

Rancang Bangun *Chatbot* Pengaduan Kekerasan Perempuan Anak Dengan Metode *Fuzzy String Matching* Dan *Enhanced Confix Stripping Stemmer*

Al Muhtadi Billah*, Diah Ayu Retnani Wulandari**, Yudha Alif Auliya***

Program Studi Sistem Informasi, Fakultas Ilmu Komputer, Universitas Jember

*172410101140@mail.unej.ac.id, **diah.retnaniw@unej.ac.id, ***yudha.alif@unej.ac.id

ABSTRACT

Violence against women is a serious issue affecting many individuals in society. In Indonesia, including Kota Probolinggo, the number of cases of violence against women remains high. This research aims to design a chatbot application as a means for reporting violence against women and children (VAWC) in Kota Probolinggo. The application provides an anonymous and safe space for victims to report their complaints without revealing their identities. Additionally, the chatbot delivers information and education about different types of violence against women and preventive measures. The development of the chatbot application utilizes Fuzzy String Matching, Enhanced Confix Stripping Stemmer, and preprocessing methods to enhance data search accuracy and provide appropriate responses to users. The testing results show that the chatbot offers responses with satisfactory accuracy. Leveraging AI technology in the form of a chatbot is expected to provide an effective solution in addressing and preventing violence against women, offering a secure and anonymous platform for victims to report incidents. The application is envisioned to offer better support and assistance to victims of violence. The chatbot application implements fuzzy string matching, the Levenshtein distance algorithm, and Enhanced Confix Stripping Stemmer to process user input before matching it with the database. During the Enhanced Confix Stripping Stemmer process, sentences are transformed into an array of basic words, enhancing the accuracy of data matching when using fuzzy string matching and the Levenshtein distance algorithm. Testing results indicate a 4.77% difference between manual calculations and the chatbot system. This difference may be attributed to variations in text processing and other factors.

Keyword: violence against women, chatbot, fuzzy string matching, enhanced confix stripping stemmer

1. Pendahuluan

Kekerasan terhadap perempuan adalah tindakan atau perlakuan berdasarkan perbedaan jenis kelamin, termasuk ancaman, pemaksaan, atau pengekangan kebebasan perempuan di lingkup publik maupun kehidupan pribadi. Kekerasan ini dapat menyebabkan penderitaan fisik, seksual, dan psikologis bagi perempuan. Di Indonesia, menurut CATAHU (Catatan Tahunan) Komnas Perempuan tahun 2023, pada tahun 2022 terdapat 457.895 kasus kekerasan terhadap perempuan, mengalami penurunan dari 459.094 kasus pada tahun sebelumnya.

Di Kota Probolinggo, pengaduan kekerasan terhadap perempuan dan anak telah digabungkan dengan Dinas Sosial menjadi Dinas Sosial & Pemberdayaan Perempuan dan Perlindungan Anak (Dinsos PPPA). Dinas ini memberikan dukungan kepada korban melalui pendekatan kekeluargaan dan jalur hukum. Namun, seringkali laporan kekerasan berasal dari pihak lain karena korban merasa takut atau khawatir akan disalahkan oleh orang di sekitarnya. Beberapa alasan lainnya termasuk ketergantungan ekonomi pada pasangan dan kurangnya pengetahuan tentang kekerasan dalam rumah tangga (KDRT).

Chatbot adalah program berbasis kecerdasan buatan yang berfungsi sebagai percakapan virtual dengan lawan bicara berupa robot chat. Penggunaan *chatbot* tidak hanya memberikan informasi kepada perempuan yang mengalami kekerasan, tetapi juga menjadi wadah untuk mengungkapkan keluhan kesah yang sulit disampaikan kepada orang lain. *Chatbot* ini dapat menjawab pertanyaan dan merespons pernyataan pengguna. Tujuan dari aplikasi *chatbot* ini adalah untuk membantu korban kekerasan terhadap perempuan dan anak di Kota Probolinggo. *Chatbot* berbasis kecerdasan buatan memberikan sarana aman dan anonim bagi korban untuk menyampaikan keluhan mereka tanpa harus mengungkapkan identitas. Selain itu, *chatbot* juga memberikan informasi dan edukasi tentang jenis-jenis kekerasan dan cara pencegahannya.

Aplikasi *chatbot* ini menggunakan metode *fuzzy string matching* untuk mencari data yang sesuai dengan input pengguna, dengan toleransi terhadap kesalahan pengetikan. *fuzzy string matching* menggunakan fungsi persamaan (*similarity function*) untuk menentukan kesamaan (Widianto 2022) dan memungkinkan aplikasi *chatbot* untuk mencocokkan kata-kata yang tidak persis sama dengan kata-kata dalam kamus atau *database*. Metode ini meningkatkan akurasi pencarian data, mengatasi potensi kesalahan pengguna, dan dapat menangani

sinonim dan variasi kata. Penggunaan metode ini meningkatkan efisiensi pengolahan data, memberikan respons instan, dan menciptakan pengalaman pengguna yang lebih baik. Sebagai pendukung dalam proses pencocokan data, diterapkan pula *enhanced confix stripping stemmer*, sebuah algoritma *stemming* yang berguna untuk mengubah kata-kata menjadi kata dasar dan merupakan algoritma perbaikan dari *Confix Stripping* (CS) yang digunakan untuk *stemming* (pencarian bentuk dasar dari suatu kata) Bahasa Indonesia (Fadziah, Rasim and Fitrajaya 2018). Dengan menggunakan algoritma ini, aplikasi *chatbot* dapat memproses kata-kata yang berbeda bentuk dan mendapatkan hasil pencarian yang lebih akurat dan sesuai dengan tujuan aplikasi.

