

PENGEMBANGAN “LOOPA” ASISTEN PENGINGAT JADWAL PINTAR BERBASIS AI

Rizal Muslim Sinaga, Muhammad Alby Savana HSB, Nazwar Farezi,

Fauzan Azima Lubis, Adidtya Perdana

Ilmu Komputer, Universitas Negeri Medan

Jalan Willem Iskandar, Pasar V Medan Estate, Percut Sei Tuan, Deli Serdang, Indonesia

sinaga.rizal456@gmail.com

ABSTRAK

Di era digital saat ini, manajemen waktu menjadi tantangan bagi banyak individu, terutama ketika sistem pengingat yang digunakan tidak mampu memahami bahasa alami pengguna. Masalah muncul ketika pengguna masih mengandalkan metode manual atau aplikasi kalender konvensional yang kurang efisien dalam menangkap kebutuhan berbasis teks alami. Penelitian ini bertujuan untuk mengembangkan Loopa, sebuah aplikasi web pintar berbasis AI yang dapat mengelola jadwal secara otomatis melalui pemahaman bahasa alami. Aplikasi ini dirancang menggunakan arsitektur client-server dengan Flask sebagai backend dan SQLite sebagai basis data, serta mengintegrasikan model NLP google/flan-t5-small yang telah di-fine-tune untuk memahami input jadwal berbentuk teks. Sistem ini juga dilengkapi fitur notifikasi otomatis menggunakan Twilio. Evaluasi model menunjukkan akurasi sebesar 62%, yang berarti sebagian besar input berhasil dipahami dengan benar, dan ROUGE-L sebesar 0,9596 yang menandakan tingkat kemiripan yang sangat tinggi antara hasil prediksi dan referensi. Hal ini menunjukkan bahwa Loopa mampu memahami konteks input dengan baik dan mengubahnya menjadi format terstruktur secara efektif. Aplikasi Loopa berhasil menunjukkan performa yang baik dalam mengubah input alami menjadi format terstruktur serta mengirimkan pengingat otomatis, sehingga dapat membantu pengguna dalam mengelola waktu secara lebih efisien dan cerdas.

Kata kunci : Manajemen jadwal, Natural Language Processing, AI, notifikasi otomatis, Flask.

1. PENDAHULUAN

Di era digital yang semakin berkembang, manajemen waktu menjadi tantangan bagi banyak individu, baik dalam kehidupan pribadi maupun profesional. Banyak orang kesulitan dalam mengingat jadwal penting seperti rapat, tugas, atau acara tertentu, terutama ketika jadwal yang padat membuat mereka lupa atau melewatkan hal-hal penting. Kurangnya sistem yang fleksibel dalam mengatur jadwal menyebabkan banyak pengguna masih mengandalkan catatan manual atau aplikasi kalender yang memerlukan input eksplisit. Hal ini sering kali tidak cukup efektif dalam membantu pengguna mengelola waktu mereka secara efisien.

Sistem pengingat tradisional, seperti kalender manual atau aplikasi standar, memiliki keterbatasan dalam memahami bahasa alami pengguna. Dengan perkembangan teknologi kecerdasan buatan (Artificial Intelligence/AI) dan pemrosesan bahasa alami (Natural Language Processing/NLP), memungkinkan adanya solusi yang lebih intuitif dan otomatis dalam manajemen jadwal. Teknologi ini memungkinkan sistem untuk memahami input pengguna dalam format bahasa alami, sehingga mempermudah pencatatan dan pengelolaan jadwal tanpa perlu memasukkan informasi secara eksplisit.

Berdasarkan permasalahan tersebut, penelitian ini bertujuan untuk mengembangkan Loopa, sebuah aplikasi web berbasis AI yang dapat membantu pengguna mengelola jadwal dengan lebih efisien. Aplikasi ini memanfaatkan teknologi NLP untuk memahami input jadwal dalam bentuk teks alami,

menyimpannya dalam database SQLite, serta mengirimkan notifikasi otomatis melalui layanan Twilio. Dengan adanya sistem yang cerdas dan adaptif, Loopa diharapkan dapat memberikan solusi inovatif yang meningkatkan efektivitas manajemen waktu pengguna dan mengurangi beban kognitif mereka dalam mengatur jadwal harian.

Loopa dikembangkan dengan pendekatan arsitektur *client-server*, di mana backend dibangun menggunakan Flask untuk menangani pemrosesan data dan autentikasi pengguna berbasis Flask-JWT, sementara frontend dirancang menggunakan framework berbasis web agar memberikan pengalaman pengguna yang optimal. Model NLP yang digunakan adalah *google/flan-t5-small*, yang telah di-fine-tune menggunakan dataset yang mencakup berbagai format input jadwal. Selain sebagai fitur utama dalam NLP, AI juga dimanfaatkan dalam pengembangan perangkat lunak untuk membantu optimasi kode, debugging, serta meningkatkan efisiensi pengembangan sistem.

Penelitian ini bertujuan untuk mengembangkan aplikasi web berbasis AI yang mampu memahami dan mengelola jadwal pengguna secara otomatis. Salah satu fokus utama adalah penerapan NLP untuk memungkinkan pengguna memasukkan jadwal dalam bahasa alami. Selain itu, penelitian ini juga mencakup pengembangan backend dengan Flask dan SQLite sebagai sistem basis data untuk menyimpan serta mengelola jadwal pengguna. Aplikasi ini juga dilengkapi dengan fitur notifikasi otomatis menggunakan Twilio, yang mengingatkan pengguna

terhadap jadwal yang telah mereka buat. Dengan pemanfaatan AI dalam pengembangan dan optimasi sistem, Loopa diharapkan dapat memberikan solusi yang lebih efisien dan efektif dalam membantu pengguna mengatur waktu mereka.

