

Technofac: Aplikasi Android Pencarian Fasilitas Kampus Berbasis *Speech-to-Text* dan *Natural Language Processing* (Studi Kasus: Universitas Indo Global Mandiri)

Siti Ayu Hermaliah^{*1}, Shinta Puspasari², Indah Permatasari³

^{1,2,3}Teknik Informatika, Fakultas Ilmu Komputer dan Sains, Universitas Indo Global Mandiri, Indonesia
Email: ¹2021110088@students.uigm.ac.id, ²shinta@uigm.ac.id, ³indah@uigm.ac.id

Abstrak

Aplikasi Technofac dikembangkan untuk memudahkan pencarian fasilitas kampus Universitas Indo Global Mandiri (IGM) dengan integrasi *Speech-to-Text* dan *Natural Language Processing* (NLP) melalui metode *Hybrid Keyword-Based*. Pipeline sistem meliputi *Automatic Speech Recognition*, *preprocessing* (*normalization*, *Punctuation Removal*, *Tokenization*, *Synonym Mapping*, dan *Abbreviation Expansion*), *intent detection* berbasis *keyword matching*, dan *Named Entity Recognition* (*pattern matching*, *POS tagging*, *dictionary*, serta *fuzzy matching*). Evaluasi kinerja NLP menunjukkan metode *Hybrid Keyword-Based* mencapai *precision* 0.9346, *recall* 0.9263, dan *F1-Score* 0.9297 (kategori *very good*), sedangkan metode *Substring Matching* mencapai *precision* 0.7879, *recall* 0.7512, dan *F1-Score* 0.7655 (kategori OK). Pengujian *blackbox* terhadap 13 fitur menunjukkan seluruh fungsi berjalan sesuai spesifikasi. Evaluasi *usability* menggunakan *System Usability Scale* pada 20 responden (n=20) menghasilkan skor 82.87 dengan Grade A (*Acceptable*) dan *confidence interval* 95% pada rentang 80.8–85.2. Meskipun sistem menunjukkan performa baik, terdapat 16 kasus *False Negative* akibat *query* ambigu dan 14 kasus *False Positive* akibat *misclassification* entitas. Penelitian ini menunjukkan potensi *rule-based NLP* untuk aplikasi pencarian fasilitas kampus, namun masih memerlukan *dialog management system* (*follow-up question*) dan *hierarchical entity recognition* untuk menurunkan kesalahan FP/FN.

Kata kunci: *hybrid keyword-based*, NER, NLP, *speech-to-text*, SUS, technofac

Abstract

The Technofac application was developed to facilitate the search for IGM University campus facilities with the integration of *Speech-to-Text* and NLP through the *Hybrid Keyword-Based* method. The system pipeline includes *Automatic Speech Recognition*, *preprocessing* (*normalization*, *Punctuation Removal*, *Tokenization*, *Synonym Mapping*, and *Abbreviation Expansion*), *keyword matching-based intent detection*, and *Named Entity Recognition* (*pattern matching*, *POS tagging*, *dictionary*, and *fuzzy matching*). NLP performance evaluation shows that the *Hybrid Keyword-Based* method achieves a *precision* of 0.9346, a *recall* of 0.9263, and an *F1-Score* of 0.9297 (*very good* category), while the *Substring Matching* method achieves a *precision* of 0.7879, a *recall* of 0.7512, and an *F1-Score* of 0.7655 (*OK* category). *Blackbox* testing of 13 features shows that all functions run according to specifications. *Usability* evaluation using the *System Usability Scale* on 20 respondents (n=20) resulted in a score of 82.87 with Grade A (*Acceptable*) and a 95% *confidence interval* in the range of 80.8–85.2. Although the system performed well, there were 16 cases of *False Negatives* due to ambiguous queries and 14 cases of *False Positives* due to entity *misclassification*. This study demonstrates the potential of *rule-based NLP* for campus facility search applications, but still requires a *dialog management system* (*follow-up questions*) and *hierarchical entity recognition* to reduce FP/FN errors.

Keywords: *hybrid keyword-based*, NER, NLP, *speech-to-text*, SUS, technofac

This work is an open access article and licensed under a Creative Commons Attribution-NonCommercial ShareAlike 4.0 International (CC BY-NC-SA 4.0)



1. PENDAHULUAN

Era digital kini mengalami banyak perubahan di berbagai institusi, termasuk dalam institusi pendidikan. Teknologi kecerdasan buatan (AI) semakin banyak digunakan, salah satunya melalui *Natural Language Processing* (NLP)[1]. Laporan UNESCO yang dikutip oleh Badriello menunjukkan bahwa lebih dari 40% lembaga pendidikan tinggi di negara-negara maju telah menerapkan teknologi AI dalam sistem penilaian otomatis dan pembelajaran adaptif[2]. NLP juga digunakan dalam salah satu aplikasi perintah suara yang paling terkenal yaitu Siri, yang merupakan aplikasi berbasis Android dikembangkan oleh Google[3]. Temuan-temuan tersebut mengafirmasi kontribusi strategis AI dan NLP

dalam layanan akademik serta dapat digunakan sebagai asisten virtual[4].

Di Universitas Indo Global Mandiri (UIGM), berbagai fasilitas kampus telah disediakan untuk mendukung kegiatan pembelajaran, termasuk beberapa gedung bertingkat, masing-masing dilengkapi dengan fasilitas yang memadai[5]. Namun, kompleksitas struktur gedung bertingkat dan penyebaran fasilitas di berbagai lokasi menimbulkan tantangan bagi sivitas akademika dan pemangku kepentingan dalam menemukan lokasi yang mereka butuhkan, karena tidak adanya sistem pencarian lokasi fasilitas berbasis peta fisik maupun digital[6].

Penelitian Irkham Huda memberikan gambaran bahwa pencarian lokasi berbasis bahasa natural berhasil mengubah kalimat pencarian dalam bahasa Indonesia menjadi permintaan spasial dengan tingkat presisi 0,97 dan *recall* 0,91 menunjukkan efektivitas tinggi dalam memahami *intent* pengguna[7]. Meskipun demikian, sistem tersebut masih memiliki limitasi berupa dependensi terhadap konektivitas internet untuk pemrosesan *query*. Penelitian Aditi dalam mengembangkan perintah suara dapat membantu penyandang tunanetra dalam pemanggilan suara, mengirim pesan dan mempermudah dalam menggunakan perangkat[8]. Pemanfaatan teknologi perintah suara dengan pendekatan NLP pada ponsel pintar sudah bisa dilakukan lebih dari sekadar mencari, melakukan *video call*, dan membalas pesan dengan suara, namun hanya bisa digunakan bila terhubung dengan internet.

Perkembangan NLP juga mendorong hadirnya pendekatan lain yang berfokus pada ekstraksi entitas menggunakan metode berbasis *deep learning* maupun teknik *text matching* sebagai alternatif pemrosesan yang lebih adaptif. ASR untuk bahasa Indonesia memiliki tantangan tersendiri dibandingkan dengan bahasa-bahasa mayor seperti Inggris atau Mandarin. Bahasa Indonesia memiliki karakteristik unik seperti struktur fonetik yang relatif konsisten, penggunaan afiks yang kompleks (awalan dan akhiran), serta keberagaman dialek dan aksen di berbagai wilayah Nusantara[9]. Meskipun demikian, perkembangan ASR bahasa Indonesia terus mengalami kemajuan dengan semakin banyaknya dataset audio bahasa Indonesia yang tersedia untuk melatih model, serta kontribusi dari berbagai institusi riset dan perusahaan teknologi lokal[10].

Penelitian Affandi berhasil mengimplementasikan sistem *Named Entity Recognition* (NER) untuk bahasa Indonesia menggunakan *fine-tuning model IndoBERT* dengan dataset *NER-Grit* menghasilkan *f1-score* 74,20% dan *loss* 0,3673, menunjukkan efektivitas pendekatan ini dalam mengatasi variasi struktur linguistik bahasa yang lebih adaptif terhadap variasi domain dan dialek bahasa Indonesia[11]. Penelitian Putra mengenai Integrasi pencocokan teks dan teknologi pengenalan suara dalam aplikasi pencarian data yang menggunakan *Python* menunjukkan hasil yang sangat memuaskan dengan tingkat akurasi, presisi, *recall*, dan *F1 score* mencapai 1. 0, serta efisiensi waktu pengujian rata-rata 15.861 detik per pencarian, membuktikan bahwa pendekatan *text matching* dapat meningkatkan daya tanggap pencarian dan kemudahan penggunaan[12]. Penelitian Afifah mengimplementasikan *chatbot* menggunakan algoritma BERT yang mampu mengenali berbagai pola kalimat dengan akurasi 85%, menunjukkan efektivitas *deep learning* dalam memahami variasi bahasa natural meskipun memerlukan *resource* komputasi yang lebih besar[13].

Berdasarkan penelitian-penelitian tersebut, terlihat bahwa setiap pendekatan memiliki kelebihan dan keterbatasan masing-masing, sehingga pemilihan metode harus disesuaikan dengan kebutuhan aplikasi dan keterbatasan perangkat. Selain itu, belum ada penelitian yang secara spesifik yang mengembangkan aplikasi pencarian fasilitas kampus yang siap pakai pada perangkat dengan sumber daya terbatas. Gap inilah yang menjadi landasan kontribusi penelitian ini.

Penelitian ini bertujuan mengembangkan aplikasi pencarian fasilitas kampus dengan pendekatan multimodal yang mengintegrasikan *Automatic Speech Recognition*, *Speech-to-Text*, dan *Text-to-Speech*, serta mengimplementasikan NLP berbasis *rule-based* dengan *Intent Detection* menggunakan *Keyword Matching* dan komponen *Named Entity Recognition* (NER) untuk pemrosesan *query* bahasa Indonesia dalam domain pencarian fasilitas kampus. Pendekatan *keyword matching* menjadi alternatif yang efektif untuk penelitian berskala kecil karena tidak memerlukan sumber daya komputasi besar, dapat berjalan secara lokal, dan mudah diimplementasikan[14]. Cakupan awal penelitian ini dibatasi pada Gedung L, Gedung B, dan Gedung C karena gedung-gedung tersebut merupakan pusat aktivitas dengan *traffic* tertinggi, dan memiliki data fasilitas lengkap[15]. Penelitian ini menggunakan pendekatan *Rapid Application Development* (RAD) yang meliputi *Requirements Planning*, *System Design*, *Development*, dan *Implementation*[16]. Temuan dari penelitian ini diharapkan dapat memberikan dukungan bagi komunitas akademik dan pihak terkait di kampus untuk menemukan fasilitas.

