

# IMPLEMENTASI NLP DENGAN KONVERSI KATA PADA SISTEM CHATBOT KONSULTASI LAKTASI

Mungki Astiningrum<sup>1</sup>, Pramana Yoga Saputra<sup>2</sup>, Maya Shoburu Rohmah<sup>3</sup>

<sup>1,2,3</sup>Program Studi Teknik Informatika, Jurusan Teknologi Informasi, Politeknik Negeri Malang

<sup>1</sup>mungki\_astiningrum@polinema.ac.id, <sup>2</sup>pramanay@gmail.com, <sup>3</sup>shoburuohmah09@gmail.com

---

## Abstrak

Penyampaian informai melalui layanan *customer service* saat ini berkembang pesat. Salah satu media layanan *customer service* adalah *chatting* yang digunakan untuk konsultasi. Penggunaan *chatting* saat berkonsultasi, memudahkan pertanyaan yang diajukan oleh *customer* beraneka ragam. Namun keterbatasan pemahaman bahasa oleh operator atau dokter menyebabkan kesalahpahaman dalam berkomunikasi. Tidak hanya hal tersebut, tetapi keterbatasan waktu kerja operator atau dokter dalam penyampaian informasi dirasa kurang efektif bagi *customer* yang ingin mendapatkan informasi yang cepat.

Berdasarkan permasalahan tersebut, peneliti bermaksud membangun sebuah aplikasi yang dapat digunakan sebagai media pengganti *chatting* yaitu berupa *chatbot* dengan menggunakan metode *Natural Language Processing*. Dalam *Natural Language Processing* dilakukan proses *Text Preprocessing* yang digunakan untuk menormalisasi input user menggunakan metode *Levenshtein Distance* dan *TFIDF*. Metode *Levenshtein distance* digunakan untuk konversi kata dengan jarak edit kata yang tepat. Selain itu metode *TFIDF* digunakan untuk pembobotan suatu dokumen yang digunakan untuk mendapatkan nilai *cosine similarity*. *Cosine similarity* digunakan untuk mencari jawaban yang sesuai dengan inputan user yaitu dengan membandingkan seluruh dokumen yang tersimpan pada *knowledge base*. Sehingga dengan *Natural Language Processing* pembangunan aplikasi chatbot dapat melakukan konversi kata dan memberikan respon sesuai dengan inputan user secara alami saat berkomunikasi dengan komputer (*chatbot*). Dari hasil pengujian sistem, bahwa kemungkinan sistem memberikan jawaban kepada user didapatkan nilai presisi terbaik sebesar 0.94 pada uji coba dan untuk nilai *recall* terbaik sebesar 0.85 yang didapatkan dari hasil uji coba relevan dari sistem memberikan jawaban.

**Kata kunci:** Natural Language Processing, Chatbot, Text Preprocessing, TFIDF, Levenshtein Distance, Cosine Similarity.

---

## 1. Pendahuluan

Penyampaian informai melalui layanan *customer service* saat ini berkembang pesat. Seperti halnya pada perkembangan layanan *customer service* di media sosial dunia maya. Layanan *customer service* dapat dilakukan dengan menjawab langsung pertanyaan *customer* oleh manusia atau dapat menggunakan mesin untuk menjawab pertanyaan yang diajukan secara otomatis.

Untuk penelitian ini, peneliti melakukan observasi di Pusat Laktasi Malang. Pada tempat observasi tersebut, peneliti mengamati seringnya layanan *customer service* dilakukan dengan menggunakan *chatting* untuk konsultasi. Dengan adanya *chatting* dapat memudahkan *customer* berkonsultasi tanpa harus bertatap muka dengan dokter. Penyampaian informasi melalui layanan *customer service* dengan aplikasi *chatting* umumnya dilakukan oleh dua individu yaitu *customer* dan operator atau dokter.