2. TINJAUAN PUSTAKA

2.1. Rekayasa Perangkat Lunak

Istilah Rekayasa Perangkat Lunak (RPL) secara umum disepakati sebagai terjemahan dari istilah Software engineering. Istilah Software Engineering mulai dipopulerkan pada tahun 1968 pada software engineering Conference yang diselenggarakan oleh NATO. Sebagian orang mengartikan RPL hanya sebatas pada bagaimana membuat program komputer. Padahal ada perbedaan yang mendasar antara perangkat lunak (*software*) dan program komputer. RPL adalah disiplin ilmu tersendiri yang membahas semua aspek produksi perangkat lunak, dimulai dengan tahap awal, seperti menganalisis kebutuhan pengguna, menentukan spesifikasi kebutuhan pengguna, desain, pengkodean, pengujian, dan pemeliharaan sistem setelah digunakan.[1]

Rekayasa Perangkat Lunak (RPL) adalah suatu disiplin ilmu yang mengulas semua aspek produksi perangkat lunak, dimulai dari tahap awal yaitu communication, requirements capturing (analisa kebutuhan pengguna), specification (menentukan spesifikasi dari kebutuhan pengguna), desain, coding, testing hingga maintenance (pemeliharaan sistem) setelah digunakan[2].

2.2. Artificial intelligent (AI)

Kecerdasan buatan atau AI merupakan cabang ilmu komputer yang memungkinkan mesin, seperti komputer, melakukan tugas sebagaimana yang dilakukan oleh manusia. Pada awalnya, komputer hanya berperan sebagai alat hitung. Namun, seiring berjalannya waktu, peran komputer semakin memasuki berbagai aspek kehidupan manusia. Komputer tidak hanya sekadar alat hitung, tetapi diharapkan dapat dikembangkan untuk menyelesaikan berbagai tugas yang dapat dilakukan oleh manusia[3].

Penerapan Artificial Intelligence (AI) dapat melibatkan berbagai teknologi seperti mesin pembelajaran (machine learning), chatbot, augmented reality (AR), virtual reality (VR), dan berbagai teknologi lainnya. Tujuan utama AI adalah memungkinkan komputer untuk menjalankan perintah dengan kemampuan yang dapat menyaingi kemampuan manusia[4].

Perkembangan sejarah kecerdasan buatan dimulai pada tahun 1950-an ketika ilmuwan seperti Alan Turing dan John McCarthy memperkenalkan gagasan tentang mesin cerdas dan kecerdasan buatan. Seiring berjalannya waktu, terjadi kemajuan signifikan dalam komputasi dan algoritma, yang memungkinkan pengembangan sistem AI yang lebih canggih dan efektif[5].

2.3. Natural Language Processing (NLP)

Kecerdasan buatan atau AI merupakan cabang ilmu komputer yang memungkinkan mesin, seperti komputer, melakukan tugas sebagaimana yang dilakukan oleh manusia. Pada awalnya, komputer hanya berperan sebagai alat hitung. Namun, seiring berjalannya waktu, peran komputer semakin memasuki berbagai aspek kehidupan manusia. Komputer tidak hanya sekadar alat hitung, tetapi diharapkan dapat dikembangkan untuk menyelesaikan berbagai tugas yang dapat dilakukan oleh manusia[6].

Natural Language Processing (NLP), sebuah subbidang kecerdasan buatan, didedikasikan untuk mengeksplorasi hubungan rumit antara komputer dan bahasa manusia. Tujuan utamanya terletak pada memahami, memeriksa, dan menghasilkan informasi tekstual dengan cara yang meniru kemampuan manusia[7].

2.4. Sistem Peningkat

Reminder atau Sistem Peningkat adalah sebuah sistem pesan yang membantu seseorang untuk mengingatkan sesuatu hal. Sistem Peningkat dapat lebih bermanfaat ketika informasi kontekstual digunakan untuk menyajikan informasi pada waktu yang tepat dan tempat yang tepat. Sistem Peningkat dapat digunakan dalam manajemen waktu yang berfungsi untuk memberi alarm peringatan berupa pemberitahuan berbasis lokasi, waktu maupun catatan yang bersifat kontekstual.[8]

Sistem peringatan digital digunakan untuk melayani orang-orang sebagai sebuah alat untuk membantu manusia mengingat informasi, atau melakukan suatu tindakan yang jika tidak dilupakan". Sistem peringatan digital membantu manusia dalam 2 area memorinya, yakni: prospective memory dan retrospective memory[9].

2.5. Penelitian Terdahulu

Penelitian sebelumnya yang dilakukan oleh Mutia Atika dan Retno Sayekti dalam jurnal Palimpsest mengkaji penggunaan kecerdasan buatan (AI) dalam sistem informasi perpustakaan, dengan penekanan pada layanan referensi, pencarian literatur, dan keamanan perpustakaan. Penelitian ini menunjukkan bahwa AI dapat meningkatkan efektivitas layanan perpustakaan dengan mengotomatiskan berbagai proses, seperti penerapan Natural Language Processing (NLP) dalam layanan rujukan, pengenalan wajah untuk keamanan, serta penggunaan robot dalam pencarian dan penyusunan koleksi buku. Selain itu, studi ini menyoroti pentingnya kerja sama antara perpustakaan dan ahli AI untuk mengembangkan layanan yang lebih inovatif dan memenuhi kebutuhan pengguna[10].

Sementara itu, pengembangan Loopa: Asisten Jadwal Peningkat Pintar Berbasis AI memiliki perhatian yang berbeda, yaitu dalam mengelola dan mengingat jadwal pengguna dengan cerdas. Loopa menggunakan AI dan NLP untuk menangkap perintah

pengguna serta mengotomatiskan pengelolaan jadwal sesuai dengan kebiasaan dan preferensi masing-masing. Berbeda dari sistem AI di perpustakaan yang fokus pada pencarian dan pengelolaan data, Loopa lebih menonjolkan personalisasi dan otomatisasi untuk membantu pengguna mengelola waktu mereka dengan lebih efisien.