2. METODE PENELITIAN

2.1 Metode Pengumpulan Data

Pengumpulan data dilakukan dengan beberapa metode. Pertama, dilakukan studi literatur, yaitu peneliti mengumpulkan dan menganalisis berbagai sumber referensi yang relevan dengan topik penelitian, seperti jurnal ilmiah dan artikel penelitian[17]. Kedua, dilakukan pendekatan langsung melalui wawancara, di mana peneliti melakukan wawancara dengan Ibu Sylvia, Kepala BAU di Universitas Indo Global Mandiri, untuk memperoleh informasi mengenai fasilitas kampus. Ketiga, dilakukan observasi lapangan di Universitas IGM yang terletak di Jalan Jenderal Sudirman No. 629 Palembang. Keempat, kuesioner digunakan untuk mengumpulkan data dengan memberikan daftar pertanyaan tertulis kepada responden guna memperoleh informasi yang diperlukan dalam penelitian[18].

2.1.1 Dataset

Perancangan dataset dilakukan dengan memperhatikan variasi linguistik bahasa Indonesia dan representasi dari berbagai kategori fasilitas kampus. Dataset dalam penelitian ini terdiri dari 31 entitas fasilitas kampus yang tersebar di Gedung L, B, dan C Universitas IGM. Tabel 1 menunjukkan representasi dari beberapa fasilitas yang digunakan dalam penelitian.

Tabel 1. Representasi Fasilitas Kampus UIGM

Nama Fasilitas	Kategori Fasilitas	Lokasi	Koordinat (Lat, Long)
Ruang B0202	Ruang Kelas	Gedung B lantai 2	-2.962875°, 104.740787°
Lab. Komputer	Laboratorium	Gedung C lantai 5	-2.965548°, 104.740983°
Perpustakaan	Fasilitas Umum	Gedung C lantai 2	-2.963568°, 104.74058°
Front-office	Ruang Administrasi	Gedung L lantai 1	-2.962878°, 104.740791°

Tabel 1 menampilkan 4 fasilitas yang dipilih sebagai representasi dari keseluruhan 31 fasilitas kampus yang menjadi dataset penelitian ini. Pemilihan keempat fasilitas tersebut menggunakan metode *purposive sampling* dengan mempertimbangkan keragaman fungsi dan karakteristik penggunaan fasilitas di lingkungan kampus UIGM.

2.1.2 Perancangan Variasi Utterance dan Anotasi Dataset

Untuk keperluan pengujian sistem NLP, dibuat 7 variasi kalimat pencarian untuk setiap fasilitas, menghasilkan total 217 *utterance* (31 fasilitas × 7 variasi). Setiap variasi dirancang untuk mencakup berbagai pola linguistik bahasa Indonesia[19]. Tabel 2 menunjukkan 7 variasi kalimat *utterance* mencari fasilitas untuk 2 representatif fasilitas dari setiap kategori.

Tabel 2. Variasi *Utterance* per Fasilitas

Fasilitas	Formal	Informal	Typo	Singkatan	Tanya langsung	filler words	imperatif
Ruang B0202	Saya ingin mencari Ruang B0202	Ruang B0202 dimana ya?	Luang B0202 dimana?	Dimana RB0202	Ruang B0202 Ada Dimana?	Eh, Ruang B0202 Dimana?	Tunjukkan Ruang B0202
Front office	Saya ingin mencari Ruang Front Office	Front office dimana ya?	Fron office dimana?	Dimana FO	Front Office Ada Dimana?	heh, dimana front office?	Tunjukkan Front Office

Proses anotasi dataset dilakukan secara manual oleh dua orang annotator berdasarkan *guideline* pelabelan yang telah disusun untuk menjaga konsistensi dan objektivitas klasifikasi. Kedua annotator yang telah familiar dengan NLP. *Guideline* anotasi disusun mencakup dua komponen utama pelabelan *intent* dan pelabelan NER.

a. Pelabelan *Intent*

Pelabelan *intent* bertujuan untuk mengklasifikasikan maksud pengguna ke dalam salah satu dari tiga

kategori *intent* (mencari_fasilitas, navigasi_ke_lokasi, informasi_fasilitas) berdasarkan indikator kata kunci. Pendekatan ini menggunakan *multi-class classification*, di mana setiap *utterance* diklasifikasikan ke dalam tepat satu dari tiga kelas *intent* yang saling eksklusif (*mutually exclusive*). Tabel 3 menunjukkan aturan pelabelan *intent*.

Tabel 3. Aturan Pelabelan *Intent*

Label <i>Intent</i>	Indikator kata kunci	Contoh
mencari_fasilitas	dimana, cari, tunjukkan, ada dimana	Dimana lab komputer?,
navigasi_ke_lokasi	navigasi, arah, rute, antar, pergi ke	Bagaimana rute ke lab jaringan?,
informasi_fasilitas	informasi, detail, tentang, kapasitas, jam	Kapasitas lab komputer berapa?

Utterance yang mengandung indikator untuk dua *intent*, akan dipilih *intent* dominan berdasarkan kata kunci yang muncul lebih dulu. Jika *Utterance* dengan *typo* atau salah eja seperti di cantumkan di Tabel 2, maka dianotasi sesuai maksud asli yang dapat diinterpretasi. Untuk kata-kata mengandung *filler word* diabaikan dalam penentuan *intent*. *Utterance* dengan singkatan dianotasi dengan *meng-expand* singkatan untuk memahami maksud.

Contoh 1. *utterance* dengan dua *intent*

Cari informasi lokasi lab komputer → cari + lokasi → mencari_fasilitas

Cari informasi lokasi lab komputer → informasi → informasi_fasilitas

Pada contoh, "cari" + "lokasi" lebih dominan daripada "informasi", maka yang dipilih adalah *intent* mencari_fasilitas.

Contoh 2. *Utterance* dengan *typo*

Luang B0202 dimana? → Ruang B0202 dimana?

Contoh 3. *Utterance* dengan singkatan

FO → *Front Office*

b. Pelabelan NER

Pelabelan NER bertujuan untuk menandai dan mengekstrak entitas fasilitas kampus dari *utterance* pengguna. Sistem mengidentifikasi tiga tipe entitas: *FACILITY* (fasilitas kampus), *BUILDING* (gedung), dan *FLOOR* (lantai). Evaluasi dilakukan menggunakan *multi-class classification*, di mana setiap *utterance* diklasifikasikan ke dalam salah satu dari 4 kategori fasilitas. Tabel 4 menunjukkan distribusi dataset yang digunakan dalam penelitian ini per kategori fasilitas.

Tabel 4. Distribusi Dataset Fasilitas Kampus

Kategori	Fasilitas	<i>Utterance</i>	Jumlah Entitas	Jumlah <i>Utterance</i>
Ruang Kelas	Ruang B0202	Dimana Ruang B0202?	14	98
Laboratorium	Lab.Komputer	Cari Lab Komputer	6	42
Fasilitas Umum	Kantin	Kantin Dimana?	8	56
Ruang Administrasi	<i>Front-office</i>	Tunjukkan Ruangan FO	3	21
Total	-	-	31	217

Proses pelabelan menggunakan format *span-based annotation* dengan indeks karakter untuk menandai posisi setiap entitas dalam teks.

```
utterance: Dimana Lab Komputer?,
intent: mencari_fasilitas,
entities: [
  {
    text: Lab Komputer,
    start: 7,
    end: 19,
    type: FACILITY,
    canonical_form: Laboratorium Komputer
  }
]
```

2.2 Pemrosesan Data

Beberapa data diperlukan untuk mengembangkan aplikasi Technofac ini. Dengan mengumpulkan informasi tentang fasilitas kampus, seperti nama gedung, deskripsi, gambar, dan koordinat geografis (*latitude* dan *longitude*), serta mengompilasinya dalam format JSON sebagai representasi awal dataset untuk memastikan konsistensi struktur data. Dataset JSON diimpor ke dalam database MySQL 8.0 yang dijalankan pada server lokal menggunakan XAMPP.

2.3 Metode Pengembangan Aplikasi

Metode yang diterapkan dalam pengembangan aplikasi ini adalah *Rapid Application Development* yang mencakup tahap *Requirements Planning*, *System Design*, *Development*, dan *Implementation*[20].

2.3.1 Requirements Planning

Proses *requirements planning* ini mencakup analisis kebutuhan meliputi analisis perangkat dan analisis fitur aplikasi. Analisa perangkat meliputi Analisa penggunaan perangkat keras, perangkat lunak dan analisa fitur. Tabel 5 menunjukkan *software* dan *hardware* pendukung dan Tabel 6 menyajikan fitur-fitur yang digunakan untuk aplikasi Technofac.

