

## BAB II. LANDASAN TEORI

### 2.1 Studi Literatur

Beberapa penelitian yang dijadikan rujukan dalam penelitian ini adalah sebagai berikut:

Pemanfaatan bot sebagai *filtering* kata pernah dilakukan oleh (Elvina et al., 2019) dengan mengkonsepkan sebuah *chatbot* yang dapat mem-*filter* pesan tidak penting sesuai dengan hasil klasifikasi *Naïve Bayes* dalam grup chat. Penelitian ini menghasilkan tingkat *precision* sebesar 94.2% dan tingkat *recall* sebesar 95.6% dengan jumlah datasetnya tidak seimbang antara pesan penting dan pesan tidak penting. Ketika jumlah datasetnya seimbang, sistem menghasilkan tingkat *precision* dan *recall* yang rendah, dengan nilai 70.1% dan 59.2%. Hal tersebut terjadi karena terdapat jarak berlebih antar kata (kelebihan spasi) pada kedua dataset.

Selain itu, sistem *filter* untuk kata non baku yang dirancang oleh (Hartina & Masri, 2020) berhasil mendeteksi kata non baku pada karya ilmiah dengan menerapkan metode klasifikasi teks *N-gram*. Dikarenakan metode ini dapat menghitung persentase bobot untuk memastikan kata mana yang akan dideteksi atau tidak. Sekitar 70% dari 10 kata berhasil dideteksi sistem dan 80% keberhasilan dari karya ilmiah.

Penelitian lain yang menerapkan metode *N-gram* juga dilakukan oleh (Enda et al., 2019) yang dikembangkan untuk mengoreksi kesalahan penulisan tata bahasa pada dokumen Spesifikasi Kebutuhan Perangkat Lunak (SKPL) dengan skenario *bigram* dan *trigram* dimana hasil koreksi pada skenario *trigram* lebih baik dari *bigram*. Skenario *trigram* menghasilkan tingkat *precision* sebesar 83%, *recall* sebesar 85% dan *f-measure* sebesar 86%.

Tabel 2.1 Studi Literatur

Referensi	Judul Penelitian	Tahun	Metodologi	Hasil
(Elvina et al., 2019)	<i>Line Bot Chat Filtering using Naïve</i>	2019	Klasifikasi pesan yang penting dan	Penelitian ini menghasilkan tingkat

	<i>Bayes Algorithm</i>		pesan yang tidak penting dengan algoritma <i>Naïve Bayes</i>	<i>precision</i> sebesar 94.2% dan tingkat <i>recall</i> sebesar 95.6%.
(Hartina & Masri, 2020)	Pendeteksi Kesalahan Pengetikan Kata Non Baku pada Karya Tulis Menggunakan Metode <i>N-Gram</i>	2020	Deteksi kata dan karakter yang tidak sesuai dengan penulisan kata baku dalam Bahasa Indonesia menggunakan metode <i>N-gram</i>	Penelitian ini berhasil mendeteksi 70% dari 10 kata yang dideteksi dan 80% keberhasilan dari pengujian karya ilmiah oleh sistem.
(Enda et al., 2019)	Identifikasi Kesalahan Tata Bahasa Pada Pernyataan Kebutuhan Menggunakan Probabilistik Model Bahasa <i>N-Gram</i>	2019	Mengidentifikasi kesalahan tata bahasa pada SKPL dengan menghitung probabilitas model bahasa <i>N-gram</i>	Penelitian ini menggunakan 2 skenario model <i>n-gram</i> , <i>bigram</i> dan <i>trigram</i> . Hasilnya skenario <i>trigram</i> lebih tinggi dari <i>bigram</i> dengan tingkat <i>precision</i>

				sebesar 83%, <i>recall</i> sebesar 85% dan <i>f-</i> <i>measure</i> sebesar 86%
--	--	--	--	---

## 2.2 Dasar Teori

### 2.2.1. Pesan

Menurut Onong Effendy, pesan merupakan “suatu komponen dalam proses komunikasi berupa paduan dari pikiran dan perasaan seseorang dengan menggunakan lambang, bahasa/lambang-lambang lainnya yang disampaikan kepada orang lain” (Effendy, 1984). Mulai dari kode pesan, isi pesan dan bentuk pesan, jika terdapat tiga bagian tersebut maka pesan dapat dipahami.

