut, maka orang yang menyaksikan dapat berasumsi bahwa *cyberbullying* adalah perilaku yang diterima secara sosial. Dalam kondisi ini, beberapa orang mungkin akan bergabung dengan penindas karena takut menjadi sasaran berikutnya dan beberapa lainnya mungkin hanya akan diam saja tanpa melakukan apapun hingga merasa tidak perlu menghentikannya.

2.2 Text Classification

Text classification (klasifikasi pada teks) bisa disebut juga sebagai *text categorization* (kategorisasi pada teks) atau *text tagging* (penandaan pada teks) yang dapat digunakan untuk mengatur, menyusun, dan mengkategorikan hampir semua jenis teks dari suatu dokumen (MonkeyLearn, 2020). Klasifikasi teks adalah salah satu tugas mendasar dalam *Natural Language Processing* (NLP) atau pemrosesan bahasa natural. Klasifikasi teks akan mengklasifikasikan teks ke dalam grup yang telah ditentukan sebelumnya untuk secara otomatis menyortir dan menganalisis informasi secara tekstual. Beberapa model klasifikasi teks yang paling populer mencakup analisis sentimen, pelabelan topik, deteksi spam, klasifikasi bahasa dan sebagainya (Roldós, 2019).

MonkeyLearn (2020) menjelaskan bahwa *text classification* sangat penting untuk digunakan karena untuk memperoleh suatu informasi, sebagian besar data yang dikumpulkan tidak terstruktur dengan baik. Karena sifat teks yang berantakan, sehingga cukup sulit dan memakan banyak waktu untuk menganalisis, memahami, mengatur, dan menyortir suatu data teks. Pada dasarnya, *text classification* dibutuhkan dalam kehidupan sehari-hari di lembaga pekerjaan seperti perusahaan, sekolah, mal, dan sebagainya yang pastinya memerlukan banyak data hingga informasi yang meluas. Sebagai salah satu contoh pada perusahaan, penerapan pengklasifikasian teks dengan *machine learning* dapat secara otomatis menyusun semua jenis teks yang relevan, dari *email*, dokumen hukum, media sosial, *chatbot*, survei, dan lainnya dengan cara yang cepat dan hemat biaya. Oleh karena itu, hal ini memungkinkan perusahaan untuk dapat menghemat waktu dalam menganalisis

data teks, mengotomatiskan proses bisnis, dan membuat keputusan bisnis berdasarkan data secara efisien.

MonkeyLearn (2020) mengungkapkan beberapa alasan utama menggunakan pengklasifikasian dengan basis *machine learning*, yakni sebagai berikut.

- *Scalability*

  Menganalisis dan mengatur suatu dokumen secara manual sangat memperlambat dan hasil yang diperoleh tidak akurat. Pembelajaran mesin dapat secara otomatis menganalisis jutaan survei, komentar, email, dll. dengan biaya yang sangat murah dan waktu yang cepat.

- *Real-time Analysis*

  Salah satu contoh kasus dimana ada situasi kritis yang perlu diidentifikasi suatu perusahaan secepatnya dan mengambil tindakan dengan segera (misalnya, krisis PR di media sosial). Klasifikasi teks dengan pembelajaran mesin dapat mengikuti *brand mentions* secara terus-menerus di dalam waktu nyata, sehingga akan mengidentifikasi informasi penting dan dapat segera mengambil tindakan.

- *Consistent Criteria*

  Anotator manusia membuat kesalahan saat mengklasifikasikan data teks karena gangguan, kelelahan, dan kebosanan, dan subjektivitas manusia membuat kriteria yang tidak konsisten. Pembelajaran mesin menerapkan lensa dan kriteria yang sama untuk semua hasil dan data. Setelah model klasifikasi teks dilatih dengan benar, model tersebut akan berfungsi dengan akurasi yang tak tertandingi.