Namun biasanya pertanyaan yang diajukan oleh *customer* beraneka ragam dan menggunakan bahasa sehari-hari yang kadang tidak dimengerti oleh

operator atau dokter. Penggunaan kosa kata yang tidak baku, bahasa gaul atau kesalahan dalam penulisan kata menyebabkan kesalahpahaman dalam berkomunikasi. Contoh: pada singkatan kata atau “bgmn” ini biasa diucapkan “Bagaimana” dan kata ini merupakan singkatan yang harus dinormalisasikan menjadi kata “Bagaimana”. Keterbatasan pemahaman mengenai kosa kata yang digunakan atau menyebabkan waktu tanya jawab yang lama untuk memahami maksud dari *customer*, dan informasi yang di dapat kurang efektif bahkan tidak akurat. Tidak hanya pemahaman kalimat saja, namun keterbatasan waktu kerja operator atau dokter dalam penyampaian informasi juga dirasa kurang efektif bagi *customer* yang ingin mendapatkan informasi yang cepat tanpa keterbatasan waktu.

Untuk itu diperlukan suatu alat media layanan informasi yang dapat merespon setiap pertanyaan *customer* tanpa ada keterbatasan waktu yaitu dengan menggunakan sistem chatbot. *Chatbot* merupakan program komputer yang digunakan untuk mensimulasikan sebuah percakapan atau komunikasi yang interaktif oleh seseorang untuk membahas sesuatu. Salah satu metode pembuatan *chatbot* adalah

dengan menggunakan metode NLP (*Natural Language Processing*) yang berisi sekumpulan pola (*pattern*) dan *template* yang digunakan chatbot untuk penelusuran jawaban dari kalimat yang masuk.

Chatbot juga dapat memberikan nilai tambah bagi pelayanan konsultasi melalui aplikasi yang interaktif sehingga dapat menjangkau *customer*.

## 2. Landasan Teori

### 2.1 Chatbot

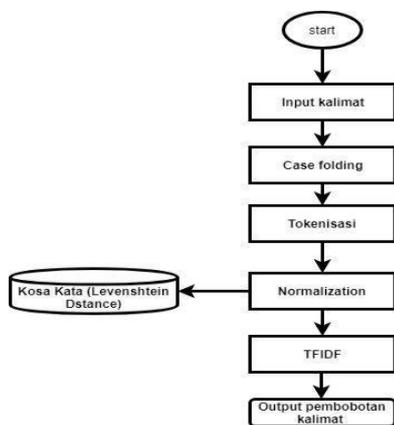
Chatbot merupakan program komputer yang berinteraksi dengan user menggunakan bahasa natural. Teknologi chatbot pertama dimulai pada tahun 1960an. Tujuan pembuatan chatbot ini adalah pengujian apakah chatbot dapat menipu user agar mereka mengira sedang berkomunikasi dengan manusia Indana, *et al* (2018).

### 2.2 Natural Language Processing

Natural Language Processing (NLP) adalah bidang penelitian dan aplikasi yang mengeksplorasi bagaimana komputer dapat digunakan untuk memahami dan memanipulasi teks bahasa alami. Peneliti NLP bertujuan untuk mengumpulkan pengetahuan tentang bagaimana manusia memahami dan menggunakan bahasa sehingga peralatan dan teknik pemasangan dapat dikembangkan untuk membuat sistem komputer memahami dan memanipulasi bahasa alami untuk melakukan tugas yang disukai Indana, *et al* (2018).

### 2.3 Text Preprocessing

Metode *Preprocessing* memainkan peran yang sangat penting dalam teknik dan aplikasi text mining. Ini adalah langkah pertama dalam proses penambangan teks. Dalam makalah ini, kita membahas tiga langkah utama dari *preprocessing* yaitu, menghentikan kata-kata penghapusan, *stemming* dan *TF/IDF algoritma* (Gambar 1).



