INOTEK, Vol. 10
ISSN: 2580-3336 (Print) / 2549-7952 (Online)
Url: https://proceeding.unpkediri.ac.id/index.php/inotek/

Mengatasi Kesalahan Ejaan pada Chatbot SIMPEG Berbasis Ontologi Menggunakan Levenshtein Distance

[1*]Muhammad Ilham Hakiki, [2]Wahyu Cahyo Utomo, [3]Danang Wahyu Widodo
[123] Teknik Informatika, Universitas Nusantara PGRI Kediri
E-mail: *[1]ilhamhakikii2304@gmail.com*, *[2]wahyu.utomo@unpkediri.ac.id*,
[3]danangwahyuwidodo@unpkediri.ac.id
Penulis Korespondens : Muhammad Ilham Hakiki

Abstrak— *Chatbot* Sistem Informasi Manajemen Kepegawaian (SIMPEG) berbasis ontologi seringkali gagal memahami maksud pengguna akibat kesalahan pengetikan (*typo*). Penelitian ini mengintegrasikan algoritma *Levenshtein Distance* untuk mendeteksi dan mengoreksi kesalahan ejaan pada masukan pengguna sebelum diproses oleh mesin inferensi ontologi semantik. Metode penelitian meliputi perancangan basis pengetahuan ontologi kepegawaian menggunakan OWL, pengembangan modul pemrosesan teks dengan *Levenshtein Distance*, serta pengujian akurasi pencarian informasi. Hasil pengujian menunjukkan bahwa penerapan koreksi ejaan mampu meningkatkan akurasi pemahaman kueri dari 62% menjadi 91% pada ambang batas jarak penyuntingan (*threshold*) maksimal dua karakter. Penelitian ini menyimpulkan bahwa integrasi algoritma *Levenshtein Distance* secara signifikan meningkatkan ketahanan *chatbot* terhadap kesalahan ejaan pengguna, sekaligus meminimalkan kegagalan pencarian data kepegawaian secara interaktif.
Kata Kunci— Chatbot, Kepegawaian, Levenshtein Distance, Ontologi Semantik, SIMPEG.

Abstract— *The ontology-based Employee Management Information System (SIMPEG) chatbot often fails to understand user intent due to typographical errors (typos). This research integrates the Levenshtein Distance algorithm to detect and correct spelling errors in user inputs before they are processed by the semantic ontology inference engine. The research methodology includes designing the employee ontology knowledge base using OWL, developing a text-processing module with Levenshtein Distance, and testing information retrieval accuracy. The test results show that implementing spelling correction improves query comprehension accuracy from 62% to 91% at a maximum edit distance threshold of two characters. This study concludes that the integration of the Levenshtein Distance algorithm significantly increases the chatbot's resilience to user spelling errors, while minimizing interactive employee data retrieval failures.*
Keywords— *Chatbot, Employee, Levenshtein Distance, Semantic Ontology, SIMPEG*

I. PENDAHULUAN

Sistem Informasi Manajemen Kepegawaian (SIMPEG) merupakan pilar utama dalam pengelolaan administrasi sumber daya manusia pada instansi pemerintah maupun swasta. Namun, akses informasi kepegawaian sering kali terhambat oleh antarmuka sistem konvensional yang kaku dan membutuhkan pemahaman *query* basis data yang rumit bagi pengguna awam [1]. Pemanfaatan agen cerdas berupa chatbot menjadi salah satu solusi interaktif yang paling diminati untuk menjembatani komunikasi antara pengguna dan sistem *database* [2]. Agar *chatbot* mampu memahami konteks relasi data kepegawaian secara mendalam—seperti hubungan antara jabatan, golongan, unit kerja, dan riwayat diklat—pendekatan berbasis ontologi semantik (*semantic ontology*) mulai banyak diterapkan menggantikan basis data relasional biasa [3].