2.3 Text Preprocessing

Text preprocessing adalah proses otomatis menganalisis dan menyortir data teks tidak terstruktur untuk mendapatkan *insights* yang berharga (Roldós, 2019). Dengan menggunakan *Natural Language Processing* (NLP), *machine learning* dan *artificial intelligence*, pemrosesan teks dapat secara otomatis memahami bahasa manusia dan mengekstrak nilai dari suatu data teks (Roldós, 2019). Secara singkat, proses ini dapat mengubah teks menjadi bentuk yang lebih mudah dicerna sistem sehingga algoritma pembelajaran mesin dapat bekerja lebih baik. Berikut ini proses penjelasan tahap *text preprocessing* (Cheng, 2020):

a. *Case Folding*

Proses pengubahan karakter huruf menjadi *lowercase* (huruf kecil) sehingga menyerupai keseragaman.

b. *Noise Removal*

Proses penghilangan digit karakter untuk mengurangi *noise* pada proses pengklasifikasian. Karakter yang perlu dihilangkan seperti angka, tanda baca dan sebagainya selain huruf alfabet yang dianggap sebagai tidak berpengaruh pada pemrosesan teks.

c. *Tokenization*

Proses memisahkan setiap kata yang menyusun suatu teks atau dokumen. Tahap ini menerima masukan serangkaian karakter dan menghasilkan deretan simbol yang masing-masing ditandai sebagai token. Tahap ini juga dilakukan berdasarkan pemotongan *string input* berdasarkan setiap kata penyusunnya.

d. *Filtering*

Proses pemilihan dan pengambilan kata-kata penting (umum) dari hasil token. Biasa disebut juga proses *stopword removal*. Pada proses ini kata umum akan dihapus untuk mengurangi jumlah kata yang disimpan oleh sistem. Metode yang digunakan adalah Stoplist yang dapat membuang kata-kata yang tidak deskriptif dengan pendekatan *bag-of-words*.

e. *Stemming*

Proses pengubahan bentuk kata yang berimbuhan menjadi kata dasar. Proses ini bertujuan untuk mencari *stem* (kata dasar) dari hasil *stopword removal* (*filtering*). Metode yang digunakan adalah Porter Stemmer yang lebih cocok untuk membuang imbuhan bahasa Indonesia dan bahasa Inggris.

2.4 Term Frequency-Inverse Document Frequency (TF-IDF)

Pada tahap ini akan dilakukan pembobotan setiap *term* berdasarkan tingkat kepentingan istilah dalam satu set data input. Metode yang akan digunakan untuk *term weighting* dalam penelitian ini adalah *Term Frequency-Inverse Document Frequency* (TF-IDF). TF-IDF adalah pembobotan statistik yang mengevaluasi seberapa relevan suatu kata dengan teks dalam kumpulan dokumen (Stecanella, 2019). Metode ini memiliki banyak kegunaan, yang paling penting dalam analisis teks otomatis, dan sangat berguna untuk menilai kata-kata dalam algoritma pembelajaran mesin untuk *Natural Language Processing* (NLP) (Stecanella, 2019).

TF-IDF berfungsi dengan meningkatkan secara proporsional dengan berapa kali sebuah kata muncul dalam dokumen tetapi diimbangi dengan jumlah dokumen yang berisi kata tersebut (Stecanella, 2019). Jadi, kata-kata yang umum di setiap

dokumen, seperti ini, apa, dan jika, berperingkat rendah meskipun mungkin muncul berkali-kali karena tidak terlalu berarti bagi dokumen tersebut secara khusus (Stecanella, 2019). TF-IDF untuk sebuah kata dalam dokumen dihitung dengan mengalikan dua metrik yang berbeda (Stecanella, 2019):

- *Term Frequency* (TF) atau frekuensi istilah suatu kata dalam dokumen. Ada beberapa cara untuk menghitung frekuensi ini, yang paling sederhana adalah hitungan mentah dari sebuah kata yang muncul dalam dokumen. Lalu, ada cara untuk menyesuaikan frekuensi, dengan panjang dokumen, atau dengan frekuensi mentah dari kata yang paling sering digunakan dalam dokumen.
- *Inverse Document Frequency* (IDF) atau frekuensi dokumen terbalik dari kata tersebut di seluruh kumpulan dokumen. Ini menjelaskan seberapa umum atau jarang sebuah kata di seluruh kumpulan dokumen. Semakin dekat dengan 0, semakin umum sebuah kata. Metrik ini dapat dihitung dengan mengambil jumlah dokumen lalu, membaginya dengan jumlah dokumen yang mengandung kata, dan menghitung logaritma.