Gambar 1. Text Preprocessing

Berikut merupakan komponen text preprocessing:

1. Case folding : *folding* yaitu proses ini akan mengubah seluruh teks pada dokumen menjadi bentuk standar yaitu huruf kecil atau *lowercase*. Tokenization adalah langkah yang membagi string teks yang lebih panjang menjadi potongan-potongan yang lebih kecil, atau token.
2. Normalization umumnya mengacu pada serangkaian tugas terkait yang dimaksudkan untuk mengonversi semua teks. Normalization memanfaatkan dataset berupa kosa kata dengan perhitungan levenshtein distance untuk menghitung jarak edit tiap kata.
3. TFIDF : memanfaatkan output dari *normalization* agar mengacu pada *knowledge base similarity* dengan menghitung pembobotan frekuensi dari term yang mengandung pada dokumen.

### 2.4 Levenshtein Distance

Levenshtein Distance adalah salah satu dari beberapa metode untuk menghitung jarak perbedaan antar kata. *Levenshtein Distance* merupakan metode dalam menghitung nilai yang didapat dari hasil operasi modifikasi satu kata dengan kata yang lain dengan bantuan *matrix*.

Cara yang digunakan adalah dengan melihat satu persatu karakter dengan karakter lainnya, apakah untuk menutupi perbedaan tersebut perlu adanya penambahan huruf, penghapusan huruf, atau penyisipan huruf. Persamaan yang digunakan untuk mencari Distance adalah sebagai berikut:

$$\begin{aligned}
 & \cdot (, ) = \{ \\
 & \cdot ((, \quad - 1) + 1) \\
 & \cdot (( \quad - 1, ) + 1) \qquad (1) \\
 & \cdot (( \quad - 1, \quad - 1) + 1( \neq )) \\
 & i : \text{iterasi kata pertama} \\
 & j : \text{iterasi kata kedua} \\
 & \text{Dist} : \text{jarak}
 \end{aligned}$$

### 2.5 TFIDF

Metode *Term Frequency-Inverse Document Frequency (TF-IDF)* adalah cara pemberian bobot hubungan suatu kata (term) terhadap dokumen.

- a. Perhitungan Term Frequency (tf) menggunakan persamaan (2.1)
 
$$= ij \qquad (2)$$

Dengan tf adalah *term frequency*, dan adalah banyaknya kemunculan term dalam dokumen, *Term*

frequency (*tf*) dihitung dengan menghitung banyaknya kemunculan term dalam dokumen .

b. Perhitungan *Inverse Document Frequency (idf)*, menggunakan persamaan (2.2)

$$= \frac{1}{N} \quad (3)$$

Dengan adalah inverse document frequency, N adalah jumlah dokumen yang terambil oleh sistem, dan adalah banyaknya dokumen dalam koleksi dimana term muncul di dalamnya. Rizqi *et al* (2016)

c. Perhitungan *term frequency Inverse Document Frequency (tfidf)*, menggunakan persamaan (2.3)

$$= , \quad (4)$$

Dengan adalah bobot dokumen, N adalah Jumlah dokumen yang terambil oleh sistem, adalah banyaknya kemunculan *term* pada dokumen , dan adalah banyaknya dokumen dalam koleksi dimana *term* muncul di dalamnya.

Bobot dokumen ( ) dihitung untuk didapaknya suatu bobot hasil perkalian atau kombinasi antara *term frequency* ( ) dan *Inverse Document Frequency* ( ) Indana, *et al* (2018).

### 2.6 Cosine Similarity

$$= \frac{|v \cdot w|}{||v|| \cdot ||w||} = \frac{\sum_{i=1}^n v_i w_i}{\sqrt{\sum_{i=1}^n v_i^2} \cdot \sqrt{\sum_{i=1}^n w_i^2}} \quad (5)$$

*Cosine similarity* merupakan rumus yang digunakan

Untuk menghitung kesamaan atau *similarity* dengan menentukan sudut antara vektor dokumen dengan vektor *query* dalam dimensi *V* pada bidang *Euclidean*. Hasil dari *cosine similarity* memiliki nilai antara 0 sampai dengan 1. Nilai 0 merupakan nilai yang didapat apabila dokumen tidak berhubungan dengan *query*, sedangkan nilai 1 berarti dokumen memiliki keterhubungan tinggi dengan *query* (Lahitani, Permanasari dan Setiawan, 2016). Indana, *et al* (2018).