- Kode pesan merupakan sederetan simbol yang disusun sedemikian rupa sehingga dapat bermakna bagi yang lain. Misalnya bahasa Indonesia yang menjadi kode dan mencakup unsur bunyi atau suara, hurup dan juga kata lalu disusun sedemikian rupa sehingga memiliki arti.
- Isi Pesan merupakan bahan atau materi yang dipilih oleh komunikator untuk mengkomunikasikan maksud atau tujuannya.
- Wujud pesan adalah sesuatu yang membungkus isi pesan itu sendiri dimana komunikator memberikan wujud nyata supaya komunikan tertarik untuk mengetahui dan memahami isi pesan tersebut. (Siahaan, 1991)

### 2.2.2. Kata Baku & Tidak Baku

Kata baku adalah kata-kata yang lazim digunakan dalam situasi formal atau resmi yang penulisannya sesuai dengan kaidah-kaidah yang dibakukan. Baku tidaknya sebuah kata dapat dilihat dari segi lafal, ejaan, gramatika, dan kenasionalan-nya. Kaidah standar yang dimaksud dapat berupa pedoman ejaan (EYD), tata bahasa baku, dan kamus (Setiawati, 2016). Lain halnya dengan kata baku yang mengacu pada Kamus Besar Bahasa Indonesia (KBBI), kata tidak baku lebih luwes.

### 2.2.3. Python

*Python* adalah bahasa pemrograman tingkat tinggi yang ditafsirkan, berorientasi objek, dengan semantik dinamis. Struktur data bawaan tingkat tinggi, dikombinasikan dengan pengetikan dinamis dan pengikatan dinamis, membuatnya sangat menarik untuk Pengembangan Aplikasi Cepat, serta untuk digunakan sebagai bahasa skrip atau perekat untuk menghubungkan komponen yang ada bersama-sama. Sintaks *python* yang sederhana dan mudah dipelajari menekankan pada keterbacaan dan karena itu mengurangi biaya pemeliharaan program. *Python* mendukung modul dan paket, yang mendorong modularitas program dan penggunaan kembali kode. Penerjemah *Python* dan pustaka standar yang luas tersedia dalam bentuk sumber atau biner tanpa biaya untuk semua *platform* utama, dan dapat didistribusikan secara bebas (Python, 2001).

### 2.2.4. Text Mining

*Text mining* merupakan salah satu teknik yang dapat digunakan untuk melakukan klasifikasi, dimana *text mining* merupakan variasi dari *data mining* yang berusaha menemukan pola yang menarik dari sekumpulan data tekstual yang berjumlah besar (Feldman & Sanger, 2006). Dalam *text mining*, terdapat tahap *text preprocessing* yang harus dilakukan seperti:

- Case Folding  
Pada tahap *case folding* dilakukan penyetaraan bentuk kalimat. Semisal mengkonversi seluruh kalimat menjadi huruf kecil (*lowercase*) atau menjadi huruf besar (*uppercase*) (Packard, 2018).
- Tokenizing  
Pada tahap *tokenizing* dilakukan proses pemilahan dari kalimat menjadi kata (Weiss et al., 2010).
- Filtering  
Pada tahap *filtering* dilakukan pemilihan kata penting dari hasil *tokenizing*. Algoritma yang digunakan bisa *stopword* (membuang kata tidak penting) atau *wordlist* (menyimpan kata penting) (Berry & Kogan, 2010).
- Stemming

Pada tahap *stemming* dilakukan penguraian dan pemetaan bermacam bentuk kata menjadi bentuk dasar dari kata tersebut (Fadillah, 2003). Semisal kata berteman, menemani, akan diurai menjadi kata dasarnya yaitu teman.

#### 2.2.5. Telegram Bot

Telegram menyediakan *bot*, yaitu akun telegram unik yang dirancang untuk dapat merespons perintah teks tertentu secara otomatis dengan memberikan jawaban yang telah diformat sebelumnya. Pengguna dapat berinteraksi dengan *bot* dengan mengirimkan sebuah pesan perintah melalui pesan pribadi atau pesan grup (Nufusula & Susanto, 2018).