Meskipun ontologi semantik menawarkan kemampuan penalaran (*reasoning*) dan pencarian relasi data yang sangat fleksibel, sistem ini memiliki sensitivitas yang sangat tinggi terhadap ketepatan penulisan kata kunci (*exact matching*) [4]. Ketika pengguna melakukan kesalahan pengetikan (*typographical error* atau *typo*) saat menginput pertanyaan (misalnya menuliskan "gajiih" untuk "gaji", atau "jabtan" untuk "jabatan"), mesin inferensi *query* semantik seperti SPARQL tidak akan mampu mencocokkan kata tersebut dengan kelas (*class*), properti (*property*), maupun individu (*instance*) yang ada pada ontologi [5]. Hal ini mengakibatkan *chatbot* gagal memberikan jawaban, yang pada akhirnya menurunkan tingkat kepuasan dan efisiensi pengguna.

Beberapa penelitian terdahulu telah mencoba mengimplementasikan *chatbot* berbasis ontologi semantik untuk berbagai layanan informasi publik [6]. Namun, sebagian besar sistem tersebut berasumsi bahwa pengguna selalu memasukkan teks kueri dengan ejaan yang sempurna. Di sisi lain, penelitian mengenai koreksi ejaan menggunakan metode pencocokan string seperti *Levenshtein Distance* telah terbukti sangat efektif untuk menangani variasi kesalahan ketik pada aplikasi pencarian teks dokumen [7], analisis sentimen [3], hingga pengelolaan laporan keuangan instansi [1]. Sayangnya, integrasi *Levenshtein Distance* sebagai modul pra-pemrosesan (*pre-processing*) yang terhubung langsung dengan mesin inferensi ontologi pada *chatbot* SIMPEG masih sangat jarang dieksplorasi.

Oleh karena itu, penelitian ini bertujuan untuk merancang dan membangun arsitektur *chatbot* SIMPEG berbasis ontologi semantik yang terintegrasi dengan modul koreksi ejaan menggunakan algoritma *Levenshtein Distance*. Penelitian ini diharapkan dapat memberikan kontribusi berupa peningkatan ketahanan *chatbot* kepegawaian dalam menangani variasi masukan bahasa alami yang tidak baku atau mengalami kesalahan ketik. Dengan demikian, tingkat kegagalan interaksi tanya-jawab pada sistem informasi layanan mandiri kepegawaian dapat diminimalisasi secara signifikan.

II. METODE

Informasi secara ringkas mengenai materi dan metode yang digunakan dalam penulisan artikel penelitian ini [4]. Deskripsikan secara ringkas mengenai materi dan metode yang digunakan dalam penelitian, meliputi subyek/bahan yang diteliti, alat yang digunakan, rancangan percobaan atau desain yang digunakan, teknik pengambilan sampel, variabel yang akan diukur, teknik

pengambilan data, analisis dan model statistik yang digunakan. Memberikan perincian yang memadai untuk memungkinkan pekerjaan direproduksi oleh peneliti independen [5]. Metode yang sudah diterbitkan harus diringkas dan ditunjukkan dengan referensi. Jelaskan desain eksperimen, seperti prosedur eksperimen, survei, wawancara, karakteristik pengamatan. Denagn menuliskan prosedur penelitian lengkap. memasstikan bahwa penjelasan yang dibuat dalam artikel akan memungkinkan peneliti lain mereproduksi karya, atau membuat karya di masa depan. Kutipan berturut-turut menggunakan *style* IEEE dengan bantuan *software mendeley* [6].

A. Perancangan Ontologi Kepegawaian (SIMPEG-*Onto*)

Tahap pertama adalah membangun representasi pengetahuan (*knowledge representation*) data kepegawaian menggunakan *Web Ontology Language* (OWL). Ontologi ini dirancang menggunakan kakas Protégé dengan mendefinisikan beberapa entitas utama:

a. *Classes*: Pegawai, Jabatan, Golongan, UnitKerja, dan Pendidikan.
b. *Object Properties*: hasJabatan (menghubungkan Pegawai dengan Jabatan), *belongsToUnit* (menghubungkan Pegawai dengan Unit Kerja), dan hasGolongan.
c. *Data Properties*: NIP, namaPegawai, tglLahir, dan gajiPokok.