### 2.7 Presisi dan Recall

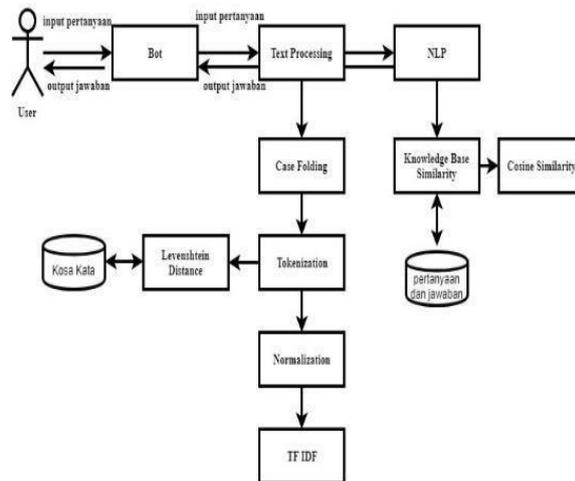
Recall menghitung jumlah informasi relevan yang diekstraksi pada sistem. Sedangkan *presisi*, menghitung jumlah informasi bernilai benar/akurat yang dikembalikan oleh sistem (Mishra dan Vishwakarma, 2016). Formula untuk perhitungan *presisi* dan *recall* dijabarkan pada Persamaan 6 dan Persamaan 7: Fahana, *et al* (2017).

$$= \frac{\text{Jumlah jawaban relevan sistem}}{\text{Total jawaban relevan dalam teks}} \quad (6)$$

$$= \frac{\text{Jumlah jawaban relevan sistem}}{\text{Total jawaban relevan dalam sistem}} \quad (7)$$

Pemberian nama judul-judul bab, kecuali bab Pendahuluan dan bab Kesimpulan dan Saran, sebaiknya secara eksplisit menyatakan isinya. Tidak perlu implisit dinyatakan sebagai Dasar Teori, Perancangan, dan sebagainya.

### 3. Perancangan



Gambar 2. Gambar Deskripsi Sistem

Berikut merupakan langkah-langkah pengolahan data menggunakan *Natural Language Processing*:

**Langkah 1:** User memasukkan kalimat input pada aplikasi *chatbot* konsultasi laktasi. Misalnya dalam implementasi user memasukkan kalimat “bayi yang mengalami bigung puting”

**Langkah 2:** Bot akan meneruskan pertanyaan tersebut pada proses *text preprocessing*. Kalimat input user dilakukan proses *case folding* yaitu proses ini akan mengubah seluruh teks pada dokumen menjadi bentuk standar yaitu huruf kecil atau *lowercase*.

**Langkah 3:** Setelah dilakukan *case folding*, dilakukan proses *tokenization* yaitu proses menganalisa input secara sintaksis. *Tokenization* adalah proses memecah aliran teks menjadi kata-kata, frasa, simbol, atau elemen bermakna lainnya yang disebut token. Tujuan dari *tokenization* adalah eksplorasi kata-kata dalam sebuah kalimat. Kannan & Gurusamy (2015) Pada proses ini, dilakukan

pemotongan string *input* berdasarkan tiap kata penyusun dari dokumen Indana, *et al* (2018).