Jadi, jika kata tersebut sangat umum dan muncul di banyak dokumen, angka ini akan mendekati 0. Jika tidak, akan mendekati 1. TF-IDF dapat diformulasikan sebagai berikut.

$$TF - IDF(w, d) = TF(w, d)(\log\left(\frac{N}{DF(w)}\right)) \qquad \ldots(2.1)$$

Dimana TF-IDF *(w, d)* adalah bobot sebuah kata di seluruh dokumen. *w* adalah kata. *d* adalah dokumen. *TF (w, d)* adalah frekuensi kemunculan kata *w* dalam

dokumen *d*. IDF *(w)* merupakan kebalikan DF dari kata *w*. *n* merupakan jumlah total dokumen. DF *(w)* adalah banyaknya dokumen yang mengandung kata *w*.

2.5 Naïve Bayes Classifier (NBC)

Naïve Bayes Classifier (NBC) adalah kumpulan algoritma klasifikasi berdasarkan Teorema Bayes. Teorema Bayes menemukan banyak kegunaan dalam teori probabilitas dan statistik (Gupta, 2017). Adanya algoritma ini, dapat mengetahui distribusi probabilitas variabel dalam kumpulan data dan memprediksi probabilitas variabel respons yang termasuk dalam nilai tertentu, mengingat atribut dari *instance* baru (Gupta, 2017). Teori ini memungkinkan untuk menguji probabilitas suatu peristiwa berdasarkan pengetahuan sebelumnya dari setiap peristiwa yang terkait dengan peristiwa sebelumnya (Gupta, 2017). Teorema Bayes dinyatakan secara matematis sebagai berikut (Gupta, 2017):

$$P(A|B) = \frac{P(B|A)P(A)}{P(B)} \quad \ldots(2.2)$$

atau

$$Posterior = \frac{Likelihood \times Prior}{Evidence} \quad \ldots(2.3)$$

Dimana:

- A dan B disebut kejadian.
- P(A|B) adalah peluang *posterior* kejadian A, jika kejadian B adalah benar (telah terjadi) setelah bukti B terlihat.
- P(B|A) adalah peluang *likelihood* kejadian B, jika kejadian A adalah benar (telah terjadi) setelah bukti A terlihat.

- P(A) adalah peluang *prior* dari A (probabilitas independen sebelumnya, yaitu probabilitas kejadian sebelum bukti terlihat).
- P(B) adalah peluang *evidence* konstanta normalisasi atau bukti.

NBC bekerja dengan sangat baik dalam situasi yang kompleks, terlepas dari asumsi dan kenaifan yang disederhanakan. Keuntungan dari pengklasifikasi ini adalah bahwa mereka memerlukan sejumlah kecil data pelatihan untuk memperkirakan parameter yang diperlukan untuk klasifikasi. Algoritma ini adalah algoritma pilihan untuk kategorisasi teks. Menurut Sklearn diketahui ada beberapa varian model dari algoritma NBC, antara lain: MultinomialNB, GaussianNB, BernoulliNB, ComplementNB, CategoricalNB, dan Out-of-coreNB. Kegunaan NBC menurut Widianto (2019) adalah dapat mengklasifikasikan dokumen teks seperti teks berita ataupun teks akademis, sebagai metode pembelajaran mesin yang menggunakan probabilitas, membuat diagnosis medis secara otomatis, mendeteksi atau menyaring spam, dan sebagainya. Adapun kelebihan penggunaan algoritma Naïve Bayes sebagai *classifier* adalah sebagai berikut (Widianto, 2019).

- Bisa dipakai untuk data kuantitatif maupun kualitatif.
- Bisa digunakan untuk klasifikasi masalah biner ataupun *multiclass.*
- Tidak memerlukan jumlah data yang banyak.
- Tidak perlu melakukan *data training* yang banyak.
- Jika ada nilai yang hilang, maka bisa diabaikan dalam perhitungan.
- Jika digunakan dalam bahasa pemrograman, kodenya sederhana.
- Perhitungannya cepat, efisien, mudah dipahami dan mudah dibuat.
- Pengklasifikasian dokumen bisa dipersonalisasi, disesuaikan dengan kebutuhan setiap orang.