Dengan menggabungkan metode *fuzzy string matching*, *enhanced confix stripping stemmer*, dan *preprocessing* dalam pengembangan aplikasi *chatbot*, diharapkan aplikasi tersebut dapat memberikan respons yang lebih akurat dan relevan bagi pengguna, serta meningkatkan efisiensi dalam mencari dan menyajikan informasi tentang kekerasan terhadap perempuan dan anak. Penggunaan teknologi AI dalam bentuk *chatbot* ini diharapkan dapat memberikan solusi yang efektif dalam penanganan dan pencegahan kekerasan terhadap perempuan, serta memberikan sarana yang aman dan anonim bagi para korban untuk menyampaikan pengaduan mereka.

2. Metode Penelitian

2.1 Fuzzy String Matching

Metode *fuzzy string matching* mencari kesamaan atau kemiripan tekstual antara *string* masukan dan *string* di kamus atau *database*. *Chatbot* menggunakan aturan pola untuk menyimpan pertanyaan dan jawaban dalam *database*, memilih jawaban yang sesuai dengan pertanyaan sebelumnya (Prianto, Andarsyah, & Harani, 2022). Metode ini mencari *string* dalam kamus data atau *database* meskipun susunan karakternya berbeda. Pencarian *string* adalah proses umum dalam komputasi karena teks adalah bentuk utama penyimpanan data (Sanjung & Norhikmah, 2021). Tujuan utama metode ini adalah memutuskan kesamaan antara *string* yang dicari dan *string* di kamus data atau *database*, meskipun tidak persis sama dalam susunan karakternya (Rohman, Aqharabah, & Solekan, 2023).

2.2 Algoritma Levenshtein Distance

Algoritma *levenshtein distance* atau disebut juga Algoritma edit *distance* adalah sebuah matriks *string* yang digunakan untuk mengukur perbedaan atau jarak (*distance*) antara dua *string*. Perhitungan edit *distance* didapat dari matriks yang digunakan untuk menghitung selisih antara dua *string* (Adawiyah & Saragih, 2022). Algoritma ini menghitung jumlah operasi penghapusan (*deletions*), penyisipan (*insertions*), dan penukaran (*substitutions*) yang dilakukan untuk menyamakan sebuah *string* dengan *string* pembandingan.

2.3 Algoritma Enhanced Confix Stripping Stemmer

Enhanced Confix Stripping (ECS) *Stemmer* merupakan algoritma perbaikan dari *Confix Stripping* (CS) yang digunakan untuk *stemming* (pencarian bentuk dasar dari suatu kata) Bahasa Indonesia (Fadziah & R, 2018). Algoritma ECS *Stemmer* adalah algoritma *stemming* kata Bahasa Indonesia yang memiliki performa terbaik dan jumlah kesalahan yang paling sedikit. Algoritma ini melakukan perbaikan aturan dalam tabel pemenggalan imbuhan dan mampu mengembalikan akhiran kata jika diperlukan.

2.4 Cara Kerja Chatbot

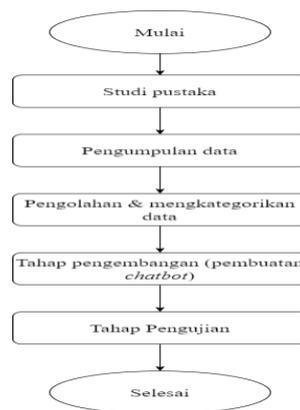
Cara kerja *chatbot* dengan menggunakan *fuzzy string matching*, algoritma *levenshtein distance*, dan algoritma *enhanced confix stripping stemmer* adalah sebagai berikut:

1. Ketika pengguna memberikan input ke *chatbot*, *fuzzy string matching* digunakan untuk mencari kata atau frasa yang mirip dengan kata atau frasa dalam *database chatbot*, bahkan jika terdapat kesalahan pengejaan atau perbedaan karakter.
2. Algoritma *levenshtein distance* dapat digunakan untuk menghitung tingkat kesamaan antara input pengguna dan kata/frasa dalam *database chatbot*. Kata atau frasa dengan *levenshtein distance* terkecil dianggap sebagai yang paling cocok.
3. Algoritma *enhanced confix stripping stemmer* digunakan untuk mengubah kata-kata dalam input pengguna maupun dalam *database chatbot* menjadi bentuk dasar sehingga *chatbot* dapat lebih mudah mencocokkan kata-kata yang berhubungan.

2.5 Jenis Penelitian

Jenis penelitian yang digunakan dalam penelitian ini merupakan penelitian kualitatif untuk memperoleh pemahaman mendalam, mengembangkan teori, menggali informasi spesifik dari narasumber. Metode kualitatif ini memungkinkan untuk menjelajahi berbagai perspektif dan kompleksitas isu kekerasan terhadap perempuan di Indonesia, sehingga mendukung pengembangan aplikasi *chatbot* yang lebih berdaya guna dan relevan.

Penelitian ini dilakukan dalam beberapa tahapan dari awal hingga akhir penelitian. Berikut beberapa tahapan pada penelitian ini yang dimana dapat dilihat pada Gambar 1.