**Langkah 4:** Selanjutnya proses *normalization* yaitu menempatkan semua kata dengan pijakan yang sama, dan memungkinkan pemrosesan untuk diproses secara seragam. Setelah salah satu kata diidentifikasi, proses *normalization* akan membandingkan setiap input kata dengan semua kata yang disimpan dalam database. Dalam membandingkan setiap input kata dengan kata yang disimpan proses ini merupakan untuk melakukan konversi kata tidak baku menjadi kata baku. Sehingga setiap kata menjadi data masukan (kata sumber) untuk proses koreksi kata dengan mencari kata pada kamus (*lexicon*) yang memiliki jarak edit minimum terhadap kata target. Kemudian ambil nilai minimal dengan rumus sebagai berikut

- $d[i-1,j] + 1 \implies d[1-1,2] + 1 = 3$
- $d[i,j-1] + 1 \implies d[1,2-1] + 1 = 1$
- $d[i-1,j-1] + \text{cost} \implies d[1-1,2-1] + 1 = 2$

Tabel 1. Contoh Perhitungan *Leveinshtein Distance* Kata “bingung”

Tabel 1. Perhitungan

		<b>B</b>	<b>I</b>	<b>G</b>	<b>U</b>	<b>N</b>	<b>G</b>
	0	1	2	3	4	6	7
<b>B</b>	1	0	1	2	3	4	5
<b>I</b>	2	1	0	1	2	3	4
<b>N</b>	3	2	1	1	2	2	3
<b>G</b>	4	3	2	1	2	3	2
<b>U</b>	5	4	3	2	1	2	3
<b>N</b>	6	5	4	3	2	1	2
<b>G</b>	7	6	5	4	3	2	<b>1</b>

Kemudian kita bandingkan dengan kata pembanding lain yang memiliki nilai terkecil *levenshtein distance*. Lakukan langkah tersebut sampai akhirnya mendapatkan hasil. Perhitungan nilai jarak Levenshtein digambarkan melalui tabel matriks 3.1

. Nilai terakhir pada tabel “1” adalah nilai jarak terkecil yang diperlukan untuk melakukan operasi *Levenshtein*.

Kemudian lakukan proses sorting untuk mengetahui hasil nilai terkecil dari seluruh kata yang telah di hitung nilai jarak edit. Nilai terkecil dari proses sorting digunakan untuk keluaran kata *normalization*.

**Langkah 5 :** Setelah dilakukan *normalization*, maka output yang dihasilkan adalah analisa sintak yaitu pembenaran kata dari proses *normalization* yang akan digunakan untuk mencari jawaban yang sesuai dengan inputan *user*. Kemudian dilakukan proses *TFIDF*. Disini memanfaatkan output dari

*normalization* agar mengacu pada *knowledge base similarity* Dr.S.Vijayarani, *et al*.

$$= t * \tag{8}$$

**Langkah 6 :** Setelah mendapatkan *Tfidf* kemudian proses *Knowledge Base Similarity* yaitu kesamaan berbasis pengetahuan adalah salah satu kesamaan semantic langkah-langkah yang didasarkan pada identifikasi tingkat kesamaan antara kata-kata dihitung menggunakan rumus cosine-similarity. Kesamaan kosinus adalah ukuran kesamaan antara kedua vektor ruang produk yang mengukur kosinus sudut di antara kedua vektor.

[10]

$$= \frac{||*||}{\sqrt{\sum=() + \sum=()}} \tag{9}$$

Nilai *cosine similarity* dokumen terbesar lah yang akan ditampilkan sebagai respon atau output dari proses *nlp* yang digunakan untuk pertanyaan *user* atau pasien. Sehingga didapatkan hasil akhir dari proses *nlp* dengan pertanyaan dan jawaban seperti tabel 2.

Tabel 2. Contoh Pertanyaan dan Jawaban pada

Pertanyaan	Jawaban
bayi yang mengalami bingung puting	Ibu harus tetap dekat dan sigap merawat apabila bayi dalam masalah menyusu. Jangan beralih dahulu ke penggunaan dot. Ibu dapat menggunakan media lain seperti Pakai sendok, pipet, <i>cupfeeder</i> , atau <i>softcupfeeder</i> .

Setelah proses metode *Natural Language Processing* dilakukan, maka akan keluar hasil dari input *chatting* tersebut dalam bentuk paling sederhana. Kemudian hasil tersebut nantinya akan digunakan untuk membantu *user* menerjemahkan inputan user dalam *chatting* sehingga membantu *customer* dalam menyampaikan keluhan atau mendapatkan informasi mengenai seputar layanan produk atau jasa dalam layanan konsultasi laktasi.