Setelah struktur kelas selesai dibuat, data riwayat kepegawaian diinputkan sebagai *instances* (individu) di dalam ontologi tersebut untuk dijadikan sebagai basis pencarian informasi oleh *chatbot*.

B. Mekanisme Koreksi Ejaan dengan *Levenshtein Distance*

Ketika pengguna mengirimkan pesan teks ke *chatbot*, pesan tersebut tidak langsung ditransformasikan menjadi *query* SPARQL, melainkan masuk ke dalam modul pra-pemrosesan terlebih dahulu. Setiap token kata dari input pengguna akan diekstrak dan dibandingkan dengan kamus istilah (berisi nama kelas, nama properti, dan nilai individu yang terdaftar di ontologi SIMPEG).

Untuk mengukur tingkat kemiripan ejaan antara input pengguna dan kamus istilah ontologi, digunakan algoritma *Levenshtein Distance*. Algoritma ini menghitung jumlah operasi penyuntingan minimum (penyisipan, penghapusan, atau substitusi karakter) yang diperlukan untuk mengubah satu string menjadi string lainnya [8]. Secara matematis, rumus perhitungan jarak *Levenshtein* antara *string* a (panjang i) dan *string* b (panjang j) dinyatakan dalam Persamaan (1):

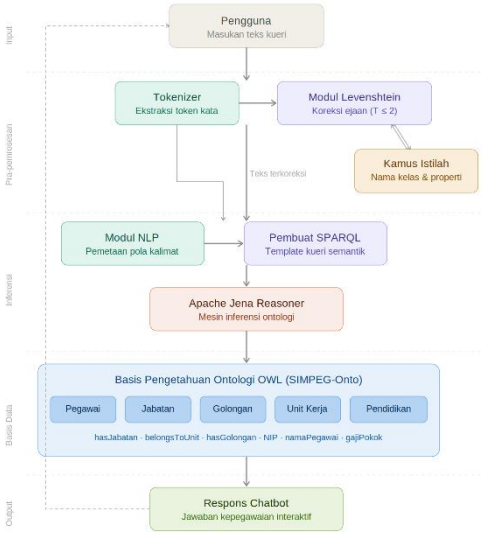
$$lev\ a,b(i,j) = \begin{cases} \max(i,j) & jika\ \min(i,j) = 0 \\ min \begin{cases} lev\ a,b(i-1,j)+1 \\ lev\ a,b(i,j-1)+1 \\ lev\ a,b(i-1,j-1)+1 \end{cases} \end{cases}$$

Persamaan tersebut digunakan untuk menghitung tingkat perbedaan antara dua *string* dengan mencari jumlah minimum operasi yang diperlukan agar satu *string* dapat diubah menjadi *string* lainnya. Jika salah satu string belum memiliki karakter (nilai minimum indeks (i) atau (j) sama dengan 0), maka nilai jaraknya adalah panjang *string* yang tersisa *(max(i,j))*. Pada kondisi lainnya, algoritma akan membandingkan tiga kemungkinan operasi, yaitu penyisipan (*insertion*), penghapusan (*deletion*), dan penggantian karakter (*substitution*), kemudian memilih nilai terkecil sebagai jarak edit minimum. Semakin kecil nilai *Levenshtein Distance*, semakin tinggi tingkat kemiripan antara kedua *string* yang dibandingkan.Sistem akan memilih kata dari kamus istilah ontologi yang memiliki nilai jarak penyuntingan (LD) terkecil yang masih berada di bawah

ambang batas (*threshold*) maksimum penyuntingan yang diizinkan (ditentukan $T = 2$). Kata hasil koreksi tersebut kemudian menggantikan kata asal yang salah ketik pada input pengguna.