2.6 Multinomial Naïve Bayes (MNB)

Multinomial Naïve Bayes (MNB) merupakan salah satu model yang memperluas penggunaan dari algoritma Naïve Bayes yang dirancang untuk menentukan frekuensi *term* atau berapa kali sebuah *term* muncul dalam suatu teks atau dokumen. Dalam penerapannya, MNB cocok untuk data yang didistribusikan secara multinomial dan merupakan model berbasis frekuensi yang diusulkan untuk klasifikasi teks atau dokumen, di mana jumlah kata digunakan untuk merepresentasikan data (Wiratama dan Rusli, 2019). Metode untuk meningkatkan kinerja Naïve Bayes diklasifikasikan ke dalam lima kategori, yaitu ekstensi struktur, pemilihan fitur, pembobotan atribut, pembelajaran lokal, dan perluasan data (Wiratama dan Rusli, 2019).

Harjito, dkk (2019) telah mencoba untuk melakukan pengkategorian dokumen berbahasa inggris menggunakan algoritma klasifikasi MNB yang dibandingkan dengan K-Nearest Neighbor (KNN), Support Vector Machine (SVM) Linear dan Random Forest. Proses klasifikasi dilakukan dengan menggunakan beberapa metode ekstraksi fitur, seperti ekstraksi Term Frequency-Inverse Document Frequency (TF-IDF), Count Vector, dan Document Vector (Doc2vec). Hasil percobaan menunjukkan bahwa MNB dengan TF-IDF mendapatkan akurasi yang cukup baik sebesar 76,00% sedangkan KNN dan SVM sebesar 72,66% dan 78,66%. Diketahui pula hasil akurasi MNB dengan Count Vector memperoleh akurasi sebesar 77% sedangkan KNN dan SVM sebesar 60% dan 70,66%. Dari penelitian tersebut menyimpulkan bahwa metode MNB dapat bekerja lebih baik dalam mengklasifikasikan dokumen dibandingkan dengan metode KNN dan SVM. MNB dapat diformulasikan sebagai berikut (Song, dkk., 2017).

$$c(d) = \arg \max_{c \epsilon C} [\log P(c) + \sum_{i=1}^{m} WT_i \, f_i \log P(w_i|c)] \quad \ldots(2.4)$$

Dimana *m* adalah jumlah kata yang berbeda dalam dokumen, w_i (i = 1, 2, …, m) adalah kata ke *i* yang muncul di dokumen, f_i (i = 1, 2, …, m) adalah frekuensi w_i di *d*, dan WT_i adalah bobot tiap kata w_i (i = 1, 2, …, m). Dalam MNB, probabilitas sebelumnya *P(c)* dapat dihitung dengan persamaan sebagai berikut (Song, dkk., 2017).

$$P(c) = \frac{\sum_{j=1}^{n} \delta(cj, c) + 1}{n + 1} \quad \ldots(2.5)$$

Selain itu, probabilitas bersyarat $P(w_i|c)$ bisa diperkirakan dalam persamaan sebagai berikut (Song, dkk., 2017).

$$P(w_i|c) = \frac{\sum_{j=1}^{n} WT_i f_{ji} \, \delta(cj, c) + 1}{\sum_{i=1}^{m} \sum_{j=1}^{n} WT_i f_{ji} \, \delta(cj, c) + m} \quad \ldots(2.6)$$

Dimana *n* adalah jumlah dokumen pelatihan, *l* adalah jumlah kelas, f_{ji} adalah frekuensi w_i dalam dokumen pelatihan ke-*j*, c_j adalah kelas dari dokumen pelatihan ke-*j*.

2.7 Confusion Matrix

Confusion matrix atau *error matrix* merupakan tabel yang sering digunakan untuk menggambarkan kinerja suatu model klasifikasi pada suatu set data uji yang nilai kebenarannya telah diketahui (Nugroho, 2019). Kinerja suatu model biasanya dievaluasi menggunakan data dalam matriks. Matriks ini berisi informasi yang

merepresentasikan dua kelas dimensi yang terdiri dari nilai aktual dan nilai prediksi dari suatu algoritma klasifikasi (Chaurasia, 2020). Tabel 2.1 berikut ini menunjukkan *confusion matrix* untuk pengklasifikasi dua kelas yang dapat dilihat sebagai berikut.