Dari analisis ini, dapat disimpulkan bahwa studi sebelumnya lebih menekankan pada peningkatan layanan perpustakaan menggunakan AI, sementara Loopa menargetkan optimalisasi manajemen jadwal individu dengan pendekatan yang lebih personal dan adaptif. Walaupun keduanya memanfaatkan AI dan NLP sebagai teknologi inti, tujuan akhir mereka berbeda—AI dalam perpustakaan bertujuan untuk mempercepat akses informasi, sedangkan Loopa berfokus pada peningkatan produktivitas pengguna lewat pengelolaan jadwal yang lebih efisien.

3. METODE PENELITIAN

Metode penelitian yang digunakan dalam pengembangan Loopa adalah metode rekayasa perangkat lunak dengan pendekatan berbasis web. Tahapan penelitian meliputi analisis kebutuhan, perancangan sistem, implementasi, pengujian, dan dokumentasi. Loopa dikembangkan menggunakan arsitektur client-server dengan backend berbasis Flask dan database SQLite. Model NLP *google/flan-t5-small* digunakan untuk memahami input jadwal, sedangkan sistem notifikasi diimplementasikan dengan Twilio. Pengujian dilakukan untuk memastikan bahwa fitur utama dalam sistem Loopa AI berjalan dengan baik melalui pendekatan end-to-end, serta evaluasi model NLP dilakukan berdasarkan hasil inferensi tanpa melibatkan umpan balik pengguna.

3.1. Analisis Kebutuhan

Analisis kebutuhan dilakukan untuk menentukan fitur utama yang harus dimiliki oleh Loopa agar dapat membantu pengguna dalam mengelola jadwal secara otomatis. Kebutuhan sistem dibagi menjadi dua kategori, yaitu kebutuhan fungsional dan non-fungsional. Kebutuhan fungsional mencakup kemampuan sistem untuk memahami input jadwal dalam bahasa alami, menyimpan data ke dalam database, serta mengirimkan notifikasi otomatis melalui Twilio. Sementara itu, kebutuhan non-fungsional meliputi keamanan sistem dengan autentikasi pengguna berbasis Flask-JWT, skalabilitas arsitektur berbasis client-server, serta efisiensi dalam pemrosesan teks menggunakan model NLP *google/flan-t5-small*. Analisis kebutuhan ini menjadi dasar dalam perancangan sistem agar aplikasi dapat berfungsi sesuai dengan tujuan penelitian.

3.2. Perancangan Sistem

Pengembangan Loopa diawali dengan analisis kebutuhan dan perancangan arsitektur sistem yang bertujuan untuk memastikan sistem dapat memahami input teks alami, menyimpannya dalam database, serta

mengirimkan notifikasi secara otomatis kepada pengguna. Arsitektur sistem menerapkan model client-server, di mana backend dikembangkan menggunakan Flask sebagai server utama dan frontend menggunakan framework berbasis web untuk menampilkan antarmuka kepada pengguna. Backend bertanggung jawab untuk memproses input pengguna, menyimpan data ke dalam database SQLite, serta menangani autentikasi

3.3. Desain Antarmuka Pengguna

Antarmuka pengguna dirancang agar sederhana dan intuitif sehingga memudahkan pengguna dalam mengelola jadwal. Struktur antarmuka terdiri dari beberapa halaman utama, yaitu halaman autentikasi untuk proses login dan registrasi, dashboard untuk menampilkan ringkasan jadwal, halaman manajemen jadwal yang memungkinkan pengguna menambah dan mengedit jadwal, serta halaman pengaturan notifikasi yang memungkinkan konfigurasi pengingat berbasis Twilio. Setiap halaman dirancang agar dapat diakses dengan mudah dan memberikan pengalaman pengguna yang optimal.

3.4. Implementasi NLP untuk Jadwal

Loopa menggunakan Natural Language Processing (NLP) untuk memahami input jadwal dalam bahasa alami. Model yang digunakan adalah *google/flan-t5-small*, yang telah di-*fine-tune* menggunakan dataset yang dirancang secara spesifik agar mampu mengenali berbagai pola input jadwal. Dataset ini digenerate dengan bantuan AI untuk mencakup berbagai variasi format waktu dan peristiwa. Proses implementasi NLP dilakukan menggunakan pustaka *Transformers* dan *PyTorch*, di mana model yang telah di-*fine-tune* akan digunakan untuk memproses input pengguna dan menghasilkan output dalam format yang dapat langsung disimpan ke database.

3.5. Pengembangan Backend dan Database

Backend dikembangkan menggunakan Flask dengan berbagai modul pendukung, seperti Flask-JWT untuk autentikasi pengguna dan SQLAlchemy untuk manajemen database. Database yang digunakan adalah SQLite, yang menyimpan informasi jadwal serta data pengguna. Backend menyediakan beberapa endpoint API untuk menangani operasi CRUD (Create, Read, Update, Delete) pada jadwal dan informasi pengguna. Setiap interaksi pengguna dengan sistem akan diproses melalui backend sebelum disimpan dalam database atau dikirim sebagai notifikasi.

3.6. Implementasi Notifikasi dengan Twilio

Sistem notifikasi diimplementasikan menggunakan layanan Twilio, yang memungkinkan pengiriman pesan pengingat kepada pengguna berdasarkan jadwal yang telah dibuat. Aplikasi memiliki mekanisme *scheduler* yang secara otomatis menjalankan proses pengecekan jadwal dan

mengirimkan notifikasi sesuai dengan waktu yang telah ditentukan. Dengan adanya fitur ini, pengguna dapat menerima pengingat secara real-time tanpa harus membuka aplikasi secara langsung.

3.7. Pemanfaat AI dalam Pengembangan Kode

Selain sebagai fitur utama dalam NLP, AI juga dimanfaatkan dalam pengembangan perangkat lunak untuk meningkatkan efisiensi dan mempercepat proses pengkodean. AI digunakan untuk membantu menghasilkan skrip awal dalam pengembangan backend dan API, memberikan rekomendasi optimasi algoritma, serta membantu dalam debugging untuk mengidentifikasi dan memperbaiki kesalahan dalam sistem. Pemanfaatan AI dalam pengembangan ini memungkinkan proses coding yang lebih cepat dan lebih efisien, sehingga mempercepat waktu pengembangan aplikasi.