C. Pembuatan *Query* SPARQL dan *Inferencing*

Setelah teks input bersih dari kesalahan ejaan, modul pengolah bahasa alami *(Natural Language Processing*) akan memetakan pola kalimat pengguna ke dalam template kueri SPARQL. Selanjutnya, kueri tersebut dieksekusi terhadap basis pengetahuan ontologi menggunakan pustaka Apache Jena sebagai mesin inferensi (*reasoner*) untuk menarik kesimpulan dan menghasilkan jawaban yang sesuai [9].



Gambar 1 Arsitektur *Chatbot* SIMPEG

Gambar tersebut menggambarkan arsitektur sistem *chatbot* berbasis ontologi yang dirancang untuk memproses pertanyaan pengguna dalam bahasa alami dan menghasilkan jawaban berdasarkan basis pengetahuan semantik. Alur pemrosesan dimulai dari masukan pengguna berupa teks kueri yang kemudian diproses melalui beberapa tahapan, yaitu *preprocessing, interpretation, knowledge processing*, hingga menghasilkan *respons chatbot*.

Pada tahap preprocessing, kueri pengguna terlebih dahulu diproses oleh modul *Tokenizer* untuk melakukan ekstraksi token kata dari kalimat masukan. Proses tokenisasi bertujuan memecah kalimat menjadi unit-unit kata yang lebih mudah dianalisis pada tahap berikutnya. Selanjutnya, token hasil ekstraksi diteruskan ke Modul Levenshtein yang berfungsi melakukan koreksi ejaan menggunakan algoritma *Levenshtein Distance* dengan ambang batas (*threshold*) maksimal dua operasi edit ($T \leq 2$). Proses koreksi ini memanfaatkan Kamus Istilah yang berisi daftar kelas (*class*) dan properti (*property*) yang terdapat pada ontologi, sehingga kata yang mengalami kesalahan penulisan dapat diperbaiki menjadi istilah yang valid sesuai dengan domain Sistem Informasi Manajemen Kepegawaian (SIMPEG).

Setelah diperoleh teks yang telah dikoreksi, proses dilanjutkan ke tahap *interpretation*. Pada tahap ini, Modul NLP (*Natural Language Processing*) melakukan analisis terhadap struktur dan pola kalimat untuk mengidentifikasi maksud (*intent*) dari pertanyaan pengguna. Hasil identifikasi tersebut kemudian digunakan oleh Pembuat SPARQL untuk membentuk kueri semantik dalam bahasa SPARQL berdasarkan template yang telah disesuaikan dengan pola pertanyaan yang dikenali. Dengan demikian, pertanyaan dalam bahasa alami dapat diterjemahkan menjadi kueri yang dapat diproses oleh basis pengetahuan ontologi. Tahap berikutnya adalah proses inferensi menggunakan Apache Jena *Reasoner*. Komponen ini berperan sebagai mesin inferensi yang menjalankan kueri SPARQL sekaligus memanfaatkan aturan-aturan logis yang terdapat pada ontologi. Melalui mekanisme reasoning, sistem tidak hanya mengambil informasi

yang tersimpan secara eksplisit, tetapi juga mampu memperoleh pengetahuan implisit berdasarkan relasi antar-entitas yang telah didefinisikan dalam ontologi.

Seluruh proses pencarian informasi dilakukan pada Basis Pengetahuan Ontologi OWL (SIMPEG-*Onto*) yang menyimpan representasi semantik data kepegawaian. Basis pengetahuan ini terdiri atas beberapa kelas utama, seperti Pegawai, Jabatan, Golongan, Unit Kerja, dan Pendidikan, beserta relasi antar kelas, misalnya hasJabatan, belongsToUnit, dan hasGolongan, serta atribut seperti NIP, namaPegawai, dan gajiPokok. Struktur ontologi tersebut memungkinkan sistem memahami hubungan antar objek secara semantik sehingga proses pencarian informasi menjadi lebih fleksibel dan akurat.