Gambar 1. Tahapan Penelitian

2.6 Studi Pustaka

Studi pustaka merupakan tahapan pertama dimana peneliti menganalisis permasalahan yang terjadi pada kantor Pemberdayaan Perempuan & Perlindungan Anak Kota Probolinggo. Teknik dalam menganalisis kebutuhan yang dilakukan oleh peneliti yakni studi pustaka untuk menghimpun informasi yang relevan dan sesuai dengan topik permasalahan. Sumber informasi dapat diperoleh dari buku-buku, karya ilmiah, ensiklopedia, internet, dan sumber-sumber lainnya. Adapun tujuan diadakannya studi pustaka yakni untuk memanfaatkan semua informasi yang relevan untuk penelitian.

2.7 Pengumpulan Data

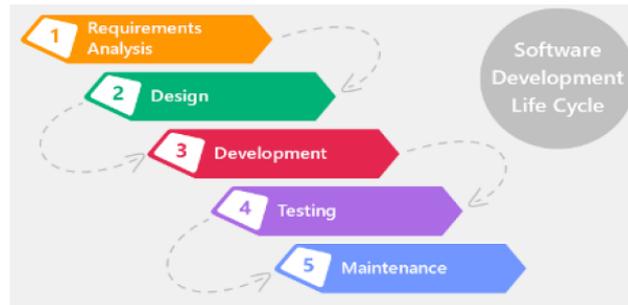
Pada tahapan pengumpulan data, peneliti melakukan wawancara terhadap petugas kantor Pemberdayaan Perempuan & Perlindungan Anak Kota Probolinggo mengenai bagaimana pelaporan kekerasan terhadap perempuan dilakukan. Data yang diperoleh pada tahapan ini berupa informasi mengenai pemahaman kekerasan terhadap perempuan menurut undang-undang yang berlaku, informasi seputar kekerasan terhadap perempuan dan cara yang tepat untuk penanganannya. Terdapat 17 data pertanyaan yang diajukan kepada korban dalam rentang tahun 2020 hingga 2021. Minimnya informasi yang didapatkan setelah melakukan wawancara selain merupakan privasi korban juga diakibatkan karena kurangnya kesadaran perempuan yang mengalami kekerasan untuk melaporkan kekerasan yang dialaminya dikarenakan mereka cenderung merasa takut dan malu apabila privasinya diketahui. Pada saat awal korban bercerita mengenai kekerasan yang dialaminya, petugas kantor PPPA hanya melakukan pendampingan, yang artinya korban dan petugas hanya melakukan percakapan tanpa adanya pencatatan dengan alasan pendekatan terhadap korban, setelah korban bersedia melaporkan kekerasan yang dialaminya, maka akan dilakukan proses Berita Acara Pemeriksaan (BAP) yang dimana proses ini akan dicatat. Pada kesimpulannya, sebelum korban bersedia melaporkan kejadian yang dialaminya, maka petugas kantor tidak akan mencatat percakapan antara korban dan petugas.

2.8 Pengolahan & Mengkategorikan Data

Pada tahap ini, dilakukan pengolahan data kebutuhan dari hasil wawancara untuk narasumber terkait. Data domain dari laporan korban dikategorikan menjadi 9 domain yang berbeda, seperti "Bantuan Dan Dukungan," "Melaporkan Kekerasan," "Penjelasan Undang-Undang," "Edukasi Dan Pencegahan", "Informasi Layanan", "Penyuluhan dan Pelatihan", "Dampak Psikologis dan Pemulihan", "Pengaduan Online", dan "Peran Saksi dan Masyarakat". Kategorisasi ini bertujuan untuk menyortir pertanyaan-pertanyaan pengguna sehingga chatbot dapat memberikan jawaban yang sesuai dengan domain pertanyaan tersebut.

2.9 Tahap Pengembangan

Merupakan tahapan pengembangan sistem setelah pengkategorian data yang telah dikumpulkan sebelumnya. Pengembangan sistem dilakukan untuk menerapkan perhitungan algoritma *enhanced confix stripping* dan algoritma *fuzzy string matching*, dan menghasilkan sistem yang dapat mencocokkan respon yang tepat dari inputan user. Pengembangan sistem pada penelitian ini menggunakan metode *waterfall* yang menggambarkan pendekatan sistematis dan berurutan pada pengembangan perangkat lunak, yang dimulai dengan spesifikasi kebutuhan pengguna (*communication*) dan berlanjut melalui tahapan-tahapan perencanaan (*planning*), pemodelan, (*modeling*), konstruksi (*construction*), serta penyerahan sistem/perangkat lunak ke pengguna (*deployment*), yang diakhiri dengan dukungan berkelanjutan pada sistem (Supiyandi, Zen, Rizal, & Eka, 2022). Metode *waterfall* yang digunakan pada penelitian ini dapat dilihat pada gambar 2 dibawah ini.



Gambar 2. Metode Waterfall (Supiyandi, et al. 2022)

1. *Requirements Analysis*: Identifikasi kebutuhan aplikasi *chatbot* untuk pengaduan kekerasan terhadap perempuan dan anak di Kota Probolinggo. Komunikasi intensif dengan Dinsos PPPA Kota Probolinggo untuk memahami persyaratan dan harapan mereka terhadap aplikasi *chatbot*.
2. *Desain Sistem*: Rancang keseluruhan sistem aplikasi *chatbot*, termasuk antarmuka pengguna, arsitektur, dan algoritma. Integrasi metode *Fuzzy String Matching* dan *Enhanced Confix Stripping Stemmer* untuk meningkatkan kemampuan pencarian dan analisis keluhan pengaduan.
3. *Development*: Bangun aplikasi *chatbot* berdasarkan desain sebelumnya dengan menggunakan bahasa pemrograman yang sesuai dan mengikuti metode *Fuzzy String Matching* dan *Enhanced Confix Stripping Stemmer*.
4. *Pengujian*: Lakukan pengujian menyeluruh untuk memastikan aplikasi *chatbot* sesuai dengan persyaratan. Identifikasi dan perbaiki masalah yang mungkin muncul selama penggunaan.
5. *Maintenance*: Lakukan pemeliharaan berkala untuk memastikan kinerja optimal dan perbaiki masalah selama penggunaan aplikasi. Tetap berkomunikasi dengan Dinsos PPPA Kota Probolinggo untuk memenuhi persyaratan dan kebutuhan yang berlaku.