3.8. Pengujian dan Evaluasi

Pengujian dilakukan untuk memastikan bahwa fitur utama dalam sistem Loopa AI berjalan dengan baik melalui pendekatan end-to-end, mulai dari input teks pengguna, pemrosesan oleh model NLP, konfirmasi hasil, penyimpanan ke jadwal, hingga pengiriman notifikasi WhatsApp dan penampilan data di dashboard admin. Evaluasi terhadap model NLP dilakukan dengan menguji berbagai format input jadwal dan menggunakan metrik seperti Accuracy, F1 Score, BLEU Score, ROUGE-1, ROUGE-2, ROUGE-L, dan Cosine Similarity untuk mengukur kinerja model dalam memahami dan merespons input pengguna secara akurat. Accuracy mengukur persentase prediksi yang benar, F1 Score memperhitungkan keseimbangan antara presisi dan recall, BLEU Score menilai kesamaan antara teks yang dihasilkan dengan referensi, ROUGE mengukur kesamaan berdasarkan n-gram dan urutan kata, sedangkan Cosine Similarity menghitung kemiripan vektor antara teks referensi dan teks hasil keluaran. Namun, evaluasi ini masih bersifat terbatas karena sistem belum diimplementasikan dalam skala produksi dan belum melibatkan umpan balik langsung dari pengguna akhir.

3.9. Dokumentasi Proyek

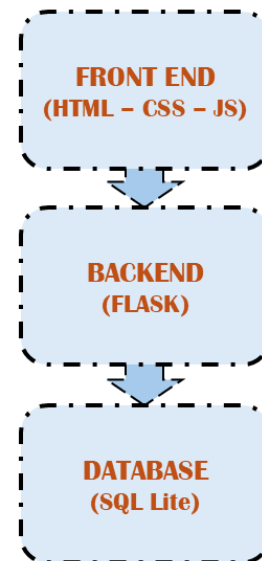
Setelah pengujian selesai, dokumentasi proyek akan disusun untuk menjelaskan cara kerja sistem, struktur kode, serta panduan penggunaan aplikasi. Dokumentasi ini mencakup diagram arsitektur sistem, alur proses NLP, serta integrasi notifikasi dengan Twilio. Dengan adanya dokumentasi yang lengkap, pengembangan di masa depan dapat dilakukan dengan lebih mudah dan sistem dapat diperbarui atau diperbaiki jika diperlukan.

4. HASIL DAN PEMBAHASAN

4.1. Hasil Penelitian

Loopa dikembangkan menggunakan arsitektur client-server, di mana backend berbasis Flask

berfungsi sebagai server utama yang menangani pemrosesan data dan komunikasi dengan database SQLite. Frontend berbasis web berinteraksi dengan backend melalui API REST untuk mengelola jadwal pengguna.

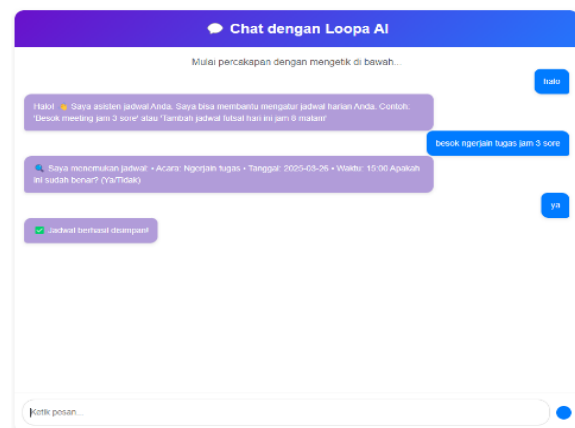


Gambar 1. Diagram Arsitektur Sistem Loopa

Dari gambar di atas diketahui bahwa:

- Sistem menggunakan arsitektur client-server dengan Flask sebagai backend.
- API REST digunakan untuk komunikasi antara frontend dan backend.
- Database SQLite digunakan untuk menyimpan data pengguna dan jadwal.

Model NLP *google/flan-t5-small* digunakan untuk memahami input jadwal dalam bahasa alami. Model ini menerima input teks dari pengguna, memprosesnya menggunakan teknik inferensi berbasis transformator, dan mengonversinya menjadi format terstruktur yang dapat disimpan dalam database. Model ini dikembangkan menggunakan *fine-tuned T5ForConditionalGeneration* dan diintegrasikan dengan Flask untuk menangani permintaan pengguna secara real-time.

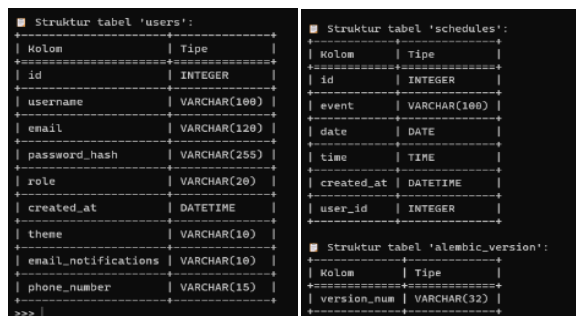


Gambar 2: Demo Loopa AI dalam Menanggapi Permintaan Pengguna

Berikut penjelasan dari Gambar 2:

- Input Pengguna – Pengguna memasukkan perintah dalam bahasa alami untuk menambahkan jadwal ke sistem.
- Interpretasi NLP – Sistem memproses input pengguna dengan mengenali aktivitas, waktu, dan tanggal secara otomatis.
- Konfirmasi Jadwal – Aplikasi meminta konfirmasi dari pengguna sebelum menyimpan jadwal yang telah diinterpretasikan.
- Penyimpanan dan Notifikasi – Setelah pengguna mengonfirmasi, sistem menyimpan jadwal dan memberikan notifikasi bahwa jadwal berhasil disimpan.