Tabel 5. *Software* dan *Hardware* Pendukung

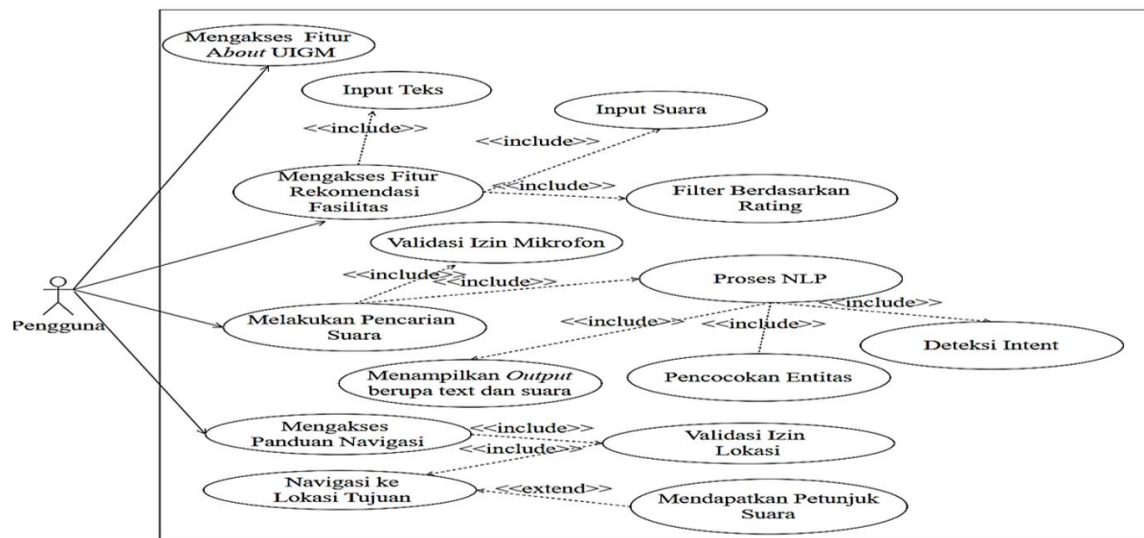
Kategori	Nama/Spesifikasi
<i>Software Code Editor</i>	VS Code
<i>Android SDK</i>	Android Studio
<i>Framework UI</i>	Flutter
<i>Framework backend/ Database</i>	Laravel/ MySQL
<i>Operating System/ Processor</i>	Windows 11/ Intel Core i5
<i>RAM/SSD</i>	8 GB / 256GB

Tabel 6. Analisa Fitur Aplikasi

Fitur	Deskripsi	Prioritas	Teknologi
<i>Speech to text</i>	Mengkonversi perintah suara menjadi teks	<i>High</i>	<i>speech_to_text, Android Speech Services</i>
<i>Text to speech</i>	Memberikan <i>output</i> berupa suara	<i>High</i>	<i>Flutter TTS</i>
Pencarian	Mencari fasilitas berdasarkan input suara	<i>High</i>	<i>Custom NLP engine (rule-based)</i>
Detail fasilitas	Menampilkan informasi fasilitas	<i>High</i>	<i>Flutter UI, Image loading</i>
Navigasi	Navigasi Suara ke Lokasi	<i>High</i>	<i>Flutter TTS</i>
Peta Visual	Peta Visual Gedung	<i>Low</i>	<i>SSID Mapping, wifi_scan 0.4.1</i>

2.3.2 System Design

System design adalah proses perancangan struktur dan alur kerja aplikasi guna memenuhi kebutuhan pengguna[21]. *Unified Modeling Language* seperti diagram *use case* diterapkan untuk memvisualisasikan model sistem dengan menggunakan *Draw.io*, serta perancangan antarmuka aplikasi meliputi *wireframe* dan *mockup* menggunakan *figma*[16]. Aktor utama pada sistem ini adalah pengguna aplikasi. Gambar 1 menunjukkan *use case diagram* aplikasi Technofac.



Gambar 1. Use Case Diagram Aplikasi Technofac

2.3.3 Development

Dalam proses pengembangan, dilakukan slicing UI menggunakan bahasa pemrograman Dart dengan framework Flutter, yaitu proses menerjemahkan desain antarmuka dari Figma menjadi baris-baris kode Flutter menggunakan struktur *widget-tree* dengan *state management Provider pattern*. Pengkodean dilakukan secara bertahap pada setiap fitur mulai dari *Splash Screen*, *Onboarding*, *Homepage*, hingga fitur pencarian suara, navigasi, dan peta visual gedung. Pengembangan dilakukan dengan Flutter SDK 3.22.0 Dart 3.4.0, backend Laravel 10, MySQL 8.0, testing pada Android Emulator API 29 dan Xiaomi Redmi 9 A (Android 10). Backend dikembangkan menggunakan bahasa PHP dengan arsitektur *Model-View-Controller (MVC)*, yang menyediakan layanan *web service lokal (localhost)* berupa REST API untuk operasi CRUD. API tersebut diakses oleh aplikasi Flutter melalui HTTP request menggunakan paket *http*, sehingga data fasilitas dapat diambil selama proses inferensi *intent* dan NER.

2.3.4 Implementation

Pada tahap ini, implementasi dilakukan dengan menggabungkan antar halaman menggunakan mekanisme navigasi. Integrasi antar halaman meliputi halaman *Splash Screen*, *Onboarding*, *Homepage*, serta navigasi antar fitur-fitur yang ada di *Homepage*. Selain itu, pada tahap ini juga diimplementasikan fitur pencarian dengan NLP. Tahapan pada sistem NLP ini mencakup ASR, *Preprocessing Teks*, *Intent Detection*, dan NER berbasis aturan sebagai berikut:

1. Record audio when user taps mic
2. Convert audio → text using ASR (*speech_to_text* v6.1.1)
3. Preprocess text: normalization, remove punctuation, tokenization, Synonym Mapping, Expansion
4. Intent detection: match keywords against intent dictionary
5. NER: apply POS tagging (e.g. WRB+NNP → facility) → pattern matching (ReqEx)→ dictionary lookup and expansion, Fuzzy Matching
6. If facility found return facility info & show navigation button

2.4 ASR (Automatic Speech Recognition)

ASR adalah teknologi yang memungkinkan komputer mengubah suara yang diucapkan oleh manusia menjadi teks yang bisa dibaca. Cara kerja sistem ASR adalah dengan menganalisis gelombang suara yang masuk, kemudian memproses gelombang suara tersebut melalui algoritma pembelajaran mesin yang kompleks untuk mengenali pola-pola fonetik, kata, dan kalimat[23]. Dalam pengembangan aplikasi Technofac, fitur pengenalan suara ASR diimplementasikan dengan menggunakan library *speech_to_text* versi 6.1.1 untuk Flutter yang mendukung mode *offline* sehingga tidak ada pengiriman data ke *cloud server*. Paket bahasa berukuran sekitar 45 MB yang diunduh otomatis saat aplikasi pertama kali dijalankan jika belum tersedia di perangkat.

2.5 Preprocessing Teks

Preprocessing teks adalah serangkaian tahapan sistematis untuk membersihkan, mengubah, dan menormalisasi data teks mentah menjadi format yang terstruktur dan siap diproses oleh sistem komputasi[7]. Setelah teks hasil ASR diperoleh, dilakukan *preprocessing* untuk menormalkan dan membersihkan teks mentah agar lebih mudah diproses oleh algoritma berikutnya. Tahapannya meliputi *normalization*, *Punctuation Removal*, *Tokenization*, *Synonym Mapping*, dan *Abbreviation Expansion*. Tabel 7 menyajikan tahapan *preprocessing*.

Tabel 7. Tahapan *preprocessing*

Tahap	Tujuan	Input	Output
<i>Lowercase Normalization</i>	Mengubah seluruh teks menjadi huruf kecil	Dimana Laboratorium Komputer?	dimana laboratorium komputer?
<i>Punctuation Removal</i>	Menghilangkan tanda baca	Dimana Laboratorium Komputer?	dimana laboratorium komputer
<i>Tokenization</i>	Memisahkan teks menjadi token/kata	Dimana Laboratorium Komputer?	[dimana, laboratorium, komputer]
<i>Synonym Mapping</i>	Menormalisasi variasi kata ke bentuk standar	lab	laboratorium
<i>Abbreviation Expansion</i>	<i>mengubah singkatan menjadi bentuk lengkapnya</i>	lab komp	laboratorium komputer

2.6 Intent Detection

Tahap *intent detection* bertujuan mengidentifikasi maksud pengguna dari input teks yang telah diproses melalui *preprocessing*. Pada aplikasi Technofac, *intent detection* menggunakan metode *keyword matching* berbasis *dictionary* untuk mencocokkan token hasil *preprocessing* dengan kamus *intent* yang telah didefinisikan[24]. Tabel 8 menunjukkan *intent dictionary*.

Tabel 8. *Intent Dictionary*

<i>Intent Name</i>	<i>Keywords</i>
mencari_fasilitas	[dimana, lokasi, cari, temukan, tunjukkan, ada dimana]
navigasi_ke_lokasi	[navigasi, arah, rute, antar, pergi ke]
informasi_fasilitas	[informasi, detail, tentang, kapasitas, jam]

2.7 NER (Named Entity Recognition)

Setelah *intent* berhasil dideteksi, tahap NER dapat mengidentifikasi suatu kata benda termasuk personal, lokasi, waktu, atau organisasi[25]. Dalam penelitian ini, identifikasi entitas akan berfokus pada objek atau subjek fasilitas, serta lokasi fasilitas yang digunakan untuk pembentukan *query*. NER pada aplikasi Technofac menggunakan pendekatan *rule-based* yang menggabungkan empat teknik yaitu POS (*Part-of-Speech*) *tagging*, *pattern matching* berbasis *regex*, *dictionary lookup*, dan *fuzzy matching*.

2.7.1 POS (Part-of-Speech) Tagging

POS *Tagging* digunakan untuk mengidentifikasi kategori gramatikal setiap token dalam kalimat input yang mendukung bahasa Indonesia. Proses POS *tagging* dilakukan dengan melakukan tokenisasi teks, kemudian setiap token dianalisis untuk menentukan kategori gramatikal yang paling sesuai berdasarkan konteks kalimat. Tabel 9 menunjukkan POS *tags* yang digunakan dalam implementasi NER untuk mengidentifikasi kandidat entitas fasilitas kampus.

Tabel 9. Penggunaan POS *Tags*

Tag	Deskripsi	Contoh Kata
WRB	<i>Wh-Adverb</i> (kata tanya)	dimana, kemana
VB	<i>Verb base form</i> (kata kerja dasar)	pergi, cari, ke, menuju
NN	<i>Noun</i> (kata benda umum)	lab, ruang, gedung
NNP	<i>Proper Noun</i> (nama khusus)	Perpustakaan, Laboratorium, Komputer

2.7.2 Pattern Matching dengan Regex

Pattern matching menggunakan ekspresi reguler (*regex*) untuk mencocokkan pola kalimat tertentu berdasarkan kombinasi POS *tag* dan kata kunci. *Regex* diimplementasikan menggunakan class *RegExp* bawaan Dart dengan *flag case-insensitive* untuk menangani variasi penulisan huruf besar atau kecil[10]. Tabel 10 menunjukkan aturan *pattern matching* dengan *regex pattern* untuk menangkap berbagai variasi input pengguna.