2.10 Tahap Pengujian

Tahapan yang dilakukan setelah program selesai dibuat yang bertujuan untuk memastikan sistem berfungsi sebagaimana seharusnya. Pengujian sistem dilakukan untuk mengukur performa dari *chatbot* dari segi kecepatan dan ketepatan dalam menjawab atau merespon input dari pengguna. Metode yang digunakan adalah *Blackbox Testing* yang berfokus pada masukan (*input*) yang diberikan kepada perangkat lunak dan keluaran (*output*) yang dihasilkan sebagai respons terhadap input tersebut dan *Whitebox Testing* untuk memeriksa logika, alur kontrol, struktur data, dan implementasi kode sumber untuk mengidentifikasi masalah potensial.

3. Hasil Dan Analisis

Sistem *chatbot* ini memungkinkan pengguna untuk melaporkan kasus kekerasan terhadap perempuan. Setelah *login* atau registrasi, pengguna akan diarahkan ke halaman *chat*. Masukan pengguna akan diproses dengan *preprocessing* untuk menghilangkan tanda baca dan kata tidak penting. Selanjutnya, kata-kata dalam masukan pengguna akan diproses menggunakan *enhanced confix stripping stemmer* untuk mengubahnya menjadi kata dasar. Selanjutnya, kata-kata tersebut akan dicocokkan dengan kamus data menggunakan *fuzzy string matching* berdasarkan kemiripan karakter penulisan kata dasar.

Chatbot akan memberikan respon dengan akurasi minimal 60%. Jika akurasi lebih rendah, pengguna akan diberitahu bahwa respon belum tersedia. Sistem ini dapat memberikan respon yang relevan dan akurat terkait kekerasan terhadap perempuan berdasarkan masukan pengguna.

Data yang digunakan sebagai data pengujian adalah sebanyak 90 data yang sudah dikembangkan dari 17 data yang didapatkan dari hasil wawancara terhadap petugas kantor PPPA Kota Probolinggo dari tahun 2020 hingga tahun 2021 dan disimpan di *database*. Teknik pengambilan data pada penelitian ini adalah dengan melakukan wawancara yang merupakan salah satu teknik pengumpulan data dengan metode *snowball sampling*, dimana metode ini meminta info sampel sehingga dapat mengembangkan kebutuhan sampel setelahnya. Metode ini sangat cocok dengan penelitian ini dimana topik penelitian ini mengenai hal-hal sensitif dan membutuhkan privasi para korban. Data yang terdapat di *database* dapat dilihat pada Tabel 1.

Tabel 1. Data yang terdapat di database

Domain	Pertanyaan/Pernyataan	Respon
Bantuan dan Dukungan	Bagaimana saya dapat melaporkan kasus kekerasan dalam rumah tangga?	Jika Anda atau seseorang yang Anda kenal mengalami kekerasan. . .
	Apa yang harus saya lakukan jika saya atau seseorang yang saya kenal menjadi korban kekerasan seksual?	Jika Anda atau seseorang yang Anda kenal menjadi korban kekerasan seksual. . .
Melaporkan Kekerasan	Bagaimana cara melaporkan kasus kekerasan perempuan dan anak kepada pihak berwenang?	Untuk melaporkan kasus kekerasan perempuan dan anak, Anda bisa . . .
	Apakah laporan kekerasan perempuan dan anak dapat dilakukan secara anonim?	Ya, Anda dapat melaporkan kasus kekerasan perempuan dan anak secara . . .

Penelitian ini hanya berfokus pada topik kekerasan terhadap perempuan dan anak. *Chatbot* akan merespon masukan *user* sesuai dengan kalimat yang diinputkan oleh *user*. Penerapan metode *fuzzy string matching* dan algoritma *enhanced confix stripping stemmer* pada penelitian ini dilakukan berdasarkan 90 data. Proses penerapan dimulai dengan tahapan *preprocessing*, yang terdiri dari 4 langkah, yaitu *case folding*, *punctuation removal*, *tokenizing*, dan *filtering*.

Tahap pertama, *case folding*, digunakan untuk mengubah kalimat masukan pengguna menjadi huruf kecil guna menyamakan format karakter. Langkah selanjutnya adalah *punctuation removal*, di mana tanda baca pada kalimat dihilangkan agar proses selanjutnya lebih fokus pada kata-kata yang ada. Setelah itu, dilakukan *tokenizing* untuk mengubah kalimat menjadi array kata-kata (tokens) sehingga dapat diolah lebih lanjut. Tahapan terakhir pada *preprocessing* adalah *filtering*, yakni proses untuk menghilangkan kata yang tidak penting pada suatu kalimat dengan cara mengecek kata-kata dalam suatu kalimat apakah termasuk kata penting atau tidak.