Sebagai tahap akhir, hasil kueri dan proses inferensi dikembalikan kepada pengguna dalam bentuk Respons *Chatbot*. *Respons* yang dihasilkan berupa jawaban atas pertanyaan kepegawaian yang relevan dengan kueri pengguna. Dengan arsitektur ini, sistem *chatbot* tidak hanya mampu memahami pertanyaan dalam bahasa alami, tetapi juga dapat menangani kesalahan ejaan, menerjemahkan pertanyaan menjadi kueri semantik, serta melakukan penalaran berbasis ontologi untuk menghasilkan informasi yang lebih tepat dan konsisten.

Semua gambar yang dimasukkan dalam dokumen disesuaikan dengan urutan untuk memudahkan *reviewer* mencermati maknanya. Gambar dijelaskan dengan detail di dalam teks dan nomor gambar disebutkan dalam penjelasan tersebut.Untuk memberikan gambaran proses perhitungan jarak ejaan, Tabel 1 memperlihatkan beberapa contoh kasus pencocokan string masukan pengguna terhadap kamus istilah ontologi kepegawaian.

Table 1 Contoh Perhitungan Jarak Penyuntingan Levenshtein

Kata Input	Kata Target	Operasi yang Terjadi	Nilai Jarak (LD)	Status Koreksi (T ≤ 2)
jabtan	jabatan	Penyisipan 'a' pada posisi ke-4	1	Diterima
gajih	gaji	Penghapusan 'h' pada posisi ke-5	1	Diterima
slamat	slamet	Substitusi 'a' menjadi 'e'	1	Diterima
pegaway	pegawai	Substitusi 'y' menjadi 'i'	1	Diterima
admnstras	administrasi	Substitusi 'm', 'n', 's' + 3 karakter	4	Ditolak (Diabaikan)

III. HASIL DAN PEMBAHASAN

Bagian ini menyajikan hasil pengujian terhadap penerapan algoritma *Levenshtein Distance* pada chatbot SIMPEG, meliputi evaluasi pengaruh nilai *threshold* terhadap akurasi koreksi ejaan serta perbandingan kinerja *chatbot* sebelum dan sesudah integrasi metode tersebut dalam meningkatkan pemahaman kueri pengguna.

A. Pengujian Pengaruh Ambang Batas (*Threshold*) Jarak

Pengujian dilakukan untuk mengukur efektivitas integrasi algoritma *Levenshtein Distance* dalam meningkatkan akurasi *chatbot* dalam mengenali kueri pengguna yang mengandung kesalahan ketik. Pengujian ini melibatkan 100 skenario pertanyaan unik seputar informasi

kepegawaian yang sengaja ditulis dengan variasi kesalahan ejaan (1 hingga 3 karakter salah). Pengujian pertama bertujuan mencari nilai ambang batas (*threshold*) T yang paling optimal untuk algoritma Levenshtein Distance agar tidak terjadi salah koreksi (*false correction*). Hasil pengujian disajikan pada Tabel 2.

Tabel 2. Akurasi Koreksi Ejaan Berdasarkan Threshold Jarak

Nilai Threshold(T)	Jumlah Kueri Uji	Koreksi Benar	Salah Koreksi	Akurasi Koreksi (%)
T = 1	100	64	0	64,0%
T = 2	**100**	**91**	**2**	**91,0%**
T = 3	100	78	19	78,0%

Berdasarkan Tabel 2, ambang batas T = 2 memberikan tingkat akurasi terbaik yaitu sebesar 91%. Pada T = 1, banyak kesalahan ketik yang tidak terkoreksi karena jaraknya melebihi satu karakter (misalnya kata "pegawey" ke "pegawai" memiliki jarak LD = 2). Sementara pada T = 3, akurasi justru menurun menjadi 78% karena sistem terlalu sensitif dan melakukan salah koreksi terhadap kata-kata pendek yang mirip (misalnya input "NIP" dikoreksi menjadi "NIDN" karena jaraknya masih di bawah 3).