Tabel 10. Aturan *Pattern Matching* dan Simbol *Regex*

Kategori	Pattern / Simbol	Deskripsi
Pattern POS	WRB + NNP/NN	Pola kata tanya + nama benda
	VB + NNP/NN	Pola kata kerja + nama benda
	WRB + VB? + NNP	Pola kata tanya + kata kerja + nama benda
	NN + NN	Pola dua kata benda berurutan
Simbol <i>Regex</i>	\b	Batas kata (<i>word boundary</i>)
	\s+	Satu atau lebih spasi
	\s*	Nol atau lebih spasi
	\w	Huruf, angka, <i>underscore</i>
	\w+	Satu atau lebih karakter <i>word</i>
	?	Opsional (boleh ada/tidak)
	()	<i>Grouping / capture</i>

2.7.3 Dictionary lookup and Abbreviation Expansion

Dictionary lookup digunakan untuk memvalidasi apakah kandidat entitas yang telah diekstrak melalui POS *tagging* dan *pattern matching* merupakan nama fasilitas yang valid dalam *database* sistem yang berisi 31 fasilitas kampus UIGM. Sebelum *dictionary lookup*, sistem menerapkan *abbreviation expansion* terhadap singkatan umum untuk menormalisasi input ke bentuk standar sesuai *naming convention database*. Tabel 11 menunjukkan *rules mapping abbreviation expansion*.

Tabel 11. *Dictionary Lookup* dengan *Abbreviation Expansion*

Singkatan	Expansion
Lab	Laboratorium
FO	Front office
BAU	Biro Administrasi Umum
BAA	Biro Administrasi Akademik

2.7.4 Fuzzy Matching

Secara linguistik, fuzzy dapat diartikan secara samar, artinya algoritma fuzzy merupakan sebuah algoritma yang tidak definitif[12]. Teknik ini mengukur tingkat kemiripan (*similarity*) antara input *query* dan nama fasilitas dalam *database* untuk menentukan apakah keduanya merepresentasikan fasilitas yang sama meskipun ada perbedaan penulisan. Tabel 12 menyajikan contoh *fuzzy matching*.

Tabel 12. Contoh *Fuzzy Matching*

input query	canonical name	match?	keterangan
perpustakaan	perpustakaan	yes	typo: huruf 'a' hilang
fron office	front office	yes	typo: huruf 't' hilang
lep komputer	lab komputer	yes	typo: huruf 'e' seharusnya 'a'

2.8 Testing

Aplikasi yang dibuat akan diuji dengan metode pengujian *Blackbox* untuk menilai kemampuan fungsionalnya, serta dengan menghitung nilai *System Usability Scale* (SUS), untuk menilai seberapa berguna sistem komputer dari sudut pandang pengguna[17].

2.8.1 Blackbox Testing

Blackbox Testing dilakukan untuk memverifikasi bahwa setiap fitur aplikasi berfungsi sesuai dengan spesifikasi yang telah ditetapkan, meliputi navigasi antar halaman, fitur pencarian dengan ASR dan NLP, serta *respons* sistem terhadap berbagai skenario input pengguna. Pengujian ini melibatkan fokus pada validasi *input-output* sistem untuk memastikan aplikasi memberikan hasil yang benar sesuai dengan requirements yang telah didefinisikan.

2.8.2 System Usability Scale

System Usability Scale dipakai untuk menilai sejauh mana aplikasi dapat digunakan dengan cara mengisi kuesioner standar yang berisi 10 pertanyaan menggunakan skala Likert 1-5. Kuesioner SUS telah diterjemahkan ke Bahasa Indonesia menggunakan *back-translation method* dan divalidasi oleh dua ahli bahasa independen dengan nilai *Cronbach's Alpha score* sebesar 0.841, menunjukkan bahwa instrumen *valid* untuk konteks Indonesia[27] dan digunakan *Standard deviation* untuk mengukur penyebaran skor individual terhadap mean[28].

a. Perhitungan *Mean*

Rumus yang digunakan untuk menghitung rata-rata skor SUS adalah sebagai berikut:

$$\bar{x} = \frac{\sum x}{n} \quad (1)$$

Dalam Persamaan (1), $\bar{x}$ menunjukkan nilai rata-rata, $\sum x$ adalah total semua skor SUS, dan n merujuk pada jumlah responden yang mengikuti pengujian.

b. Perhitungan *Standard Deviation*

$$\sigma = \sqrt{\frac{\sum (xi - \bar{x})^2}{n-1}} \quad (2)$$

Dalam Persamaan (2), σ adalah standar deviasi, xi adalah skor SUS responden ke- i , $\bar{x}$ adalah rata-rata skor SUS, n adalah Jumlah responden, dan $n - 1$ adalah *degrees of freedom* (derajat kebebasan).

2.9 Evaluasi

Evaluasi sistem dilakukan untuk mengukur performa NLP berdasarkan input *query* pengguna hasil konversi dari *voice input* melalui ASR. Pengukuran performa menggunakan *confusion matrix* yang merepresentasikan hubungan antara prediksi sistem dengan *ground truth annotations* [18].

- True Positive* (TP): Total jumlah kasus yang diidentifikasi sebagai kelompok positif dan memang benar sesuai dengan *ground truth*.
- False Positive* (FP): Total jumlah kasus yang diidentifikasi sebagai kelompok positif padahal kenyataannya bukan kategori tersebut.
- False Negative* (FN): Total jumlah kasus yang diidentifikasi sebagai kelompok negatif padahal sebenarnya termasuk kategori positif.

Berdasarkan *confusion matrix*, dihitung tiga metrik evaluasi, dapat dilihat pada persamaan berikut.

$$Precision = \frac{TP}{TP+FP} \quad (3)$$

Precision mengukur ketepatan prediksi sistem (seberapa banyak prediksi positif yang benar). Dalam aplikasi pencarian fasilitas, *precision* tinggi berarti sistem jarang memberikan hasil yang tidak relevan.

$$Recall = \frac{TP}{TP+FN} \quad (4)$$

Recall mengindikasikan sistem mampu mendeteksi sebagian besar kategori yang seharusnya dikenali atau mengekstrak sebagian besar entitas yang ada. Dalam konteks aplikasi, *recall* berarti sistem jarang gagal mengenali *intent* pengguna atau entitas fasilitas yang disebutkan.

$$F1\ Score = 2 \times \frac{Recall \times Precision}{Recall+Precision} \quad (5)$$

F1-score digunakan untuk menilai kinerja sistem secara keseluruhan dengan menggabungkan *precision* dan *recall*, sehingga mencerminkan kemampuan sistem NLP dalam menangkap semua entitas atau intent yang relevan sekaligus meminimalkan kesalahan prediksi.

3. HASIL DAN PEMBAHASAN

Berikut ini adalah hasil dari implementasi antarmuka aplikasi pencarian fasilitas kampus berbasis Android, serta hasil implementasi NLP pada fitur pencarian fasilitas. Selain itu, juga disajikan hasil pengujian dan evaluasi terhadap sistem yang telah dikembangkan.

3.1 Hasil Implementasi Antarmuka Aplikasi

3.1.1 Antarmuka Halaman *Splash Screen* dan *Onboarding*

Antarmuka halaman *Splash Screen* berfungsi sebagai tampilan awal saat aplikasi dijalankan pertama kali. Antarmuka ini menggunakan warna utama putih dan menampilkan logo serta nama aplikasi yaitu TechnoFac di tengah layar. Gambar 2 menunjukkan hasil implementasi antarmuka halaman *Splash Screen*.



Gambar 2. Antarmuka Halaman *Splash Screen*

Onboarding aplikasi ini terdiri dari tiga halaman: halaman pertama berisi sambutan singkat tentang Technofac, halaman kedua menunjukkan cara menggunakan perintah suara, dan halaman ketiga menampilkan contoh perintah pencarian fasilitas kampus. Gambar 3, 4, dan 5 menyajikan antarmuka *onboarding*.



Gambar 3. Antarmuka *Onboarding* 1



Gambar 4. Antarmuka *Onboarding* 2



Gambar 5. Antarmuka *Onboarding* 3

3.1.2 Antarmuka Halaman *Homepage* dan Menu Tentang Kampus UIGM

Antarmuka *homepage* aplikasi Technofac menampilkan beberapa pilihan menu utama. Antarmuka halaman Tentang Kampus UIGM berisi informasi dasar mengenai Universitas IGM. Gambar 6 menunjukkan hasil implementasi antarmuka *homepage*, dan Gambar 7 menunjukkan hasil implementasi antarmuka menu Tentang Kampus UIGM.



Gambar 6. Antarmuka *Homepage*



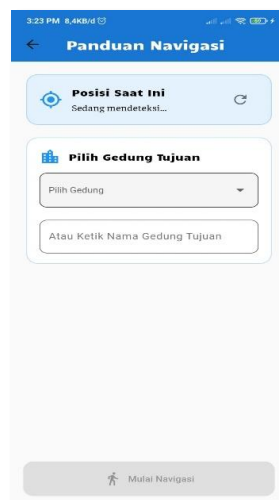
Gambar 7. Antarmuka *About UIGM*.

3.1.3 Antarmuka Halaman Pencarian dan Panduan Navigasi

Pada halaman ini, pengguna memiliki opsi untuk melakukan pencarian dengan suara. Sistem mengolah suara yang diberikan dengan memanfaatkan teknologi *Speech-to-Text* dan NLP untuk memahami tujuan pengguna berdasar kata kunci atau sinonim. Selanjutnya, sistem menunjukkan hasil pencarian yang mencakup informasi mengenai fasilitas yang relevan, seperti nama fasilitas, tempat, penjelasan, dan gambar. Untuk fitur Peta visual, sistem mengembangkan peta *offline* berbasis *Rule-based SSID Mapping* yang memanfaatkan *WiFi scanning* untuk mengidentifikasi zona lokasi (gedung) berdasarkan *pattern matching* nama SSID access points kampus. Gambar 8, 9, dan 10 yang memperlihatkan hasil implementasi tampilan pencarian, hasil implementasi tampilan navigasi.