#### 4. Implementasi

##### 4.1 Implementasi Basis Data

Implementasi dari perancangan basis data sesuai perancangan yang telah dilakukan sebelumnya pada bab sebelumnya.

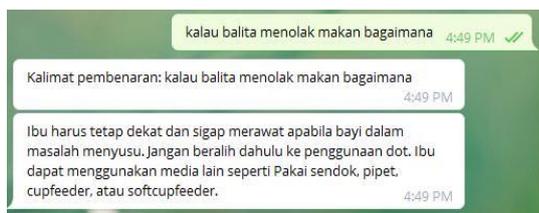


Gambar 3. Implementasi Basis Data

1. Database kosa kata berisikan seluruh kosa kata yang terdapat pada knowledge base dengan memodifikasi tambahan kata yang sering diajukan dalam pertanyaan seputar konsultasi laktasi
2. Database pertanyaan atau knowledge base adalah sekumpulan pertanyaan dan jawaban

##### 4.2 Implementasi Program

Implementasi sistem ini dilakukan dengan pemrograman Python. Setiap menggunakan sistem ini kita bisa mencoba mengetahui bagaimana proses Natural Language Processing itu bekerja maka dari itu di halaman ini kita bisa mencoba sebuah pertanyaan lalu pertanyaan tersebut akan dijawab oleh sistem menggunakan metode Natural Language Processing.



Gambar 4. Halaman Simulasi Tanya Jawab

#### 5. Pengujian dan Pembahasan

##### 5.1 Pengujian

###### 5.1.1 Pengujian Metode

Setelah pengujian fungsional selesai dilakukan, maka langkah selanjutnya adalah pengujian metode. Pengujian metode dilakukan dengan membuat skenario tes. Skenario yang digunakan berdasarkan metode Natural Language Processing. Berikut adalah skenario yang dilakukan:

- a. Bertanya menggunakan pertanyaan yang berasal dari jawaban di knowledge base Dari keseluruhan percobaan sistem menjawab semuanya dengan

benar 100%. Jika digambarkan dalam rumus maka:

Jumlah total pertanyaan yang ditanyakan = 16  
 Jumlah total pertanyaan yang dijawab benar = 16

$$\text{Akurasi} = \left( \frac{16}{16} \right) \times 100\%$$

$$\text{Akurasi} = 100\%$$

- b. Bertanya menggunakan pertanyaan sembarang Berdasarkan hasil pengujian maka diperoleh data jawaban rekomendasi sistem untuk pertanyaan yang berasal dari pertanyaan customer dan dibandingkan dengan pertanyaan yang ada di Knowledge Base. Untuk tingkat keakurasian sistem, maka dilakukan perhitungan persentase keakuratan dari persamaan berikut:

Maka dari itu bot menjawab 51 menjawab pertanyaan dengan benar dan 60 pertanyaan kurang tepat. Dapat disimpulkan bahwa:

Jumlah total pertanyaan yang ditanyakan = 60  
 Jumlah total pertanyaan yang dijawab benar = 51

$$\text{Akurasi} = \left( \frac{51}{60} \right) \times 100\%$$

$$\text{Akurasi} = 85\%$$

##### 5.1.2 Pengujian Pengguna

- a. Pengujian pengguna menggunakan pengujian kuisisioner. Kuisisioner ini diisi oleh 29 responden setelah menguji coba sistem aplikasi, dengan memberi pertanyaan mengenai layanan konsultasi laktasi.