SQLite digunakan sebagai database utama karena ringan dan mudah dikelola dalam skala proyek ini, sementara SQLAlchemy digunakan sebagai ORM untuk mempermudah operasi CRUD. Struktur database yang diberi nama `loopa.db` terdiri dari tiga tabel utama: `schedules`, `alembic_version`, dan `users`.



The image shows three database table structures in a terminal-like window. The first table is 'users' with columns: id (INTEGER), username (VARCHAR(100)), email (VARCHAR(120)), password_hash (VARCHAR(255)), role (VARCHAR(20)), created_at (DATETIME), theme (VARCHAR(10)), email_notifications (VARCHAR(10)), and phone_number (VARCHAR(15)). The second table is 'schedules' with columns: id (INTEGER), event (VARCHAR(100)), date (DATE), time (TIME), created_at (DATETIME), and user_id (INTEGER). The third table is 'alembic_version' with columns: version_num (VARCHAR(32)).

Gambar 3: Tabel-Tabel pada Database `loopa.db`

Berikut penjelasan dari Gambar 3:

- Tabel `schedules` menyimpan data jadwal pengguna. Setiap entri berisi `id` sebagai primary key, `event` sebagai deskripsi acara, `date` dan `time` untuk waktu pelaksanaan, `created_at` sebagai timestamp pencatatan, serta `user_id` sebagai foreign key yang menghubungkannya dengan tabel `users`.
- Tabel `alembic_version` digunakan oleh Alembic untuk mencatat versi terakhir migrasi skema database. Tabel ini hanya memiliki satu kolom, yaitu `version_num`, yang menyimpan string versi migrasi agar struktur database tetap konsisten saat diperbarui.
- Tabel `users` menyimpan informasi pengguna, termasuk `id` sebagai primary key, `username`, `email`, dan `password_hash` untuk kredensial. Tabel ini juga mencakup `role`, `created_at`, `theme`, `email_notifications`, dan `phone_number` untuk pengaturan preferensi pengguna.

Evaluasi dilakukan terhadap model NLP menggunakan beberapa metrik seperti Accuracy, F1 Score, BLEU Score, ROUGE, dan Cosine Similarity. Hasil evaluasi ditunjukkan pada Tabel berikut:

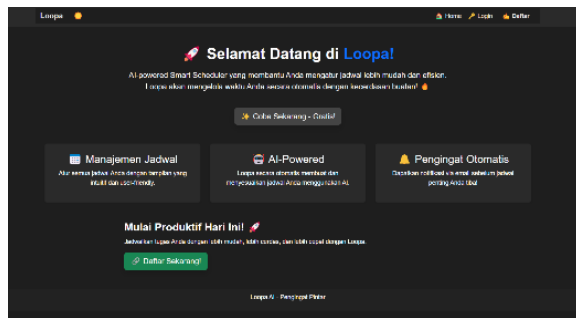
Tabel 1. Metrik Evaluasi Model NLP

Metrik	Nilai
Accuracy	0,62
F1 Score	0,4493
BLEU Score	0,3303
ROUGE-1	0,9596
ROUGE-2	0,9099
ROUGE-L	0,9596
Cosine Similarity	0,9333

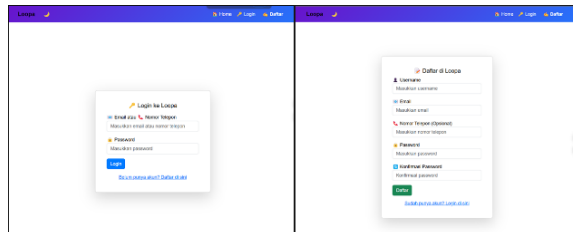
Berikut penjelasan dari tabel 1:

- Accuracy (0.62) – Model memiliki tingkat ketepatan 62%, yang berarti 62% dari prediksi yang dihasilkan sesuai dengan data referensi.
- F1 Score (0.4493) – Metrik ini menunjukkan keseimbangan antara presisi dan recall, dengan skor sekitar 44,93%, yang mengindikasikan masih adanya ruang untuk perbaikan dalam keselarasan prediksi.
- BLEU Score (0.3303) – Skor BLEU yang relatif rendah menunjukkan bahwa teks yang dihasilkan memiliki kesamaan yang terbatas dengan teks referensi, yang bisa disebabkan oleh variasi dalam formulasi kalimat.
- ROUGE-1 (0.9596) – Model memiliki tingkat kemiripan yang tinggi dalam unigram (kata per kata), yang menunjukkan bahwa banyak kata dalam teks referensi muncul dalam prediksi.
- ROUGE-2 (0.9099) – Metrik ini mengukur kesamaan berdasarkan bigram (dua kata berurutan), dengan skor yang tetap tinggi, menandakan bahwa model cukup baik dalam mempertahankan struktur kata.
- ROUGE-L (0.9596) – Skor ini mengindikasikan bahwa model dapat menangkap urutan kata yang panjang dengan baik, mirip dengan referensi.
- Cosine Similarity (0.9333) – Nilai ini menunjukkan bahwa vektor representasi teks yang dihasilkan sangat mirip dengan teks referensi, dengan tingkat kemiripan lebih dari 93%.