Gambar 8. Antarmuka Pencarian



Gambar 9. Antarmuka Navigasi



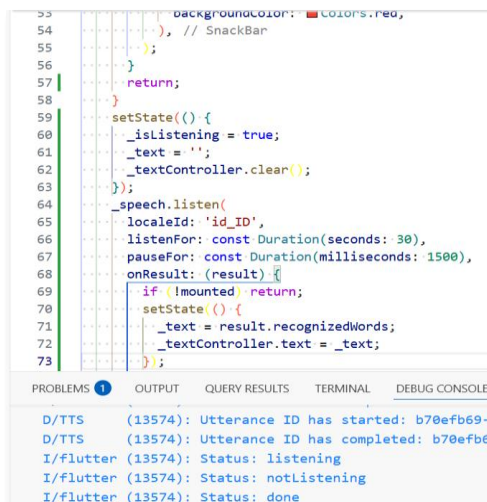
Gambar 10. Antarmuka Peta Visual

3.2 Hasil Implementasi NLP Pada Aplikasi Technofac

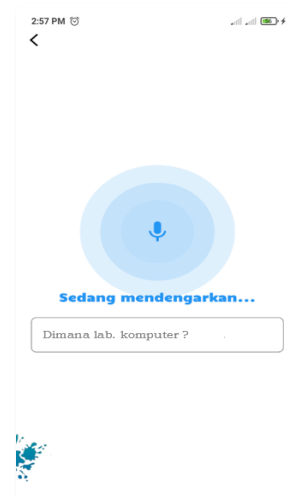
Dalam aplikasi Technofac, penggunaan NLP dilakukan dengan metode ASR, Deteksi intent, dan NER. Di bawah ini adalah metode NLP yang diterapkan dalam aplikasi Technofac.

3.2.1 Implementasi ASR

Dalam pengembangan aplikasi Technofac, fitur pengenalan suara ASR diterapkan dengan memanfaatkan pustaka *speech_to_text* versi 6.1.1 untuk Flutter yang mendukung mode *offline* sehingga tidak ada pengiriman data ke *cloud server*. Gambar 11 Menunjukkan Tampilan implementasi ASR melalui VSCode dan Gambar 12 menunjukkan implementasi fitur *speech recognition* pada antarmuka aplikasi.



Gambar 11. Implementasi kode ASR



Gambar 12. Implementasi Fitur *Speech Recognition* pada antarmuka aplikasi

Library STT ini berfungsi sebagai antarmuka (*wrapper*) untuk *engine* pengenalan suara bawaan perangkat Android *Speech Services*. Model bahasa menggunakan paket bahasa Indonesia *offline* yang diunduh melalui *SpeechToText.instance.initialize(onDevice: true)*. Paket bahasa berukuran sekitar 45 MB yang diunduh otomatis saat aplikasi pertama kali dijalankan jika belum tersedia di perangkat. Konfigurasi meliputi pengaturan *locale* "id-ID" untuk bahasa Indonesia, *sample rate* 16 kHz dengan durasi maksimal *recording* 60 detik dan memerlukan izin *RECORD_AUDIO* dari pengguna. Konfigurasi izin akses mikrofon di *AndroidManifest.xml* di implementasikan agar aplikasi Flutter dapat merekam suara, mendukung fitur *ASR offline*, dan memastikan akses legal ke perangkat. Teknologi Pengenalan Suara ini bertindak sebagai penghubung antara ucapan pengguna dan aplikasi. Hasil dari proses ini, yaitu berupa transkrip teks yang digunakan untuk identifikasi intent dan pengenalan nama entitas.

3.2.2 Implementasi *preprocessing*

Setelah teks hasil ASR diperoleh, dilakukan *preprocessing* untuk menormalkan dan membersihkan teks mentah. Tahapan *preprocessing* yang meliputi *lowercase normalization* untuk menyeragamkan semua huruf menjadi huruf kecil, *punctuation removal* untuk menghilangkan tanda baca yang tidak diperlukan, *tokenization* untuk memecah teks menjadi *array token individual*, *synonym mapping* untuk memetakan kata-kata sinonim ke bentuk standar, dan *abbreviation expansion* untuk mengubah singkatan menjadi bentuk lengkapnya. Setiap tahapan dirancang untuk mengurangi variasi bentuk kata dan meningkatkan konsistensi data sehingga sistem dapat memproses *query*. Berikut proses *preprocessing*:

Input: Dimana lab. komputer

Preprocessing:

1. *Lowercase*: dimana lab. komputer
2. *Punctuation Removal*: dimana lab komputer
3. *Tokenization*: [dimana, lab, komputer]
4. *Synonym Mapping & Abbreviation Expansion*: [dimana, laboratorium, komputer]

Output : dimana laboratorium komputer

3.2.3 Implementasi *Intent Detection*

Pada aplikasi Technofac, *intent detection* menggunakan metode *keyword matching* berbasis *dictionary* untuk mencocokkan token hasil *preprocessing* membandingkan token-token dari *query* pengguna dengan daftar *keyword* yang telah dipetakan ke masing-masing kategori *intent*. Proses eksekusi dimulai dengan memasukkan *utterance* pengguna, kemudian dilakukan *preprocessing* untuk menyiapkan teks sebelum diteruskan ke tahap *Intent Detection*. Berikut proses eksekusi *intent detection*:

Input: Dimana lab. komputer

Preprocessing

1. *Lowercase*: dimana lab. komputer
2. *Punctuation Removal*: dimana lab komputer
3. *Tokenization*: [dimana, lab, komputer]
4. *Synonym Mapping & Abbreviation Expansion*: [dimana, laboratorium, komputer]

Output : dimana laboratorium komputer

Intent Detection

1. *Check mencari_fasilitas keywords*: “dimana” (*match found*), *Score*: 1/3 *tokens matched*
2. *Check navigasi_ke_lokasi keywords*: *No match found*, *Score*: 0/3 *tokens matched*
3. *Check informasi_fasilitas keywords*: *No match found*, *Score*: 0/3 *tokens matched*
4. *detectedIntent* = mencari_fasilitas

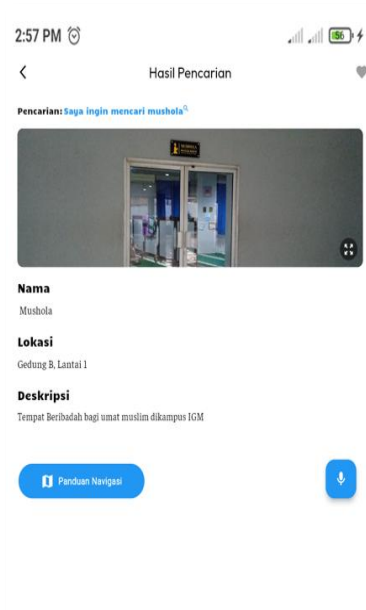
3.2.4 Implementasi NER

Implementasi NER pada aplikasi Technofac diuji menggunakan berbagai variasi *utterance* pengguna yang mencakup kasus normal dan kasus dengan *typo* atau kesalahan penulisan. Pengujian ini mendemonstrasikan bagaimana setiap komponen NER (*POS tagging*, *pattern matching*, *dictionary lookup*, *abbreviation expansion*, dan *fuzzy matching*) bekerja secara berurutan dalam mengekstrak entitas fasilitas kampus. Tabel 13 menunjukkan proses NER dengan berbagai skenario input pengguna.

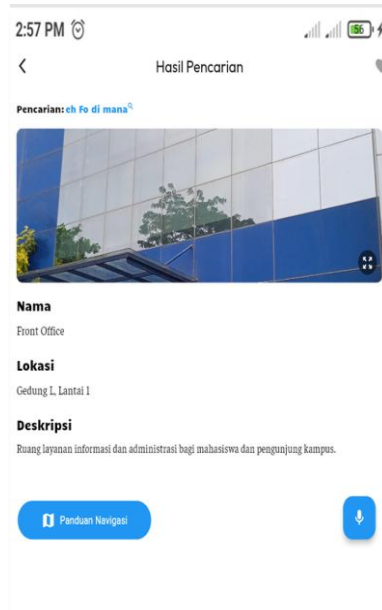
Tabel 13. Skenario NER

No.	POS Tagging	Pattern Matching (Regex)	Dictionary Lookup	Fuzzy Matching
1	[Saya] [ingin mencari][mushola] [PRP] [VB] [NN]	<i>Pattern</i> : VB + NN <i>Regex</i> : \b(mencari)s+(\w+)	Ditemukan didatabase	-
2	[Aula][dimana][ya] [NNP][WRB][UH]	<i>Pattern</i> : NNP + WRB <i>Regex</i> : \b(\w+)s+(dimana)	Ditemukan didatabase	-
3	[Kantin][dimana] [NN] [WRB]	<i>Pattern</i> : NN + WRB <i>Regex</i> : \b(\w+)s+dimana	Ditemukan didatabase	-
4	[Dimana] [teater] [WRB] [NN]	<i>Pattern</i> : WRB + NN <i>Regex</i> : \b(dimana)s+(\w+)	Ditemukan didatabase	-
5	[Perpustakaan][Ada] [Dimana] [NNP][VB][WRB]	<i>Pattern</i> : NNP + VB + WRB <i>Regex</i> : \b(\w+)s+(ada)? s*(dimana)	Ditemukan didatabase	perpustakaan vs perpustakaan memiliki kecocokan
6	[Eh][FO][Dimana] [UH][NNP][WRB]	<i>Pattern</i> : NNP + WRB <i>Regex</i> : \b(\w+)s+dimana	<i>Abbreviation</i> : FO → front office Ditemukan didatabase	-
7	[Tunjukkan] [Lep] [Komputer] [VB] [NN] [NN]	<i>Pattern</i> : VB + NN + NN <i>Regex</i> : \b(tunjukkan)s+(\w+s+\w+)	Ditemukan didatabase	lep komputer vs lab komputer → Then <i>abbreviation</i> : lab → laboratorium memiliki kecocokan

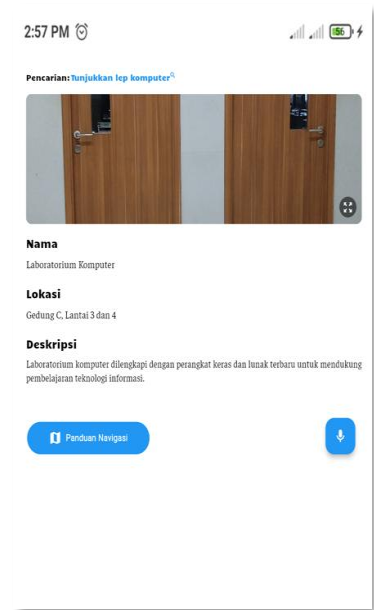
Berdasarkan skenario pada Tabel 13, contoh implementasi NER dapat dilihat pada gambar berikut ini:



Gambar 13. Implementasi NER Skenario No.1



Gambar 14. Implementasi NER Skenario No.5



Gambar 15. Implementasi NER Skenario No.7

3.3 Hasil Pengujian

3.3.1 Blackbox Testing

Uji *black box* menerapkan metode *Use Case*, yang mengevaluasi seluruh fungsi perangkat lunak dengan melakukan proses sistem dari awal sampai akhir. Hasil dari pengujian *black box* ditampilkan pada Tabel 14 di bawah ini.