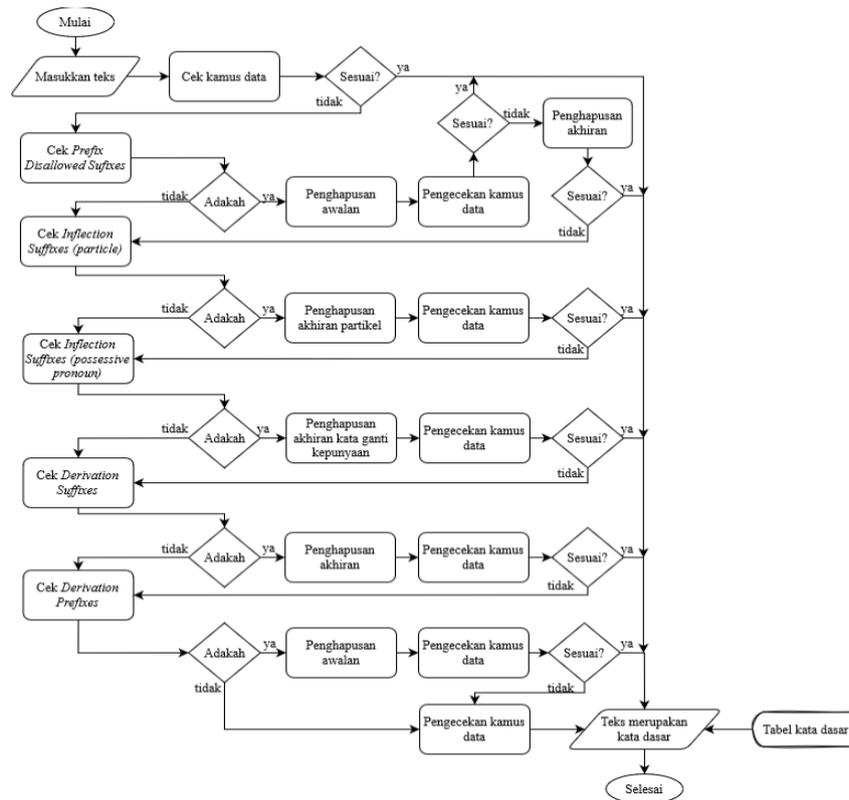
Setelah proses *preprocessing*, akan dilakukan proses *stemming* untuk mengubah kata-kata dalam kalimat menjadi kata dasar. Kata dasar yang dihasilkan dari proses *stemming* akan diperiksa dengan kamus data kata dasar yang telah tersimpan di *database*. Jika kata dasar tersebut terdapat dalam kamus data, proses dihentikan karena kata sudah sesuai dengan bentuk dasarnya. Namun, jika kata belum terdapat dalam kamus data, maka akan dilanjutkan pada penerapan algoritma *enhanced confix stripping stemmer* (ECS *Stemmer*).

Algoritma ECS *stemmer* memiliki beberapa proses, di mana kata yang tidak ditemukan dalam kamus data akan dicek terhadap aturan awalan dan akhiran yang tidak diperbolehkan, seperti be-...-i, di-...-an, ke-...-i, ke-...-kan, me-...-an, se-...-i, se-...-kan. Selanjutnya, dilakukan penghapusan awalan dan akhiran yang meliputi *Inflection Suffixes* (akhiran), *Derivation Suffixes* (akhiran), dan *Derivation Prefixes* (awalan). *Inflection Suffixes* melibatkan penghapusan partikel seperti ...-kah, ...-lah, ...-pun, ...-tah, dan penghapusan kata ganti kepemilikan seperti ...-ku, ...-nya, ...-mu. *Derivation Suffixes* melibatkan penghapusan akhiran ...-an, ...-i, ...-kan. Sedangkan, *Derivation Prefixes* melibatkan penghapusan awalan (be-..., di-..., ke-..., me-..., pe-..., se-..., dan te-...).

Setiap dilakukannya proses penghapusan awalan dan akhiran, kata akan dicek kembali dengan kamus data kata dasar. Jika kata masih belum ditemukan dalam kamus data, proses *stemming* akan berlanjut. Namun, jika kata tersebut sudah sesuai dengan kata dasar dalam kamus data, maka proses dihentikan.

Dengan penerapan metode *fuzzy string matching*, data kata atau kalimat pengguna akan dicocokkan dengan kamus data menggunakan algoritma *levenshtein distance*. Pencarian dilakukan dari kata pertama hingga kata terakhir dalam kalimat. Semakin banyak kata yang mirip dalam kamus data, data tersebut akan ditampilkan oleh algoritma ini, sehingga memperkuat kemungkinan mendapatkan respon yang tepat.

Dengan demikian, penerapan metode *fuzzy string matching* dan algoritma *enhanced confix stripping stemmer* pada penelitian ini membantu dalam menghasilkan respons *chatbot* yang lebih akurat dan sesuai dengan kata dasar yang seharusnya. Proses *stemming* juga dapat dilihat pada Gambar 3 dibawah ini.



Gambar 3. Flowchart Proses Stemming Algoritma Enhanced Confix Stripping Stemmer

3.1. Hasil Perhitungan Manual

Perhitungan manual metode yang digunakan pada penelitian ini, yakni meliputi *preprocessing*, *enhanced confix stripping stemmer*, dan *fuzzy string matching* menggunakan algoritma *levenshtein distance* dilakukan untuk mengetahui perbandingan *string* yang dimasukkan oleh *user* dengan data yang ada di database.