Tabel 2.1 Confusion Matrix

Confusion Matrix		**Predicted**	
		Positive (1)	Negative (0)
Actual	Positive (1)	True Positive (TP)	False Negative (FN)
	Negative (0)	False Positive (FP)	True Negative (TN)

Berdasarkan Tabel 2.1 istilah-istilah pada matriks diatas dapat dijabarkan sebagai berikut (Chaurasia, 2020).

- *True Positive* (TP) merupakan kondisi dimana kelas prediksi dan kelas aktual suatu model klasifikasi bernilai positif untuk keduanya.
- *False Positive* (FP) merupakan kondisi dimana suatu model klasifikasi mendapatkan nilai negatif untuk kelas prediksi dan nilai positif untuk kelas aktual.
- *True Negative* (TN) merupakan kondisi dimana kelas prediksi dan kelas aktual suatu model klasifikasi bernilai negatif untuk keduanya.
- *False Negative* (FN) merupakan kondisi dimana suatu model klasifikasi mendapatkan nilai positif untuk kelas prediksi dan nilai negatif untuk kelas aktual.

Setelah mengetahui istilah-istilah nilai pada *confusion matrix*, dapat dilakukan pengukuran dan perhitungan istilah secara matematis untuk mengevaluasi kinerja algoritma klasifikasi seperti *accuracy*, *precision, recall* dan *f1-score*. *Accuracy* didefinisikan sebagai proporsi dari seluruh jumlah prediksi yang dinilai benar. Perhitungan *accuracy* dapat dilihat pada rumus sebagai berikut (Chaurasia, 2020).

$$Accuracy\ (a) = \frac{TP + TN}{TP + FN + TN + FP} \quad \ldots(2.7)$$

Precision didefinisikan sebagai proporsi rasio dari seluruh nilai positif yang diklasifikasikan dengan benar dengan seluruh nilai positif yang diprediksi. Perhitungan *precision* dapat dilihat pada rumus sebagai berikut (Chaurasia, 2020).

$$Precision\ (p) = \frac{TP}{TP + FP} \quad \ldots(2.8)$$

Recall didefinisikan sebagai proporsi rasio dari seluruh nilai positif yang diprediksi dengan benar untuk seluruh nilai positif di kelas actual. Perhitungan *recall* dapat dilihat pada rumus sebagai berikut (Chaurasia, 2020).

$$Recall\ (r) = \frac{TP}{TP + FN} \quad \ldots(2.9)$$

F1-score didefinisikan sebagai gabungan *weighted average* dari *precision* dan *recall.* Perhitungan *f1-score* dapat dilihat pada rumus sebagai berikut (Chaurasia, 2020).

$$F1 - score = \frac{2(p \times r)}{p + r} \quad \ldots(2.10)$$

2.8 N-gram

N-gram banyak digunakan dalam *text mining* dan *natural language processing* (NLP). Pada dasarnya, *n-gram* merupakan sekumpulan kata yang muncul bersamaan dalam paragraf tertentu (Ganesan, 2020). Saat menghitung *n-gram*, biasanya akan memindahkan satu kata maju ke depan. Ketika mengembangkan model bahasa, *n-gram* digunakan untuk mengembangkan berbagai jenis model seperti *unigram*, *bigram* dan *trigram* (Ganesan, 2020). Penggunaan lain dari *n-gram* adalah untuk mengembangkan fitur untuk model pembelajaran mesin yang diawasi seperti SVM, MaxEnt, Naive Bayes, dll (Ganesan, 2020). Pokoknya adalah menggunakan token seperti *bigram* bukan hanya *unigram* di ruang fitur. Salah satu keuntungan dengan menggunakan *n-gram* adalah probabilitas kemunculan kata-kata tertentu dalam urutan tertentu dapat meningkatkan prediksi pada sistem otomatis (Ganesan, 2020). Untuk menghitung berapa banyak *n-gram* dalam sebuah kalimat dapat dilihat pada rumus sebagai berikut (Ganesan, 2020). Dimana X adalah jumlah kata dalam kalimat K yang diberikan.

$$N - gram_K = X - (N - 1) \quad \ldots(2.11)$$