Tabel 14. Hasil Pengujian *Blackbox*

Test Case	Aktivitas Pengujian	Hasil yang Diharapkan	Kesimpulan
<i>Splash Screen</i>	Aplikasi dibuka pertama kali	<i>Splash screen</i> tampil 2–3 detik lalu pindah ke <i>Onboarding</i>	Berhasil
<i>Onboarding</i>	Melihat <i>slide</i> , <i>swipe</i> Lanjut, tekan Mulai	Semua <i>slide</i> tampil benar, tombol Mulai menuju <i>Homepage</i>	Berhasil
<i>Homepage</i>	Menunggu <i>loading</i>	4 menu utama tampil	Berhasil
Cari Fasilitas	Klik Cari Fasilitas	Navigasi ke halaman pencarian suara	Berhasil
Izin Mikrofon	Tap ikon mikrofon pertama kali	<i>Pop-up</i> izin mikrofon muncul	Berhasil
ASR (<i>Speech-to-Text</i>)	Pengguna berbicara	Teks transkripsi tampil	Berhasil
Suara Tidak Terdeteksi	Pengguna tidak bicara / suara tidak jelas / <i>noise</i> terlalu tinggi	Muncul <i>pop-up</i> Suara tidak dikenali, dengan pilihan Coba Lagi atau Batal	Berhasil
Hasil Pencarian Ditemukan	<i>Entity valid</i>	Tampil info fasilitas + tombol navigasi	Berhasil
Tombol Navigasi	Klik Panduan Navigasi	Muncul panduan suara (<i>text-to-speech</i>)	Berhasil
Fitur WiFi & Lokasi	Akses fitur navigasi	<i>Permission</i> lokasi & WiFi; jika WiFi tidak aktif muncul peringatan	Berhasil
Marker Posisi	Terhubung ke SSID WiFi kampus	Marker biru tampil posisi user, Marker merah posisi gedung tujuan	Berhasil

Peta Visual Gedung	Klik pilihan gedung	Peta visual gedung dan lantai tampil	Berhasil
--------------------	---------------------	--------------------------------------	----------

Pengujian *blackbox* aplikasi dilakukan pada 13 fitur yang dikembangkan. Setiap fitur diuji berdasarkan fungsinya untuk memastikan sistem merespons tindakan pengguna seperti yang diharapkan. Hasil dari pengujian tersebut menunjukkan bahwa semua fitur yang diuji berfungsi dengan baik dan mampu memberikan hasil yang sesuai.

3.3.2 System Usability Scale

Pengujian SUS dilakukan dari perspektif pengguna akhir, sehingga hasil penilaian mencerminkan pengalaman mereka[27]. Metode ini menggunakan 10 pernyataan dalam kuesioner berbentuk skala likert sebagai instrumen penilaian, pernyataan tersebut adalah:

- Q1: Saya berpikir akan menggunakan sistem ini lagi.
Q2: Saya merasa sistem ini rumit untuk digunakan.
Q3: Saya merasa sistem ini mudah untuk digunakan.
Q4: Saya membutuhkan bantuan dari orang lain atau teknisi dalam menggunakan sistem ini.
Q5: Saya merasa fitur-fitur sistem ini berjalan dengan semestinya.
Q6: Saya merasa ada banyak hal yang tidak konsisten (tidak serasi) pada sistem ini.
Q7: Saya merasa orang lain akan memahami cara menggunakan sistem ini dengan cepat.
Q8: Saya merasa sistem ini membingungkan.
Q9: Saya merasa tidak ada hambatan dalam menggunakan sistem ini.
Q10: Saya perlu membiasakan diri terlebih dahulu sebelum menggunakan sistem ini.

Pengujian *usability* diberikan kepada 20 responden mahasiswa aktif Universitas IGM dari berbagai program studi (D3 dan S1) angkatan Tahun 2021-2024. Prosedur pengujian diawali dengan *briefing* selama 10 menit untuk menjelaskan fitur-fitur aplikasi dan tujuan pengujian, dilanjutkan dengan *hands-on testing* selama 15 menit dimana responden mencoba menggunakan aplikasi Technofac. Selama *hands-on testing*, masing-masing responden dapat melakukan 10-11 kali pencarian fasilitas kampus. Pengujian diakhiri dengan pengisian kuesioner SUS selama 5 menit untuk mengukur tingkat kepuasan dan kemudahan penggunaan aplikasi, sehingga total durasi pengujian adalah ± 25 menit per responden.

Kuesioner SUS menggunakan skala likert 5 level, dimana responden diminta untuk memberikan penilaian berdasarkan pilihan STS (Sangat Tidak Setuju), TS (Tidak Setuju), RG (Ragu-ragu), S (Setuju), dan SS (Sangat Setuju) terhadap 10 pernyataan yang ada. Untuk pertanyaan yang memiliki nomor ganjil (1, 3, 5, 7, 9), nilai yang diberikan oleh responden akan dikurangi satu. Sementara itu, untuk pertanyaan dengan nomor genap (2, 4, 6, 8, 10), nilai yang diperoleh dihitung dengan cara mengurangkan lima dari skor yang diberikan. Total dari seluruh skor pernyataan kemudian dikalikan dengan 2,5 untuk mendapatkan nilai SUS. Tabel 15 menunjukkan hasil perhitungan skor SUS yang telah dikurangi berdasarkan penomoran ganjil genap.

Tabel 15. Hasil Perhitungan Nilai SUS

Respon den	Skor Hasil Hitung										Nilai	Juml ah(x)	$(x - \bar{x})^2$
	Q1	Q2	Q3	Q4	Q5	Q6	Q7	Q8	Q9	Q10			
R1	4	2	4	3	4	3	4	2	3	3	32	80	9
R2	4	3	3	3	3	3	4	2	3	3	31	77.5	30.25
R3	3	3	3	3	4	3	3	3	4	2	31	77.5	30.25
R4	3	2	4	3	4	3	4	3	4	2	32	80	9
R5	3	3	4	3	4	3	3	3	3	2	31	77.5	30.25
R6	4	3	4	3	4	2	4	4	4	3	35	87.5	20.25
R7	4	3	4	3	4	3	4	4	4	2	35	87.5	20.25
R8	4	3	4	3	4	4	4	4	4	2	36	90	49
R9	4	2	4	3	4	4	4	4	4	2	35	87.5	20.25
R10	4	2	4	3	4	3	4	3	4	3	34	85	4

R11	3	3	3	3	4	3	3	3	4	2	31	77.5	30.25
R12	4	2	4	3	4	4	4	3	4	3	35	87.5	20.25
R13	4	3	4	2	4	4	4	4	4	3	36	90	49
R14	4	3	4	2	4	4	4	3	4	3	35	87.5	20.25
R15	4	3	4	3	4	3	3	2	4	3	33	82.5	0.25
R16	3	3	4	3	3	3	3	3	4	2	31	77.5	30.25
R17	4	3	3	3	4	2	4	2	4	3	32	80	9
R18	4	2	4	3	4	3	4	3	4	3	34	85	4
R19	3	3	4	3	3	3	3	4	3	2	31	3	30.25
R20	4	2	4	3	4	3	4	3	4	3	34	85	4
Jumlah											1660	420	
Mean ($\bar{x}$)											83		

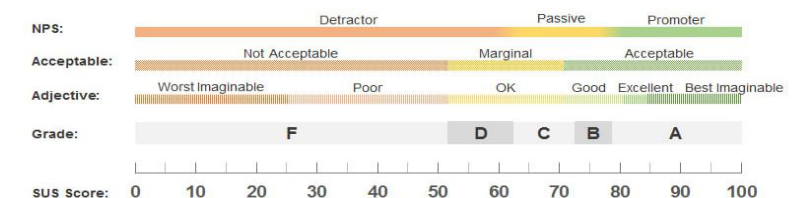
Berdasarkan hasil survei yang diberikan kepada 20 responden, *Standard Deviation*, *Confidence Interval* dapat dilihat pada perhitungan berikut:

$$\text{Standard Deviation} = \sigma = \sqrt{\frac{\sum (x_i - \bar{x})^2}{n-1}} = \sqrt{\frac{420}{19}} \approx 4.70$$

$$\text{Standard Error} = SE = \frac{SD}{\sqrt{n}} = \frac{4.70}{\sqrt{20}} \approx 1.05$$

$$\begin{aligned} \text{Confidence Interval} = CI &= \bar{x} \pm t_{0.025, df} \times SE = \bar{x} \pm t_{0.025, 19} \times SE \\ &= 83 \pm 2.093 \times 1.05 \\ &= 83 \pm 2.20 = [80.8, 85.2] \end{aligned}$$

Selanjutnya, menentukan penilaian berdasarkan kategori skala interpretasi SUS, ditunjukkan pada Gambar 16 di bawah ini



Gambar 16. Skala Interpretasi SUS[30]

Berdasarkan hasil analisis, batas error dalam estimasi skor SUS dihitung menggunakan *Standard Error* (SE) dan *Confidence Interval* (CI) untuk memastikan tingkat presisi pengukuran. Nilai *Standar deviasi* sebesar 4.70 menunjukkan tingkat dispersi data yang relatif rendah, yang mengindikasikan adanya konsistensi persepsi responden mengenai aspek usability sistem. Nilai standar deviasi yang kecil ini menunjukkan bahwa pengalaman pengguna bersifat homogen[31] dengan *Standard Error* sebesar ± 1.05 menunjukkan tingkat variabilitas estimasi *mean* yang rendah sehingga rata-rata skor SUS dapat dianggap stabil dan representatif terhadap sampel.