Tahapan yang digunakan pada *preprocessing* adalah *case folding*, *punctuation*, *tokenizing*, dan *filtering*. Selanjutnya di *stemming* dengan menerapkan *enhanced confix stripping stemmer*. Apabila sudah diproses dengan *preprocessing* dan *ECS Stemmer*, akan menghasilkan kata-kata berbentuk *array*. Kata-kata *array* tersebut akan diproses menggunakan *fuzzy string matching* dengan algoritma *levenshtein distance*. Rumus yang digunakan:

$$D(i, j) = \min D(i - 1, j) + 1 \dots \dots \dots (1)$$

$$D(i, j) = \min D(i, j - 1) + 1 \dots \dots \dots (2)$$

$$D(i, j) = \min D(i - 1, j - 1) + 1, \text{ jika } X(i) \neq Y(j) \dots \dots \dots (3)$$

$$D(i, j) = \min D(i - 1, j - 1) + 1, \text{ jika } X(i) = Y(j) \dots \dots \dots (4)$$

dimana :

D = Jarak *Levenshtein*

i = Panjang *input* pengguna/*user*

j = Panjang *string* target

Pada penelitian ini, apabila *user* memasukkan kalimat “Saya mengalami kekerasan dalam rumah tangga, apa yang harus saya perbuat?”, kalimat tersebut akan diproses dulu dengan *preprocessing* sehingga membentuk suatu *array* yang hasilnya adalah ["mengalami", "kekerasan", "rumah", "tangga", "diperbuat"]. Selanjutnya akan diproses *stemming* dengan *enhanced confix stripping stemmer* sehingga menghasilkan ["alami", "keras", "rumah", "tangga", "buat"]. Setelah itu akan dilakukan proses pencocokan *string* menggunakan *fuzzy string matching* dengan algoritma *levenshtein distance*.

Pada database terdapat tabel yang berisi domain dan juga respon yang sesuai. Kalimat “Saya mengalami kekerasan dalam rumah tangga, apa yang harus saya perbuat?” akan dicocokkan dengan data di *database*.

Hasil dari kalimat yang dimasukkan oleh *user*, yakni ["alami", "keras", "rumah", "tangga", "buat"], akan dicocokkan dengan pertanyaan pada tabel di database, dan pertanyaan tersebut akan diproses juga supaya dalam bentuk *array*. Jadi, perhitungan manualnya adalah sebagai berikut:

1. Kata-kata yang akan dibandingkan: *Input* pengguna (setelah *preprocessing* dan *enhanced confix stripping stemmer*): ["alami", "keras", "rumah", "tangga", "buat"]

2. Pertanyaan 1 (setelah preprocessing dan enhanced confix stripping stemmer): ["lapor", "keras", "rumah", "tangga"]
3. Matriks D dengan ukuran (m+1) x (n+1), di mana m adalah panjang input pengguna (5 kata) dan n adalah panjang pertanyaan (4 kata).

String Sumber: alamikerasumahtanggabu
 String Target: laporkerasumahtangga

Langkah pertama adalah membuat tabel matriks dengan dimensi yang sesuai untuk perhitungan *levenshtein distance*. Tabel matriks ini akan memiliki baris sebanyak karakter dalam string target ditambah satu, dan kolom sebanyak karakter dalam string sumber ditambah satu.

S(t)	l	a	p	o	r	k	e	r	a	s	r	u	m	a	h	t	a	n	g	g	a
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21

Gambar 4. Matriks string target

Langkah kedua adalah mengisi matriks dengan nilai *levenshtein distance*. Untuk setiap sel matriks (i, j), kita akan membandingkan karakter pada posisi i dalam string target dengan karakter pada posisi j dalam string sumber. Jika karakter sama, maka nilai *levenshtein distance* akan sama seperti pada diagonal atas kiri. Jika karakter berbeda, maka nilai *levenshtein distance* dihitung dengan mencari nilai minimum dari tiga sel sebelumnya (diagonal atas kiri, atas, dan kiri) dan ditambahkan 1.

	S(t)	l	a	p	o	r	k	e	r	a	s	r	u	m	a	h	t	a	n	g	g	a
S(s)	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21
a	1	1	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19
l	2	2	1	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19
a	3	3	1	2	2	3	4	5	6	7	7	8	9	10	11	12	13	14	15	16	17	18
p	4	4	2	2	3	3	4	5	6	7	8	8	9	10	11	12	13	14	15	16	17	18
o	5	5	3	3	2	3	4	5	6	7	8	9	9	10	11	12	13	14	15	16	17	18
r	6	6	4	4	3	2	3	4	5	6	7	8	8	9	10	11	12	13	14	15	16	17
k	7	7	5	5	4	3	2	3	4	5	6	7	8	8	9	10	11	12	13	14	15	16
e	8	8	6	6	5	4	3	2	3	4	5	6	7	8	8	9	10	11	12	13	14	15
r	9	9	7	7	6	5	4	3	2	3	4	5	6	7	8	9	10	10	11	12	13	14
a	10	10	8	8	7	6	5	4	3	2	3	4	5	6	7	8	9	10	11	11	12	13
h	11	11	9	9	8	7	6	5	4	3	3	4	5	6	7	8	9	10	10	11	12	12
t	12	12	10	10	9	8	7	6	5	4	4	3	4	5	6	7	8	9	9	10	11	11
a	13	13	11	11	10	9	8	7	6	5	5	4	3	4	5	6	7	8	9	9	10	11
n	14	14	12	12	11	10	9	8	7	6	6	5	4	3	4	5	6	7	8	8	9	10
g	15	15	13	13	12	11	10	9	8	7	7	6	5	4	3	4	5	6	7	8	8	9
g	16	16	14	14	13	12	11	10	9	8	8	7	6	5	4	3	4	5	6	7	8	8
a	17	17	15	15	14	13	12	11	10	9	9	8	7	6	5	4	3	4	5	6	7	8
t	18	18	16	16	15	14	13	12	11	10	10	9	8	7	6	5	4	3	4	5	6	7
a	19	19	17	17	16	15	14	13	12	11	11	10	9	8	7	6	5	4	3	4	5	6
n	20	20	18	18	17	16	15	14	13	12	12	11	10	9	8	7	6	5	4	4	5	6
g	21	21	19	19	18	17	16	15	14	13	13	12	11	10	9	8	7	6	5	5	5	6
g	22	22	20	20	19	18	17	16	15	14	14	13	12	11	10	9	8	7	6	6	6	6
a	23	23	21	21	20	19	18	17	16	15	15	14	13	12	11	10	9	8	7	7	7	7
t	24	24	22	22	21	20	19	18	17	16	16	15	14	13	12	11	10	9	8	8	8	8