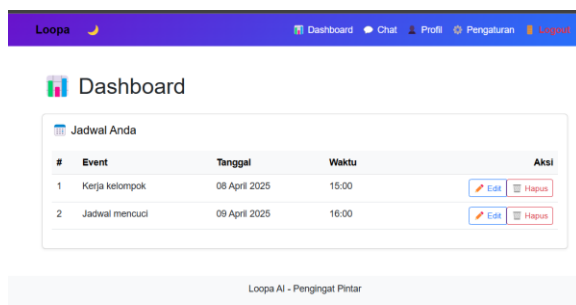
Tampilan web aplikasi ini dirancang untuk memberikan pengalaman pengguna yang intuitif dan fungsional, dimulai dari halaman home dengan dark mode sebagai tampilan default yang dapat diubah langsung melalui tombol toggle di navbar. Pengguna dapat melakukan registrasi atau login untuk mengakses fitur utama, seperti dashboard yang menampilkan ringkasan jadwal, chatbot untuk input kegiatan, serta halaman profil dan pengaturan notifikasi. Seluruh antarmuka dirancang dengan alur navigasi yang konsisten dan responsif, termasuk admin dashboard yang digunakan untuk memantau aktivitas aplikasi secara menyeluruh.



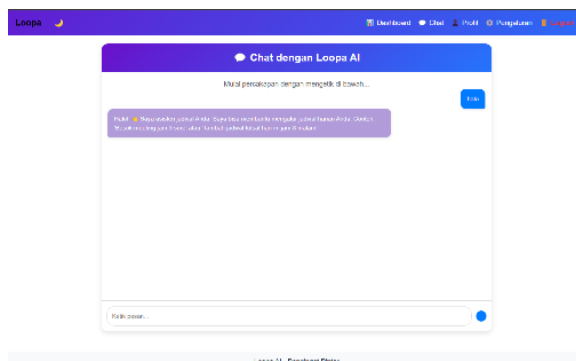
Gambar 4. Tampilan Halaman Home Loopa



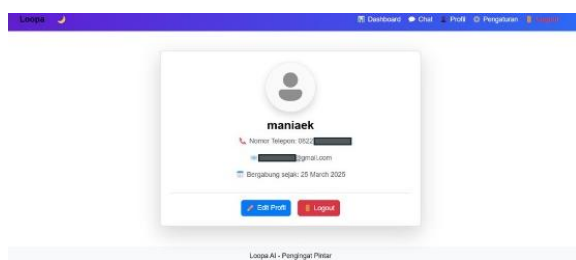
Gambar 5. Tampilan Halaman Login/Register



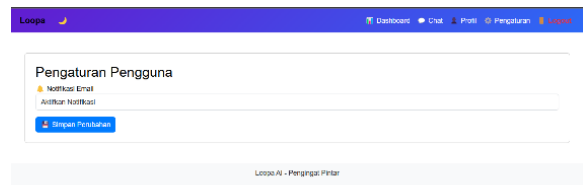
Gambar 6. Tampilan Dashboard Jadwal Pengguna



Gambar 7. Tampilan Chatbot untuk Input Jadwal



Gambar 8. Tampilan Profil Pengguna

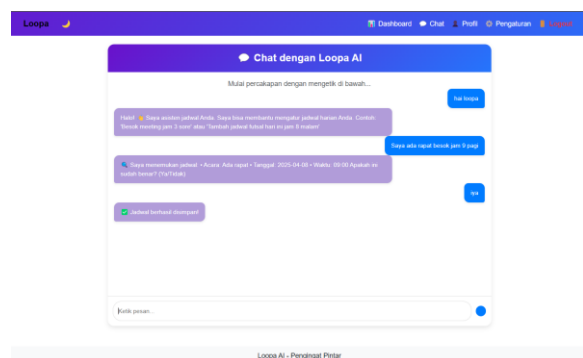


Gambar 9. Tampilan Pengaturan Notifikasi.

Dalam implementasi sistem Loopa, fitur notifikasi otomatis dikembangkan menggunakan Twilio untuk mengirimkan pesan WhatsApp kepada pengguna. Scheduler yang berjalan setiap satu menit bertanggung jawab untuk mengecek apakah ada jadwal yang perlu dikirim sebagai pengingat kepada pengguna. Jika ada jadwal yang sesuai dengan waktu saat ini, sistem akan secara otomatis mengirimkan pesan melalui Twilio API.

4.2. Pengujian Aplikasi

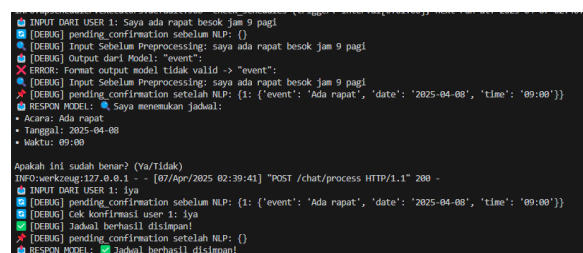
Pengujian dilakukan untuk memastikan bahwa seluruh fitur utama dari aplikasi Loopa AI berjalan sesuai dengan fungsinya. Uji coba dilakukan mulai dari input pesan berbasis teks, pemrosesan NLP hingga konfirmasi dan penyimpanan ke jadwal, notifikasi WhatsApp, serta tampilan dashboard admin.



Gambar 10. Tampilan Input Chat dari Pengguna

Berikut penjelasan dari Gambar 10:

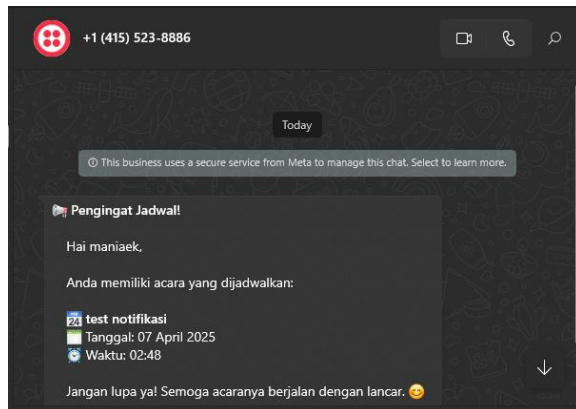
- Pengguna mengetikkan kalimat "Saya ada rapat besok jam 9 pagi" di kolom input aplikasi.
- Kalimat ini dikirimkan ke backend dan diproses menggunakan model NLP yang sudah dilatih.
- Proses ini merupakan awal dari workflow utama sistem dalam mendeteksi jadwal otomatis dari teks.