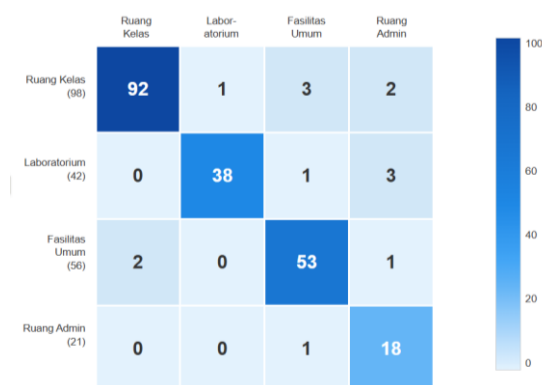
df/alpha	0.4	0.25	0.1	0.05	0.025	0.01	0.005
1	0.3249	1.0000	3.0777	6.3138	12.7062	31.8205	63.6567
2	0.2887	0.8165	1.8856	2.9200	4.3027	6.9646	9.9248
3	0.2767	0.7649	1.6377	2.3534	3.1825	4.5407	5.8409
4	0.2707	0.7407	1.5332	2.1318	2.7765	3.7470	4.6041
5	0.2672	0.7267	1.4759	2.0150	2.5706	3.3649	4.0321
6	0.2648	0.7176	1.4398	1.9432	2.4469	3.1427	3.7074
7	0.2632	0.7111	1.4149	1.8946	2.3646	2.9980	3.4995
8	0.2619	0.7064	1.3968	1.8595	2.3060	2.8965	3.3554
9	0.2610	0.7027	1.3830	1.8331	2.2622	2.8214	3.2498
10	0.2602	0.6998	1.3722	1.8125	2.2281	2.7638	3.1693
11	0.2596	0.6974	1.3634	1.7959	2.2010	2.7181	3.1058
12	0.2590	0.6955	1.3562	1.7823	2.1788	2.6810	3.0545
13	0.2586	0.6938	1.3502	1.7709	2.1604	2.6503	3.0123
14	0.2582	0.6924	1.3450	1.7613	2.1448	2.6245	2.9768
15	0.2579	0.6912	1.3406	1.7531	2.1315	2.6025	2.9467
16	0.2576	0.6901	1.3368	1.7459	2.1199	2.5835	2.9208
17	0.2573	0.6892	1.3334	1.7396	2.1098	2.5669	2.8982
18	0.2571	0.6884	1.3304	1.7341	2.1009	2.5524	2.8784
19	0.2569	0.6876	1.3277	1.7291	2.0930	2.5395	2.8609

Gambar 17. Distribusi-t [32]

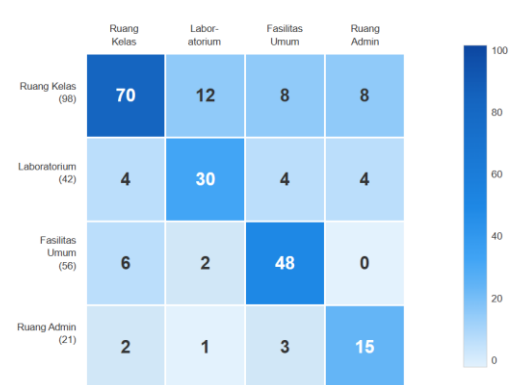
Dengan tingkat kepercayaan 95% serta nilai *t-Student* pada Gambar 17 yaitu $t_{0.025,19} = 2.093$, diperoleh *margin of error* sebesar ± 2.20 , menghasilkan interval estimasi antara 80.8 hingga 85.2. Selain itu, mean skor SUS sebesar 82.87 berada pada Grade A, yang termasuk kategori *Excellent* dan dapat diterima (*Acceptable*) oleh penggunaanya.

3.4 Evaluasi Kinerja NLP

Evaluasi kinerja sistem NLP dilakukan untuk mengukur efektivitas model dalam mengidentifikasi dan mengklasifikasikan entitas fasilitas kampus dari input suara pengguna. Evaluasi dilakukan menggunakan *multiclass classification*, setiap *utterance* diklasifikasikan ke dalam salah satu dari 4 kategori fasilitas. *Ground truth* dalam evaluasi ini adalah jumlah *utterance* aktual untuk setiap kategori fasilitas. Pendekatan yang diimplementasikan merupakan metode *hybrid keyword-based*, integrasi antara teknik *keyword matching* yang diperkuat dengan serangkaian proses *preprocessing* (*normalization, tokenization, synonym mapping, abbreviation expansion*) serta komponen NER berbasis *pattern matching, POS tagging, dictionary, dan typo correction* melalui *fuzzy matching*. Untuk memvalidasi pendekatan *hybrid keyword-based*, dilakukan perbandingan dengan metode *baseline* yaitu *substring matching* sederhana yang hanya mencocokkan kata kunci tanpa *preprocessing*. Gambar 18 menunjukkan *confusion metrik* untuk *hybrid keyword-based* dan Gambar 19 menunjukkan *confusion metrik* untuk *substring matching*. Serta diperoleh Tabel 16 dan Tabel 17 menyajikan metrik evaluasi untuk kedua metode.



Gambar 18. Hybrid Keyword-Based



Gambar 19. Substring matching

Tabel 16. Hasil Evaluasi Hybrid Keyword-Based

Kategori Fasilitas	#Utterance	TP	FP	FN	Precision	Recall	F1
Ruang Kelas	98	92	2	6	0.9787	0.9388	0.9583

Laboratorium	42	38	1	4	0.9744	0.9048	0.9383
Fasilitas Umum	56	53	5	3	0.9138	0.9464	0.9298
Ruang Administrasi	21	18	6	3	0.75	0.8571	0.8000
<i>Weighted Average</i>	217	201	14	16	0.9348	0.9263	0.9297

Tabel 17. Hasil Evaluasi *Substring Matching*

Kategori Fasilitas	#Utterance	TP	FP	FN	Precision	Recall	F1
Ruang Kelas	98	70	12	28	0.8537	0.7143	0.7778
Laboratorium	42	30	15	12	0.6667	0.7143	0.6897
Fasilitas Umum	56	48	8	8	0.8571	0.8571	0.8571
Ruang Administrasi	21	15	12	6	0.5556	0.7143	0.625
<i>Weighted Avg.</i>	217	163	47	54	0.7879	0.7512	0.7655

Tabel 18. Perbandingan Performa

Kategori	Hybrid Keyword-based			Substring Matching			
	Prec	Rec	F1	Prec	Rec	F1	Δ F1
Ruang Kelas	0.9787	0.9388	0.9583	0.8537	0.7143	0.7778	0.1803
Laboratorium	0.9744	0.9048	0.9383	0.6667	0.7143	0.6897	0.2486
Fasilitas Umum	0.9138	0.9464	0.9298	0.8571	0.8571	0.8571	0.0727
Ruang Administrasi	0.75	0.8571	0.8	0.5556	0.7143	0.625	0.175
<i>Weighted Average</i>	0.9346	0.9263	0.9297	0.7879	0.7512	0.7655	0.1642

Evaluasi terhadap 217 *utterance* menunjukkan bahwa metode *Hybrid Keyword-Based* mencapai F1-Score 0.9297 yang masuk dalam kategori interpretasi *very good*[33], sedangkan *Substring Matching* memperoleh 0.7655 dengan interpretasi OK, dengan selisih 0.1642 poin dan reduksi total error sebesar 0.703. Pada metode *Hybrid Keyword-Based* menunjukkan *False Negative* sebanyak 16 kasus yang umumnya disebabkan oleh *query* ambigu tanpa parameter spesifik. Selain itu, *False Positive* sebanyak 14 kasus terjadi akibat kesalahan klasifikasi Entitas. Metode *Substring Matching* menunjukkan tingkat *False Negative* yang lebih tinggi (54 kasus) akibat ketidakmampuan menangani variasi *utterance* serta *False Positive* berlebih (47 kasus) yang dipicu oleh *partial word matching* dengan variasi ejaan atau tanpa validasi *word boundary*.

Contoh 1. Kasus FN *hybrid keyword-based* : Dimana ruang bimbingan? *query* ruang bimbingan ada *didatabase* namun sistem memberikan *output* berupa fasilitas tidak ditemukan, karena sistem hanya mengenali jika diberikan pencarian yang spesifik contohnya “Dimana ruang bimbingan ilkom”, untuk perbaikan seharusnya sistem mengembalikan *follow-up question* seperti “ruang bimbingan fakultas apa yang ingin dicari”, maka sistem akan mengenali Entitas spesifik yang diucapkan pengguna.

Contoh 2. Kasus FP *hybrid keyword-based* : Dimana admin prodi SI? keliru dipetakan ke admin prodi TI, menandakan keterbatasan sistem dalam membedakan identitas entitas spesifik dalam kategori yang sama, diperlukan *hierarchical entity recognition system* yang dapat membedakan level spesifisitas entitas untuk memvalidasi apakah *query* merupakan *location query* yang valid.

4. KESIMPULAN

Penelitian ini berhasil mengimplementasikan aplikasi pencarian fasilitas kampus berbasis suara dan NLP menggunakan metode *Hybrid Keyword-Based* yang mengintegrasikan *keyword matching*, *preprocessing*, serta komponen *Named Entity Recognition* berbasis *pattern matching*, *POS tagging*, *dictionary*, dan *fuzzy matching*. Pengujian *blackbox* terhadap 13 fitur menunjukkan seluruh fungsi berjalan sesuai spesifikasi, dan evaluasi *usability* menggunakan *System Usability Scale* pada 20 responden menghasilkan skor 82.87 dengan Grade A kategori *Excellent*, sehingga dapat diterima (*acceptable*) oleh penggunaanya dengan *confidence interval* 95% pada rentang 80.8-85.2. Evaluasi kinerja NLP terhadap 217 *utterance* menunjukkan metode *Hybrid Keyword-Based* mencapai F1-Score

0.9297 kategori *very good* meskipun masih terdapat 16 kasus *False Negative* akibat *query* ambigu dan 14 kasus *False Positive* akibat *misclassification* entitas. Sebagai saran untuk penelitian selanjutnya, implementasi *follow-up question* untuk menangani *query* ambigu dengan mengajukan pertanyaan lanjutan, sedangkan *hierarchical entity recognition system* disarankan untuk membedakan tingkat spesifisitas entitas dan memvalidasi *query*.