Table 3. Pengujian pengguna

Rata-rata	Sangat Setuju	Setuju	Cukup	Tidak Setuju
	28.55%	48.5%	15.1%	7.7%

$$\text{Recall} = \frac{51}{51+9} = \frac{51}{60} \times 100\% = 0.85 \text{ atau } 85\%$$

- b. Pengujian *Presisi dan Recall*

Pengujian *Presisi dan Recall* digunakan untuk mengetahui relevan dari jawaban yang diberikan oleh sistem dengan jumlah data 60 pertanyaan.

$$\text{Presisi} = \frac{51}{51+3} = \frac{51}{54} \times 100\% = 0.94 \text{ atau } 94\%$$

#### 5.2 Pembahasan

##### 5.2.1 Pembahasan Pengujian Metode

Pembahasan Bertanya Menggunakan Pertanyaan Yang Berasal Dari Jawaban di Knowledge Base. Berdasarkan hasil pengujian maka diperoleh data

jawaban rekomendasi sistem untuk pertanyaan yang berasal dari Knowledge Base dan dibandingkan dengan realisasi jawaban yang ada di Knowledge Base itu sendiri. Untuk tingkat keakurasian sistem, maka dilakukan perhitungan persentase keakuratan dari persamaan berikut:

Jumlah total pertanyaan yang ditanyakan = 16  
 Jumlah total pertanyaan yang dijawab benar = 16

$$\frac{16}{16} \times 100\% = 100\%$$

Hasil pengujian menunjukkan bahwa presentase kecocokan sistem dengan data actual adalah rata-rata 100 %.

- a. Bertanya menggunakan pertanyaan sembarang Berdasarkan hasil pengujian maka diperoleh data jawaban rekomendasi sistem untuk pertanyaan yang berasal dari pertanyaan customer dan dibandingkan dengan pertanyaan yang ada di Knowledge Base. Untuk tingkat keakurasian sistem, maka dilakukan perhitungan persentase keakuratan dari persamaan berikut:

Jumlah total pertanyaan yang ditanyakan = 60  
 Jumlah total pertanyaan yang dijawab benar = 51

$$\frac{51}{60} \times 100\% = 85\%$$

Hasil pengujian menunjukkan bahwa presentase kecocokan sistem dengan data aktual adalah rata-rata 85 %.

### 5.2.2 Pembahasan Pengujian Pengguna

- a. Pengujian Kuisisioner

Tabel 4. Pengujian Kuisisioner

Rata-rata	Sangat Setuju	Setuju	Cukup	Tidak Setuju
	28.55%	48.5%	15.1%	7.7%

- b. Pengujian sistem menggunakan presentase dari presisi dan recall. Penggunaan *presisi* dan *recall* digunakan untuk perhitungan apakah sistem mampu memberikan informasi relevan dalam pencarian dokumen untuk jawaban.
- c. Hasil dari presisi yaitu sebesar :  $0,94 = 0,94 \times 100\% = 94\%$  dan recall yaitu sebesar :  $0,85 = 0,85 \times 100\% = 85\%$ .

## 6. Kesimpulan dan Saran

### 6.1 Kesimpulan

Berdasarkan penelitian yang telah dilakukan oleh penulis disimpulkan bahwa :

1. Dengan menggunakan metode *Levenshtein Distance* dapat menormalisasi kata yang tidak baku menjadi kata baku sehingga apabila ada kalimat pertanyaan yang tidak baku bisa ternormalisasi menjadi kalimat baku. Dengan kalimat baku tersebut, maka sistem chatbot laktasi dapat membalas pertanyaan user dengan benar
2. Dengan menerapkan metode *TFIDF* dan *Cosine Similarity* sistem bisa mencari jawaban yang sesuai dengan pertanyaan user. Hal ini dibuktikan dengan pengujian 60 data pertanyaan. Pada pengujian tersebut dilakukan perhitungan presisi dan recall. Hasil dari perhitungan precision adalah 0.94. Hal ini menunjukkan bahwa kemungkinan sistem memberikan jawaban kepada user adalah 0.94 atau 94%. Sedangkan hasil dari perhitungan recall adalah 0.85. Hal ini menunjukkan dari jawaban yang diberikan oleh sistem kepada user yang relevan adalah 0.85 atau 85%.
3. Dengan diterapkannya metode *Levenshtein Distance* dan *TFIDF* dan *Cosine Similarity* pada aplikasi chatbot membuat aplikasi chatbot ini layak digunakan oleh *customer service* pusat laktasi. Hal ini ditunjukkan dari hasil *User Acceptance Test* yakni diperoleh Sangat Setuju 28.55%, Setuju 48.5%, Cukup 15.1%, dan Tidak Setuju 7.7%