Gambar 5. Matriks *levenshtein distance* string target dan string sumber

Hasil akhir dari perhitungan adalah nilai *levenshtein distance* di sel terakhir (paling kanan bawah) dari matriks, yaitu 8. Ini menandakan bahwa jumlah operasi edit minimum yang diperlukan untuk mengubah string sumber "alamikerasumahtanggabu" menjadi string target "laporkerasumahtangga" adalah 8. Berikut adalah langkah-langkah penjelasan dari perhitungan nilai *levenshtein distance*:

1. Tambahkan "l" di depan string sumber: "lamikerasumahtanggabu"
2. Ganti "i" pada posisi ke-6 dengan "o": "lamokerasumahtanggabu"
3. Tambahkan "p" di depan string sumber: "plamokerasumahtanggabu"
4. Ganti "m" pada posisi ke-5 dengan "t": "platomerasumahtanggabu"
5. Hapus "u" pada posisi ke-13: "platomerasmahtanggabu"
6. Hapus "b" pada posisi ke-16: "platomerasmahtangga"
7. Hapus "u" pada posisi ke-15: "platomerasmahtangga"
8. Hapus "i" pada posisi ke-14: "platomerasmahtangga"

9. Hapus "g" pada posisi ke-12: "platomerasmahtangaat"
10. Tambahkan "a" di depan string sumber: "aplatomerasmahtangaat"

Dengan demikian, string sumber "alamikerasmahtangabuat" berhasil diubah menjadi *string* target "laporkerasumahtangga" melalui operasi-edit yang disebutkan di atas. Setelah melakukan proses ini kepada pertanyaan lainnya yang tersimpan di *database*, maka nanti akan didapatkan nilai jarak *levenshtein* terkecil, dan yang terkecil akan digunakan sebagai respon yang sesuai dan ditampilkan kepada *user*. Untuk menghitung akurasi, penelitian ini menggunakan Langkah-langkah sebagai berikut:

1. Hitung *levenshtein distance* antara *string* sumber dan *string* target menggunakan algoritma *levenshtein distance* seperti yang telah dijelaskan sebelumnya. Misalkan hasil *levenshtein distance* adalah LD (dalam kasus ini, LD = 8).
2. Hitung panjang *string* target (jumlah karakter dalam *string* target). Sebagai contoh, panjang *string* target "laporkerasumahtangga" adalah 21.
3. Hitung tingkat akurasi dengan rumus:
4. Akurasi = ((panjang *string* target - LD) / panjang *string* target) * 100
5. Di mana LD adalah *levenshtein distance* yang telah dihitung.
6. Substitusikan nilai LD dan panjang *string* target ke dalam rumus dan hitung akurasi:
7. Akurasi = ((21 - 8) / 21) * 100

$$= (13 / 21) * 100$$

$$= 0.619047619 * 100$$

$$= 61.90 \text{ (dibulatkan menjadi dua desimal).}$$

Jadi, tingkat akurasi dari input pengguna dengan pertanyaan 1 memiliki akurasi 61.90%.

3.2. Pengujian Sistem

Setelah proses pembuatan sistem selesai, selanjutnya adalah pengujian sistem. Tujuan dari pengujian ini untuk mengetahui apakah sistem berjalan dengan baik, serta mengukur performa dari *chatbot* dalam ketepatan jawaban dengan masukan *user* dan juga kecepatan dalam menjawab *inputan user*. Pengujian ini akan melihat ketepatan dalam menjawab masukan *user* dan mengukur tingkat akurasi dengan data yang ada pada *database*. Pengujian sistem *chatbot* kekerasan terhadap perempuan anak dapat dilihat pada tabel.

Tabel 2. Hasil pengujian sistem

Hasil Pengujian Sistem	
Masukan user	Saya mengalami kekerasan dalam rumah tangga, apa yang harus saya perbuat?
Hasil stemming	["alami", "keras", "rumah", "tangga", "buat"]
String di database yang memiliki akurasi tertinggi	Bagaimana saya dapat melaporkan kasus kekerasan dalam rumah tangga?
Domain	Bantuan Dan Dukungan
Respon chatbot	Jika Anda atau seseorang yang Anda kenal mengalami kekerasan. . .
Respon Sesuai?	Ya
Akurasi yang dihasilkan sistem	66.67%

Dari data yang diuji diatas, *chatbot* berhasil memberikan hasil dengan tingkat akurasi sebesar 66.67%. Pengujian dilakukan untuk memastikan *chatbot* berjalan dengan baik dan memberikan respons yang tepat sesuai masukan pengguna. Dengan metode *fuzzy string matching* menggunakan algoritma *levenshtein distance*, *chatbot* dapat mengenali kata-kata kunci dan memberikan respons relevan dari *database*. Respon *bot* yang ditampilkan hanya yang memiliki akurasi diatas 60%, dan untuk yang dibawah 60%, *bot* akan menampilkan pesan bahwa respon untuk masukan dari *user* belum tersedia.