*Bot* sendiri dapat dibuat dengan agen *bot* milik telegram yang bernama BotFather. BotFather merupakan *bot* yang mengatur semua peraturan yang ada pada setiap *bot*. *Bot* ini dapat membantu pengguna membuat *bot* baru dan mengubah konfigurasi *bot* yang telah dibuat sebelumnya. Komunikasi antara *bot* dengan server pengembang dapat dilakukan dengan *API* yang disediakan telegram, yaitu *Bot API*. Selain itu supaya *bot* dapat merespon sesuai dengan keinginan pengembang, telegram menyediakan *library bot* di berbagai macam bahasa pemrograman, misal pada *python* dengan *library pyTelegramBotAPI*. Banyak sekali hal-hal yang dapat dilakukan oleh *bot*, seperti mengirim notifikasi yang disesuaikan, membuat alat khusus (seperti peramalan cuaca, *translator*, dll), membuat permainan (*game*), dan hal lainnya (Telegram, 2015a).

#### 2.2.6. Telegram Bot API

*Bot API* adalah antarmuka berbasis *HTTP* yang dibuat bagi pengembang yang tertarik untuk membuat *bot* di telegram. *Bot API* memungkinkan pengembang melakukan komunikasi antara *bot* telegram dengan server pengembang. Setiap *bot* memiliki token-nya sendiri, yang digunakan untuk memberikan akses komunikasi dan kontrol *bot* pada server pengembang. Semua komunikasi yang dilakukan ke *Bot API* harus dikirim melalui *HTTPS* dengan format

[https://api.telegram.org/bot<token>/METHOD\\_NAME](https://api.telegram.org/bot<token>/METHOD_NAME)

Selain via server telegram, membangun server lokal juga menjadi alternatif komunikasi server. Permintaan (*request*) dari *bot* akan dikirim ke server lokal tersebut tanpa harus melalui format *HTTPS* telegram. Penggunaan server lokal memberikan beberapa keuntungan seperti mengunduh file dengan ukuran tak

terbatas (*unlimited*) dan melampirkan file sampai dengan 2000 Mb (Telegram, 2015b).

### 2.2.7. *N-gram*

*N-gram* adalah sekumpulan *n*-item yang disusun secara berskala dari sebuah data teks. Nilai dari *N* sangat beragam sesuai kebutuhan, mulai dari satu hingga sepanjang teks yang ada. Misalnya terdapat sebuah kata "PALSU", maka terdapat banyak ragam dari *N-gram* yang dibuat seperti pada Tabel 2.2.

Tabel 2.2 Contoh *N-gram* dari kata "PALSU"

Kata	N	<i>N-Gram</i> yang dibuat
PALSU	1	(P), (A), (L), (S), (U)
	2	(PA), (AL), (LS), (SU)
	3	(PAL), (ALS), (LSU)

*N-gram* dapat dipakai untuk menyelesaikan berbagai masalah, satu diantaranya digunakan untuk memprediksi kata yang akan dipakai. Contoh fungsi *auto-correct* pada keyboard smartphone atau pencarian kata kunci tertentu pada pencarian Google untuk membantu pengguna (Pratomo & Ijtihadie, 2016).

### 2.2.8. *Confusion Matrix*

*Confusion matrix* merupakan metode yang biasanya digunakan untuk perhitungan akurasi. Saat menguji keakuratan hasil pencarian akan dievaluasi berdasarkan nilai *recall*, *precision*, *accuracy* dan *f-measure* (Amrizal, 2018). Tabel *confusion matrix* dapat dilihat pada Tabel 2.3.

Tabel 2.3 Tabel *Confusion Matrix*

		Prediksi	
		Prediksi Benar	Prediksi Salah
Aktual	Aktual Benar	TP	FN
	Aktual Salah	FP	TN

Keterangan:

TP (*True Positive*) : Jumlah prediksi yang benar bersifat positif

FP (*False Positive*) : Jumlah prediksi yang salah bersifat positif

TN (*True Negative*) : Jumlah prediksi yang benar bersifat negatif

FN (*False Negative*) : Jumlah prediksi yang salah bersifat positif

Maka, dapat dirumuskan seperti berikut:

$$Recall = \frac{TP}{TP + FN} \quad (2.1)$$

$$Precision = \frac{TP}{TP + FP} \quad (2.2)$$

$$F - Measure = \frac{2 \times (Recall \times Precision)}{Recall + Precision} \quad (2.3)$$

$$Accuracy = \frac{TP+FP}{TP+FP+TN+FN} \quad (2.4)$$