### 6.2 Saran

Saran yang diberikan untuk pengembangan sistem ini ke depannya adalah sebagai berikut:

1. Penelitian ini masih dapat dikembangkan menjadi sistem yang lebih kompleks, tidak hanya terfokus pada menjawab pertanyaan dalam bentuk text saja namun bisa dengan gambar atau pesan suara.
2. Perbaikan pada membenaran kosa kata ditambahkan untuk proses staging dimana proses tersebut menggunakan SPOK untuk melakukan konversi kata dan menambahkan metode lain yang lebih efisien.
3. Untuk pengembangan selanjutnya, dapat dikembangkan menggunakan media sosial lain seperti Instagram, twitter atau facebook.
4. Optimasi sistem lebih ditingkatkan ketika memproses perbaikan kata dan pencarian jawaban.

**Daftar Pustaka :**

- Dr.S.Kannan & Gurusamy, Vairaprakash, March 2015. *“Preprocessing Techniques for Text Mining. Department of Computer Applications, Department of Computer Applications, Madurai Kamaraj University”*. Query: Journal of Information ..., 2017
- Dr. S. Vijayarani, Ms. J. Ilamathi, Ms. Nithya, *Preprocessing Techniques for Text Mining - An Overview*. Department of Computer Science, School of Computer Science and Engineering. Dr.S.Vijayarani et al.
- Indana, Fahma, Arina., Cholissodin, Imam, Setya, Perdana, Rizal., 2018. *Identifikasi Kesalahan Penulisan Kata (Typographical Error) pada Dokumentasi Berbahasa Indonesia Menggunakan Metode N-gram dan Levenshtein Distance*, e-ISSN : 2548-964X
- Fahana, Jefree., Umar, Rusydi., Ridho, Faizin., 2017 *“Pemanfaatan Telegram Sebagai Notifikasi Serangan untuk Keperluan Forensik Jaringan”*, ISSN 2579-5341.
- Langgeng, Adiasto, Aloysius., Witanti, Wina, Yuniarti, Rezki, 2016. *“Sistem Koreksi Kesalahan Pengetikan Menggunakan Levenshtein Distance Pada Layout Qwerty “.. ISSN: 2503-2844.*
- Rizqi, Alfirna L, Erna, Adhistya P, Akhmad, Noor S, 2016. *“Cosine Similarity to Determine Similarity Measure: Study Case in Online Essay Assessment”* DOI: [10.1109/CITSM.2016.7577578](https://doi.org/10.1109/CITSM.2016.7577578)
- Benedictus, Ruspandi R., Wowor, Hans., Sambul, Alwin., 2017. *Rancang Bangun Chatbot Helpdesk untuk Sistem Informasi Terpadu Universitas Sam Ratulangi*. . ISSN : 2301-8364.
- Cholissodin, Tusty Imam., Widodo, Agus Wahyu., 2017 *“Deteksi Kesalahan Ejaan dan Penentuan Rekomendasi Koreksi Kata yang Tepat Pada Dokumen Jurnal JTIK Menggunakan Dictionary Lookup dan Damerau-Levenshtein Distance”* ISSN : 2548-964X.
- Utomo, Dias, Sholeh, Muchammad, Arry Avorizano, 2017. *“Membangun Sistem Mobile Monitoring Keamanan Web Aplikasi Menggunakan Suricata dan Bot Telegram Channel”*.
- Wael H. Gomaa , Aly A. Fahmy, April 2013. *“A Survey of Text Similarity “Approache.* <https://pdfs.semanticscholar.org/5b5c/a878c534aee3882> <https://pdfs.semanticscholar.org/>