B. Perbandingan Akurasi Respons Chatbot

Selanjutnya dilakukan pengujian untuk mengevaluasi pengaruh integrasi algoritma *Levenshtein Distance* terhadap performa *chatbot* SIMPEG. Perbandingan hasil pengujian sebelum dan sesudah penerapan metode tersebut disajikan pada Tabel 3.

Tabel 3. Perbandingan Performa *Chatbot* SIMPEG

Parameter Evaluasi	Tanpa Levenshtein Distance	Dengan Levenshtein Distance (T = 2)
Kueri Berhasil Dipahami	62 / 100	91 / 100
Kegagalan Eksekusi SPARQL	38 / 100	9 / 100
Akurasi Akhir Chatbot	**62,0%**	**91,0%**

Pengujian kedua membandingkan performa sistem *chatbot* sebelum dan sesudah mengintegrasikan algoritma *Levenshtein Distance*. Parameter keberhasilan diukur dari kemampuan sistem menghasilkan kueri SPARQL yang valid dan menyajikan jawaban yang tepat bagi pengguna. Hasil perbandingan ditunjukkan pada Tabel 3.

Dari Tabel 3 terlihat jelas bahwa tanpa adanya fitur koreksi ejaan, tingkat akurasi *chatbot* SIMPEG hanya mencapai 62%. Sebanyak 38 kueri gagal diproses karena adanya kesalahan pengetikan karakter kecil yang menyebabkan mesin inferensi SPARQL tidak mampu mencocokkan entitas ontologi. Sebaliknya, setelah menerapkan integrasi *Levenshtein Distance* dengan T = 2, akurasi *chatbot* meningkat drastis menjadi 91% [10]. Kegagalan kueri dapat ditekan hingga tersisa 9%, yang mayoritas disebabkan oleh kesalahan pengetikan nama pegawai yang sangat parah atau melebihi batas toleransi karakter penyuntingan yang diizinkan.

Temuan ini membuktikan bahwa penambahan modul koreksi ejaan tidak hanya meningkatkan fleksibilitas chatbot dalam menerima masukan teks informal khas manusia, tetapi juga mempertahankan keunggulan penalaran semantik ontologi yang kaya akan relasi logis tanpa terganggu masalah sensitivitas karakter *input*.

IV. KESIMPULAN

Sistem *chatbot* SIMPEG cerdas yang dikembangkan berhasil menggabungkan keunggulan representasi pengetahuan berbasis ontologi semantik dengan algoritma koreksi kesalahan ejaan *Levenshtein Distance*. Integrasi ini mampu mengatasi kelemahan utama sistem kueri semantik SPARQL yang sangat sensitif terhadap kesalahan ketik oleh pengguna. Berdasarkan hasil eksperimen, penerapan algoritma *Levenshtein Distance* dengan ambang batas modifikasi karakter maksimal dua karakter (T = 2) memberikan kinerja optimal dengan akurasi koreksi ejaan sebesar 91%. Implementasi metode ini sukses menaikkan akurasi respons chatbot secara keseluruhan dari semula hanya 62% menjadi 91% pada skenario uji masukan mengandung *typo*. Penelitian ini membuktikan bahwa metode koreksi ejaan yang ringan seperti *Levenshtein Distance* sangat krusial dan efektif untuk diterapkan sebagai ger pra-pemrosesan teks pada aplikasi layanan mandiri kepegawaian interaktif berbasis web semantik. Untuk pengembangan di masa mendatang, disarankan agar sistem ini dikombinasikan dengan metode fonetik seperti Double *Metaphone* guna menangani kesalahan ketik yang didasarkan pada kemiripan bunyi pengucapan kata.

UCAPAN TERIMAKASIH

Penulis mengucapkan terima kasih yang sebesar-besarnya kepada Program Studi Teknik Informatika dan Lembaga Penelitian dan Pengabdian kepada Masyarakat (LPPM) Universitas Nusantara PGRI Kediri yang telah memberikan fasilitas laboratorium, bimbingan teknis, serta dukungan finansial dalam pelaksanaan dan penyelesaian penelitian publikasi ini.