Gambar 11. Hasil Output Pemrosesan NLP

Berikut penjelasan dari Gambar 11:

- Menunjukkan input alami dari user: "Saya ada rapat besok jam 9 pagi".
- Ada debug NLP: proses preprocessing, output model, dan hasil ekstraksi entitas (event, date, time).
- Terlihat proses konfirmasi user dan penyimpanan ke jadwal berhasil.
- Mewakili seluruh alur pemrosesan NLP dalam proyek: mulai dari input → deteksi jadwal → konfirmasi → penyimpanan.



Gambar 12. Notifikasi WhatsApp Otomatis dari Twilio

Berikut penjelasan dari Gambar 12:

- Pada saat waktu kegiatan mendekati jadwal yang telah ditentukan, sistem akan mengirimkan notifikasi.
- Pesan dikirimkan melalui API Twilio ke nomor WhatsApp pengguna.
- Isi notifikasi mencantumkan nama event dan waktu pelaksanaan, sesuai data jadwal.

Admin Dashboard									
Total Pengguna 3					Total Jadwal 3				
Daftar Pengguna					Jadwal Terbaru				
ID	Username	Email	Role		ID	Event	Tanggal	Waktu	User
1	mariaek	mariaek@gmail.com	user		3	Jadwal meeting	09-04-2025	16:30	mariaek
2	admin	admin@admin.com	admin		2	Kerja kelompok	08-04-2025	15:30	mariaek
3	rsalimaga10	rsalimaga10@gmail.com	user		1	Pengajian tugas	26-03-2025	15:30	rsalimaga10

Gambar 13. Tampilan Data Jadwal dan Pengguna pada Panel Admin

Berikut penjelasan dari Gambar 13:

- Panel ini menampilkan statistik pengguna dan jadwal dalam sistem.
- Admin dapat melihat daftar pengguna lengkap beserta peran mereka (user/admin).
- Terdapat juga daftar jadwal terbaru yang berhasil disimpan oleh pengguna melalui fitur NLP.
- Panel ini mendukung admin untuk mengelola akun pengguna dan mengedit atau menghapus jadwal jika diperlukan.

4.3. Pembahasan

Berdasarkan hasil penelitian dan pengujian yang telah dilakukan, sistem Loopa berhasil diimplementasikan dengan baik menggunakan arsitektur *client-server*. Backend yang dikembangkan menggunakan Flask berperan sebagai pusat logika bisnis, pengelolaan data, serta komunikasi dengan database SQLite. Komunikasi antara frontend dan backend berlangsung melalui API REST secara *real-time*, memungkinkan pengguna untuk mengelola jadwal mereka dengan cepat dan efisien.

Model NLP google/flan-t5-small yang telah *define-tuning* memainkan peran penting dalam memahami input teks dari pengguna. Sebagai contoh, ketika pengguna mengetik "Saya ada rapat besok jam 9 pagi", model berhasil mengidentifikasi nama kegiatan (rapat), waktu (jam 9 pagi), serta tanggal (besok), dan kemudian menghasilkan output JSON terstruktur. Proses ini dilanjutkan dengan konfirmasi kepada pengguna sebelum data tersebut disimpan ke dalam database.

Pengujian terhadap model menghasilkan nilai akurasi sebesar 0,62 dan F1 Score sebesar 0,4493, yang menunjukkan bahwa model cukup mampu mengklasifikasikan input dengan benar meskipun masih ada ketidakseimbangan antara presisi dan recall. Skor BLEU sebesar 0,3303 mengindikasikan bahwa model belum sepenuhnya optimal dalam menyusun struktur kalimat yang sesuai dengan referensi, namun metrik lain seperti ROUGE-1 (0,9596), ROUGE-2 (0,9099), dan ROUGE-L (0,9596) menunjukkan kemiripan kata dan struktur yang sangat tinggi. Selain itu, Cosine Similarity sebesar 0,9333 memperkuat bahwa secara semantik, keluaran model memiliki makna yang sangat dekat dengan referensi. Secara keseluruhan, meskipun ada tantangan dalam penyusunan kalimat secara presisi, model menunjukkan performa yang baik dalam memahami konteks dan menghasilkan respons yang relevan secara makna.

Dari sisi pengujian sistem secara menyeluruh, fitur notifikasi otomatis menggunakan Twilio terbukti berjalan dengan baik. Pesan WhatsApp berhasil dikirim kepada pengguna tepat waktu sesuai dengan jadwal yang telah disimpan. Scheduler yang berjalan setiap satu menit terbukti efektif dalam mengecek jadwal mendatang dan memicu pengiriman notifikasi tanpa keterlambatan. Integrasi antara backend, database, dan modul eksternal seperti Twilio menunjukkan kinerja yang stabil dan dapat diandalkan.

Basis data menggunakan SQLite dan ORM SQLAlchemy memberikan performa yang ringan dan cukup untuk kebutuhan sistem ini. Struktur database yang digunakan terdiri dari tiga tabel utama: *users*, *schedules*, dan *alembic_version*. Selain itu, sistem juga menyediakan halaman admin yang menampilkan statistik pengguna aktif, jumlah jadwal, serta daftar kegiatan yang telah tercatat, menandakan bahwa fitur monitoring dan administrasi sistem berjalan dengan

baik. Dari sudut pandang rekayasa perangkat lunak, Loopa dirancang dengan struktur modular yang jelas. Modularitas ini mendukung pengembangan berkelanjutan, mempercepat proses debugging, dan memberikan fleksibilitas dalam ekspansi sistem ke depan. Validasi peran pengguna (admin dan user) juga diterapkan untuk membatasi akses terhadap fitur-fitur sensitif dan menghindari penyalahgunaan.