DAFTAR PUSTAKA

- [1] F. Rumaisa, Y. Puspitarani, A. Rosita, A. Zakiah, and S. Violina, "Penerapan Natural Language Processing (NLP) di bidang pendidikan," *Jurnal Inovasi Masyarakat*, vol. 1, no. 3, pp. 232–235, 2021, doi: 10.33197/jim.vol1.iss3.2021.799.
- [2] A. K. Badriello, T.A Yiswi, "Dampak Artificial Intelligence (AI) Terhadap Pendidikan di Indonesia," *Jurnal Pendidikan Indonesia*, vol. 6, no. 7, pp. 3112–3121, 2025, doi: 10.59141/japendi.v6i7.8433.
- [3] K. V Kulhalli, K. Sirbi, and M. A. J. Patankar, "Personal Assistant with Voice Recognition Intelligence," *International Journal of Engineering Research and Technology*. Accessed: Nov. 23, 2025. [Online]. Available: <https://share.google/VhRZN5GjZ5dTbZcAj>
- [4] M. Jafar, A. M. I. T. Asfar, A. M. I. Akbar, and Asfar, "Artificial Intellegence dalam Pendidikan dan Penelitian: Tantangan dan Solusi Menghadapinya," *Simposium Nasional*, vol. 1, no. 1, pp. 1–9, 2024, doi: <https://doi.org/10.15294/snkpti.v1i1.3900>.
- [5] A. Alfarizi, J. Alie, and T. Yustini, "Pengaruh Brand Image, Lokasi Kampus dan Dukungan Fasilitas Terhadap Keputusan Mahasiswa Dalam Memilih Perguruan Tinggi Swasta (PTS) di Kota Palembang," *Jurnal Manajemen Bisnis Kewirausahaan*, vol. 1, no. 2, pp. 31–44, 2022, doi: 10.56910/jumbiwira.v1i2.60.
- [6] Mappedin, "Menerapkan aplikasi navigasi kampus yang sukses," Mappedin.com. Accessed: Nov. 22, 2025. [Online]. Available: <https://www.mappedin.com/resources/blog/implementing-a-successful-campus-navigation-app/>
- [7] I. Huda, "Implementasi Natural Language Processing (NLP) untuk Aplikasi Pencarian Lokasi," *Jurnal Nasional Teknologi Terapan*, vol. 3, no. 2, p. 15, 2021, doi: 10.22146/jntt.35036.
- [8] A. Bhalerao, S. Bhilare, A. Bondade, and M. Shingade, "Smart Voice Assistant : a universal voice control solution for non-visual access to the Android operating system," *International Research Journal of Engineering and Technology*. Accessed: Nov. 25, 2025. [Online]. Available: <https://share.google/eEEkfPdbhCo5Ra1Rg>
- [9] Y. Fatman and I. Islamiyati, "Pengenalan Suku Kata Bahasa Indonesia Menggunakan Metode LPC Dan Backpropagation Neural Network," *Journal of Infomation Technology Computer Science*, vol. 5, no. 3, p. 155, 2020, doi: 10.31328/jointecs.v5i3.1331.
- [10] A. Z. Ezra William, "Speech recognition dengan Whisper dalam bahasa Indonesia," *Action Research Literate*, vol. 9, no. 2, pp. 386–397, 2025, doi: doi.org/10.46799/ar.v9i2.2573.
- [11] A. Arrizal, "Named Entity Recognition (NER) pada Teks Berbahasa Indonesia dengan Fine-Tuning IndoBERT," unsri. Accessed: Nov. 23, 2025. [Online]. Available: https://repository.unsri.ac.id/161994/3/RAMA_55201_09021182126027_0001108401_01_from_t_ref.pdf
- [12] Y. D. Putra, Y. P. Putra, and P. S. Saputra, "Analisa Pencarian Dan Pencocokan String Dalam Aplikasi Berbasis Python Dengan Library Fuzzy Wuzzy Terhadap Dataset CNN Daily Mail," *Jurnal Komputer dan Teknologi Sains*, vol. 4, no. 2, pp. 7–13, 2025, doi: <https://doi.org/10.37637/komteks.v4i2.2354>.
- [13] N. Afifa, and R. E. Saputra, "Implementasi NLP Pada Chatbot Layanan Akademik Dengan Algoritma Bert," *eProceedings of Engineering*. Accessed: Apr. 23, 2025. [Online]. Available: <https://openlibrarypublications.telkomuniversity.ac.id/index.php/engineering/article/view/19542>
- [14] Noviyanti and Sholihin, "Implementasi Natural Language Processing (NLP) Untuk Personal Asistant Berbasis Voice Commad," *Jurnal Riset Informatika dan Inovasi*. Accessed: Nov. 24, 2025. [Online]. Available: <https://www.jurnalmahasiswa.com/index.php/jriin/article/view/347>

- [15] Wikipedia, "Universitas Indo Global Mandiri," *Universitas IGM*. Accessed: Apr. 12, 2025. [Online]. Available: https://id.wikipedia.org/wiki/Universitas_Indo_Global_Mandiri#Referensi
- [16] N. Agung Cipto Roso, A. Herdiansah, and T. Handayani, N. Fitrawati, "Penerapan Metode Rapid Application Development (Rad) Pengembangan Sistem Informasi Manajemen Barang Reject," *Jurnal Informatika*, vol. 8, no. 3, p. 326, 2024, doi: 10.31000/jika.v8i3.11841.
- [17] I. Permatasari, P. M. Hutabarat, and E. Purnamasari, "Faktor Evaluasi Usabilitas dalam Sistem e-learning dengan Panduan Tinjauan Sistematis PRISMA," *Journal of Engineering and Sustainable Technology*, vol. 9, no. 02, pp. 805–810, 2023, doi: 10.31949/jensitec.v9i02.3662.
- [18] Yuniarni, and Y. B. Pratama, "Pengembangan Virtual Assistant menggunakan Teknologi NLP dengan Metode Algoritma Machine Learning untuk Layanan Informasi Akademik di SMA Negeri 1 Parittiga Berbasis Web," *Jurnal Teknik Mesin, Industri, Elektro dan Ilmu Komputer*, vol. 3, no. 5, pp. 205–219, 2025, doi: doi.org/10.61132/mars.v3i5.1152.
- [19] M. F. R. Khairul and R. S. Perdana, "Arsitektur Sistem Percakapan Otomatis Berbahasa Indonesia dengan Normalisasi Bahasa Informal Menjadi Baku," *Jurnal Teknologi Informasi dan Ilmu Komputer*, vol. 11, no. 5, pp. 1009–1016, 2024, doi: 10.25126/jtiik.2024117984.
- [20] I. Musyaffa, "Metode Pengembangan Rapid Application Development," Fit Tech Inova Global. Accessed: Apr. 21, 2025. [Online]. Available: <https://www.fittechnova.com/blog/detail/metode-pengembangan-rad-rapid-application-development>
- [21] P. R. Setiawan, M. Asfi, A. Seviana, S. Pranata, and W. E. Septian, "Design System pada Perancangan Antar Muka Perangkat Lunak Sistem Akses Digital," *Jurnal Teknologi Terpadu*, vol. 9, no. 1, pp. 56–64, 2023, doi: doi.org/10.54914/jtt.v9i1.619.
- [22] C. Annisa Fitri, S. Hidayat Al Ikhsan, and F. Agus Setiawan, "Aplikasi Augmented Reality Untuk Pengenalan Fasilitas Umum Universitas Ibn Khaldun Bogor," *Jurnal Mahasiswa Teknik Informatika*, vol. 8, no. 3, pp. 2856–2862, 2024, doi: 10.36040/jati.v8i3.9558.
- [23] I. A. Fathony, A. Mareta, and B. Adiana, "Optimisasi Whisper Speech-to-Text Bahasa Indonesia dengan Hybrid Cloud dan Multi-Engine," *Jurnal Elektronika dan Komputer*, vol. 18, no. 1, pp. 60–72, 2025, doi: <https://doi.org/10.51903/5n2d3s08>.
- [24] M. D. Etsa, H. Sujaini, and N. Safriadi, "Pengaruh Metode Dictionary Lookup pada Cleaning Korpus Terhadap Akurasi Mesin Penerjemah Statistik Indonesia–Melayu Pontianak," *Jurnal Edukasi dan Penelitian Inform*, vol. 4, no. 1, p. 49, 2018, doi: 10.26418/jp.v4i1.24595.
- [25] A. F. Hidayatullah, A. P. Wibowo, and K. R. Nastiti, "Named Entity Recognition on Tourist Destinations Reviews in the Indonesian Language," *Jurnal Linguistik Komputasional*, vol. 6, no. 1, pp. 30–35, 2023, doi: 10.26418/jlk.v6i1.89.
- [26] S. Puspasari, and R. Gustriansyah, "Measuring Perceived Usability Of Artificial Intelligence-Based Quizzes In A Virtual Museum," *Jurnal Ilmu Pengetahuan dan Teknologi Komputer*, vol. 10, no. 4, pp. 953–960, 2025, doi: 10.33480/jitk.v10i4.6298.MEASURING.
- [27] Z. Sharfina, and H. B. Santoso "An Indonesian Adaptation of the System Usability Scale (SUS)," *Journal of Physics: Conference Series*, vol. 1566, no. 1, 2016, doi: 10.1088/1742-6596/1566/1/012051.
- [28] S. A. Niwitpong, W. Khumpasee, and S. Niwitpong, "Confidence intervals for the variance of Delta-Inverse Gaussian distribution with application to traffic mortality count," *Symmetry*, vol. 3042, no. 1, 2025, doi: 10.3390/sym17030387.
- [29] R. Gustriansyah, N. Suhandi, S. Puspasari, and A. Sanmorino, "Machine Learning Method to Predict the Toddlers' Nutritional Status," *Journal of Informatics, telecommunication, and electronics*, vol. 16, no. 1, pp. 32–43, 2024, doi: 10.20895/infotel.v15i4.988.
- [30] P. Lawson, "How to SUS out usability scores," The Think Blog. Accessed: Jun. 16, 2025. [Online]. Available: <https://www.thinkcompany.com/blog/how-to-sus-out-usability-scores/>
- [31] P. K. Rahman, "Analisis Tingkat Efektivitas Sistem Informasi Akademik IAIN Syaikh Abdurrahman Siddik Bangka Belitung," *Jurnal Kependidikan dan Sosial Keagamaan*, vol. 7, no. 1, pp. 1–17, 2021, doi: 10.32923/edugama.v7i1.1805.
- [32] Yuvalianda, "Distribusi-t: Solusi Pengujian Sampel Kecil!," Blog Yuva. Accessed: Nov. 22, 2025. [Online]. Available: <https://yuvalianda.com/distribusi-t/>
- [33] N. Buhl, "Skor F1 dalam Pembelajaran Mesin," Encord. Accessed: Nov. 30, 2025. [Online]. Available: <https://encord.com/blog/f1-score-in-machine-learning>