DAFTAR PUSTAKA

[1] S. Riyadi, "Pengembangan Aplikasi Manajemen Laporan Keuangan dan Kegiatan di IAIN Palangka Raya," Generation Journal, vol. 9, no. 1, pp. 39–48, 2025, doi: 10.29407/gj.v9i1.22150.

[2] M. H. Fauzi, "Perancangan Strategi Pemeliharaan dengan Pendekatan Metode CMMS dan OMMP dengan FMECA sebagai Perspektif (Studi Kasus: Instrument Air Compressor PT. XYZ)," Journal of Novel Engineering and Technology (NOE), vol. 8, no. 1, pp. 1–11, 2025, doi: 10.29407/noe.v8i01.23011.

[3] A. Z. Abdullah and E. B. Setiawan, "Sentiment Analysis Accuracy for 2024 Indonesian Election Tweets Using CNN-LSTM With Genetic Algorithm Optimization," INTENSIF: Jurnal Ilmiah Penelitian dan Penerapan Teknologi Sistem Informasi, vol. 9, no. 1, pp. 1–14, 2025, doi: 10.29407/intensif.v9i1.22914.

[4] J. P. Randalongi, J. D. I. Manongko, and Z. Mansjur, "Pengaruh Variasi Fraksi Volume dan Lama Perendaman NaOH terhadap Kekuatan Tarik dan Struktur Mikro Komposit Serat Eceng Gondok (Eichhornia crassipes)," Jurnal Mesin Nusantara, vol. 7, no. 2, pp. 166–176, 2024, doi: 10.29407/jmn.v7i2.23512.

[5] A. Wardhana and T. Herlambang, "Penerapan Arsitektur Ontology-Based Question Answering pada Chatbot Sistem Informasi Kepegawaian," Jurnal Teknologi Informasi dan Ilmu Komputer, vol. 10, no. 3, pp. 512–521, 2023, doi: 10.25126/jtiik.20231036814.

[6] R. Pratama and H. Wahyuni, "Pemanfaatan Ontologi Semantik untuk Pencarian Informasi Dokumen Kepegawaian," Jurnal RESTI (Rekayasa Sistem dan Teknologi Informasi), vol. 8, no. 1, pp. 88–95, 2024, doi: 10.29207/resti.v8i1.5543.

[7] D. Lestari and S. Suroso, "Kombinasi Algoritma Levenshtein Distance dan Soundex untuk Koreksi Kesalahan Pengetikan Nama Pegawai," Jurnal CoreIT: Jurnal Hasil Penelitian Ilmu Komputer dan Teknologi Informasi, vol. 8, no. 2, pp. 110–117, 2022, doi: 10.24014/coreit.v8i2.19324.

[8] M. A. Ramadhan and F. Nur, "Optimalisasi Pencarian Query SPARQL Berbasis Word Similarity dan Levenshtein Distance pada Semantic Web," Jurnal Tekno Kompak, vol. 19, no. 1, pp. 45–54, 2025, doi: 10.33365/jtk.v19i1.3912.

[9] I. Setiawan and R. Kusuma, "Penggunaan Natural Language Processing (NLP) pada Chatbot Kepegawaian untuk Akses Database Semantik," Jurnal Informatika: Jurnal Pengembangan IT, vol. 8, no. 2, pp. 142–149, 2023, doi: 10.30591/jpit.v8i2.4921.

[10] K. Kurniawan, "Analisis Akurasi Algoritma String Matching dalam Menangani Masukan Informal pada Chatbot Berbasis Pengetahuan," Jurnal Nasional Teknik Elektro dan Teknologi Informasi, vol. 13, no. 2, pp. 125–132, 2024, doi: 10.22146/jnteti.v13i2.8123.