Sebagai sistem berbasis AI, Loopa tidak hanya mengimplementasikan model NLP untuk pemrosesan bahasa alami, tetapi juga memanfaatkan teknologi AI dalam pengembangan perangkat lunaknya. Tools seperti GitHub Copilot dan ChatGPT digunakan untuk membantu dalam perancangan kode, penyusunan skenario pengujian otomatis, hingga optimalisasi arsitektur sistem. Pengujian dilakukan menggunakan Pytest, yang menunjukkan hasil pengujian unit terhadap modul-modul penting dalam sistem.

Meskipun sistem telah menunjukkan hasil yang cukup baik, terdapat beberapa area yang masih dapat ditingkatkan. Salah satu tantangan utama adalah ketidakmampuan model NLP dalam menangani variasi input yang ambigu atau tidak umum. Ke depan, pendekatan hybrid antara model berbasis data dan rule-based NLP dapat dipertimbangkan untuk meningkatkan akurasi pemahaman, terutama pada perintah yang tidak standar. Penambahan dataset pelatihan khusus, penerapan active learning dari umpan balik pengguna, serta migrasi ke database yang lebih scalable seperti PostgreSQL dapat mendukung peningkatan performa dan kestabilan sistem dalam skala besar.

Secara keseluruhan, Loopa berhasil mewujudkan sebuah sistem pengingat jadwal cerdas berbasis AI yang dapat memahami perintah pengguna dalam bahasa alami, mengelola jadwal secara real-time, serta memberikan notifikasi otomatis. Dengan dukungan arsitektur modular, sistem keamanan yang memadai, dan integrasi layanan eksternal yang solid, Loopa tidak hanya cerdas dari sisi fungsional, tetapi juga andal dan efisien dari sisi teknis serta pengembangan perangkat lunaknya.

5. KESIMPULAN DAN SARAN

Loopa merupakan aplikasi berbasis kecerdasan buatan yang berhasil diimplementasikan untuk membantu manajemen jadwal melalui pemrosesan bahasa alami menggunakan model NLP google/flan-t5-small yang telah di-*fine-tune*. Sistem ini mampu menginterpretasikan input jadwal dalam bahasa alami, menyimpannya ke dalam database SQLite, dan mengirimkan notifikasi otomatis melalui Twilio secara tepat waktu. Hasil pengujian menunjukkan bahwa sistem bekerja dengan cukup baik dalam mengenali entitas penting seperti waktu dan aktivitas, meskipun masih terdapat keterbatasan dalam menangani variasi kalimat yang ambigu serta belum adanya umpan balik langsung dari pengguna. Oleh karena itu, untuk penelitian selanjutnya disarankan agar dilakukan pengembangan model NLP dengan dataset yang lebih

luas dan representatif, integrasi dengan layanan kalender digital, peningkatan keamanan sistem melalui enkripsi dan autentikasi dua faktor, serta penerapan fitur tambahan seperti manajemen tugas berbasis prioritas dan analisis pola aktivitas pengguna untuk meningkatkan efektivitas dan manfaat aplikasi secara menyeluruh.

DAFTAR PUSTAKA

- [1] R. Habibullah and B. Sugiantoro, "Rekayasa Perangkat Lunak dalam Pendidikan Pesantren," *Nusant. J. Pendidik. Indones.*, vol. 3, no. 1, pp. 83–100, 2023, doi: 10.14421/njpi.2023.v3i1-5.
- [2] S. Tari, F. Idifitriani, and N. Dery Sofya, "Rekayasa Perangkat Lunak Crowdfunding Basiru Menggunakan Pemrograman Php Dan Freamwork Codeigniter," *J. Inform. Teknol. dan Sains*, vol. 4, no. 1, pp. 15–20, 2022, doi: 10.51401/jinteks.v4i1.1339.
- [3] E. Suryokta, W. Taruklimbong, and H. Sihotang, "Peluang dan Tantangan Penggunaan AI (Artificial Intelligence) dalam Pembelajaran Kimia," *J. Pendidik. Tambusai*, vol. 7, no. 3, pp. 26745–26757, 2023.
- [4] Miftahul Huda and Irwansyah Suwahyu, "Peran Artificial Intelligence (Ai) Dalam Pembelajaran Pendidikan Agama Islam," *Ref. Islam. J. Stud. Islam*, vol. 2, no. 2, pp. 53–61, 2024, doi: 10.61220/ri.v2i2.005.
- [5] S. Rifky, "Dampak Penggunaan Artificial Intelligence Bagi Pendidikan Tinggi," *Indones. J. Multidiscip. Soc. Technol.*, vol. 2, no. 1, pp. 37–42, 2024, doi: 10.31004/ijmst.v2i1.287.
- [6] D. O. Sihombing, "Implementasi natural language processing (NLP) dan algoritma cosine similarity dalam penilaian ujian esai otomatis," *Jurnal Sistem Komputer dan Informatika (JSON)*, vol. 4, no. 4, pp. 396–406, 2022, doi: 10.30865/json.v4i2.5374.
- [7] N. Nurwanda, N. Suarna, and W. Prihartono, "Penerapan Nlp (Natural Language Processing) Dalam Analisis Sentimen Pengguna Telegram Di Playstore," *JATI (Jurnal Mhs. Tek. Inform.)*, vol. 8, no. 2, pp. 1841–1846, 2024, doi: 10.36040/jati.v8i2.8469.
- [8] D. A. Subandi Subandi, Basuki Hari Prasetyo, "Pembangunan Sistem Informasi Penjadwalan Reminder Tagihan Airtime Pelanggan VMS Berbasis Web," *J. Bit*, vol. 17, no. 2, pp. 46–52, 2020.
- [9] J. Yefta, "Implementasi Information Reminder System Menggunakan Data Sharing Pada Keluarga Berbasis Web," vol. 13, no. 2, pp. 171–184, 2024.
- [10] M. Atika and R. Sayekti, "Studi Literatur Review Sistem Informasi Perpustakaan Berbasis Artificial Intelligence (AI): Literatur Review," *Journal of Information and Library Science*, vol. 14, no. 1, pp. 39–52, 2023.