4. Kesimpulan

Kesimpulan setelah dilakukan pengujian sistem dari penelitian ini adalah sebagai berikut:

Aplikasi *chatbot* yang menggunakan metode *fuzzy string matching*, algoritma *levenshtein distance*, dan *enhanced confix stripping stemmer* mengolah input pengguna sebelum mencocokkannya dengan *database*. Selama proses *enhanced confix stripping stemmer*, kalimat diubah menjadi *array* yang berisi kata-kata dasar yang membentuk kalimat tersebut. Hal ini meningkatkan akurasi pencocokan data dengan *database* saat metode *fuzzy string matching* dan algoritma *levenshtein distance* diterapkan. Dalam pengujian, perhitungan manual dan perhitungan oleh sistem menghasilkan perbedaan sebesar 4,77%. Perbedaan ini bisa disebabkan oleh beberapa faktor, seperti perbedaan dalam pemrosesan input dan teks target oleh *chatbot* dan perhitungan manual.

Acknowledgements

Penelitian ini didukung oleh Program Studi Sistem Informasi, Fakultas Ilmu Komputer, Universitas Jember.

Daftar Pustaka

- [1] Adamopoulou, E., & Moussiades, L. (2020). An Overview of Chatbot Technology. IFIP International Federation for Information Processing.
- [2] Adawiyah, R., & Saragih, N. E. (2022). Implementasi Algoritma Levenshtein Distance Dalam Mendeteksi Plagiarisme. *Journal Computer Science and Information Technology(JCoInT)*.
- [3] Fadzhiah, Y. N., Rasim, & Fitrajaya, E. (2018). Penerapan Algoritma Enhanced Confix Stripping dalam Pengukuran Keterbacaan Teks Menggunakan Gunning Fog Index. *Jurnal Aplikasi dan Teori Ilmu Komputer*.
- [4] Fauzy, M. N., & Kusriani. (2019). Chatbot Menggunakan Metode Fuzzy String Matching Sebagai Virtual Assistant Pada Pusat Layanan Informasi Akademik. *Jurnal INFORMA Politeknik Indonusa Surakarta*.
- [5] Hidayatullah, N., Wibawa, A. P., & Rosyid, H. A. (2019). Penerapan ECS Stemmer untuk Modifikasi Nazief & Adriani Berbahasa Jawa. *Jurnal RESTI (Rekayasa Sistem dan Teknologi Informasi)*.
- [6] Huang, W., Hew, K. F., & Fryer, L. K. (2021). Chatbots for language learning—Are they really useful? A systematic review of chatbot-supported language learning. *JCAL - Journal of Computer Assisted Learning*.
- [7] Jasmi, R., Baizal, Z., & Richasdy, D. (2022). Question Answering Chatbot using Ontology for History of the Sumedang Larang Kingdom using Cosine Similarity as Similarity Measure. *JURNAL MEDIA INFORMATIKA BUDIDARMA*.
- [8] Jumardi, R., Farokhah, L., & Maghfirah. (2020). Kolaborasi Digital Signage dan Chatbot Messenger Sebagai Layanan Penyedia Informasi Akademik. *JURNAL MEDIA INFORMATIKA BUDIDARMA*.
- [9] Prianto, C., Andarsyah, R., & Harani, N. H. (2022). Rancang Bangun Kamus Digital Berbasis Chatbot Menggunakan Pendekatan Pattern Matching. *JURNAL MEDIA INFORMATIKA BUDIDARMA*.
- [10] Rohman, I. A., Aqharabah, B. H., & Solekan, R. (2023). Chatbot Untuk Cek Persediaan Stok Barang Menggunakan Metode Fuzzy String Matching Berbasis Mobile. *PROSIDING SEMINAR NASIONAL TEKNOLOGI DAN SAINS*.
- [11] Sanjung, A. G., & Norhikmah. (2021). Analisis Ketetapan Respon Chatbot Menggunakan Algoritma Boyer Moore. *SISTEMASI: Jurnal Sistem Informasi*.
- [12] Supiyandi, Zen, M., Rizal, C., & Eka, M. (2022). Perancangan Sistem Informasi Desa Tomuan Holbung Menggunakan Metode Waterfall. *JURIKOM (Jurnal Riset Komputer)*.
- [13] Tahitoe, A. D., & Purwitasari, D. (2021). Implementasi Modifikasi Enhanced Confix Stripping Stemmer Untuk Bahasa Indonesia Dengan Metode Corpus Based Stemming.
- [14] Wardani, N. W., & Nugraha, P. S. (2020). Stemming Teks Bahasa Bali dengan Algoritma Enhanced Confix Stripping. *International Journal of Natural Sciences and Engineering*.
- [15] Widiyanto, T. (2022). Rancang Bangun Aplikasi Chatbot Untuk Pendukung Perdagangan Dengan Menggunakan Metode Fuzzy String Matching-RUP (Studi Kasus : Warung Kedelai Edamame Kalibagor). *Evolusi: Jurnal Sains dan Manajemen*.
- [16] Wirawan, I. A., & Paryatna, I. M. (2020). Implementation of the String Matching Method on Anggah-Ungguhing Balinese Language Dictionary. *International Journal of Interactive Mobile Technologies (iJIM)*.