

IMPLEMENTASI FITUR *CHATBOT* AI DAN *REAL-TIME NOTIFICATION* PADA SISTEM *HELPDESK* PERGURUAN TINGGI MENGGUNAKAN METODE AGILE

Radhitya Mugi Pradana¹, Jajam Haerul Jaman², Iqbal Maulana³

^{1,2,3}, Universitas Singaperbangsa Karawang; Jl. HS.Ronggo Waluyo, Puseurjaya, Telukjambe Timur, Karawang, Jawa Barat 41361; Telp. (0267) 641177

Keywords:

Agile Kanban; Chatbot; Helpdesk; Notifikasi Real-Time; Otomatisasi Alur Kerja.

Correspondent Email:

2210631170144@student.unsika.ac.id

Abstrak. Layanan *helpdesk* di perguruan tinggi kerap menghadapi tantangan efisiensi operasional, mencakup tingginya volume pelaporan, proses penanganan tiket yang masih dilakukan secara manual, serta minimnya transparansi status keluhan bagi pengguna. Penelitian ini mengusulkan penerapan dua fitur pada sistem *helpdesk* perguruan tinggi yang telah berjalan, yaitu agen percakapan (*chatbot*) berbasis otomatisasi alur kerja (*workflow automation*) dan mekanisme notifikasi menggunakan protokol WebSocket, sehingga pembaruan status dapat dikirim secara *real-time* tanpa pemuatan ulang halaman. Sistem dikembangkan menggunakan pendekatan *mixed methods* yang dipadukan dengan metodologi *Software Development Life Cycle (SDLC) Agile Kanban*, dan diimplementasikan pada arsitektur berbasis *framework* PHP serta JavaScript modern. Fitur *chatbot* berperan mengotomatisasi respons terhadap pertanyaan rutin (FAQ) dan memandu pembuatan tiket tanpa intervensi staf, sementara mekanisme notifikasi mendorong pembaruan status tiket secara instan ke antarmuka pengguna. Pengujian fungsional menggunakan *Blackbox Testing* dengan teknik *Equivalence Partitioning* menunjukkan bahwa seluruh fungsi berjalan sesuai spesifikasi. Evaluasi penerimaan pengguna melalui *User Acceptance Testing (UAT)* menghasilkan tingkat penerimaan sebesar 93,75% dari sisi pengguna akhir dan 94,28% dari sisi administrator, mengindikasikan tingkat kegunaan dan penerimaan sistem yang sangat baik. Implementasi kedua fitur tersebut mampu mengotomatisasi penanganan pertanyaan rutin, mempercepat penyampaian informasi status tiket, serta meningkatkan transparansi layanan tanpa memerlukan perubahan pada arsitektur inti sistem *helpdesk* yang telah berjalan.



Copyright © [JITET](http://www.jitet.org) (Jurnal Informatika dan Teknik Elektro Terapan). This article is an open access article distributed under terms and conditions of the Creative Commons Attribution (CC BY NC)

Abstract. Helpdesk services in higher education institutions frequently face operational efficiency challenges, including high reporting volumes, manual ticket handling processes, and limited transparency regarding complaint status for users. This study proposes the implementation of two features within an existing university helpdesk system: a conversational agent (*chatbot*) based on workflow automation, and a notification mechanism using the WebSocket protocol, enabling status updates to be delivered in real time without requiring a page reload. The system was developed using a mixed methods approach combined with the Agile Kanban Software Development Life Cycle (SDLC) methodology, and implemented on an architecture based on modern PHP and JavaScript frameworks. The chatbot feature automates responses to routine inquiries (FAQ) and guides users through ticket creation without staff intervention, while the notification mechanism pushes ticket status updates instantly to the user interface. Functional testing using Blackbox Testing with the Equivalence Partitioning technique showed that all functions operated according to specification. User acceptance evaluation through User Acceptance Testing (UAT) yielded an acceptance rate of 93.75% from end users and 94.28% from administrators, indicating a high level of system usability and acceptance. The implementation of

both features successfully automated the handling of routine inquiries, accelerated the delivery of ticket status information, and improved service transparency without requiring changes to the existing core architecture of the helpdesk system.

1. PENDAHULUAN

Perkembangan teknologi informasi telah mendorong transformasi digital di berbagai sektor, termasuk pendidikan tinggi, yang ditandai dengan meningkatnya pemanfaatan sistem informasi untuk mendukung layanan akademik dan administratif. Transformasi tersebut diarahkan pada penyederhanaan proses kerja serta penyediaan layanan yang lebih responsif bagi sivitas akademika. Penerapan sistem informasi manajemen, pengarsipan elektronik, dan komunikasi digital pada perguruan tinggi terbukti mempercepat pengolahan data dan memperkuat kolaborasi antarunit [1].

Salah satu fungsi utama dalam mendukung operasional teknologi informasi di perguruan tinggi adalah layanan *helpdesk*, yang menangani insiden teknis dan permintaan layanan dari sivitas akademika. Namun, mekanisme layanan *helpdesk* pada banyak perguruan tinggi negeri masih menghadapi tantangan efisiensi. Ketidakseimbangan antara tingginya volume pelaporan insiden dengan keterbatasan jumlah teknisi yang tersedia kerap menyebabkan penanganan masalah tidak dapat dilakukan sesegera mungkin [2]. Kondisi ini diperburuk oleh proses penanganan tiket yang belum sepenuhnya terautomasi, sehingga pengguna kesulitan memantau status penyelesaian keluhan mereka secara langsung. Ketidakjelasan estimasi waktu penyelesaian pada akhirnya berdampak pada produktivitas kerja dan tingkat kepuasan pengguna layanan.

Sejumlah penelitian terdahulu telah mengupayakan mitigasi terhadap permasalahan tersebut, meskipun pendekatan yang digunakan umumnya berkembang secara terpisah. Implementasi *chatbot* berbasis web terbukti mampu memberikan dukungan layanan informasi selama 24/7, mengotomasi respons terhadap pertanyaan rutin, dan mengurangi waktu tunggu pengguna tanpa memerlukan intervensi manusia secara langsung [3], [4]. Mekanisme notifikasi *real-time* menggunakan protokol *WebSocket* juga telah diadopsi pada berbagai sistem

untuk mendorong pembaruan status secara instan tanpa memerlukan pemuatan ulang halaman [5]. Namun, penelitian yang mengintegrasikan otomatisasi alur kerja *chatbot* dan notifikasi *real-time* berbasis *WebSocket* secara simultan, tanpa mengubah arsitektur inti sistem warisan yang sudah berjalan, masih relatif jarang ditemukan [6], [7]. Kesenjangan ini menjadi celah penelitian yang relevan untuk dieksplorasi, khususnya bagi institusi pendidikan dengan keterbatasan sumber daya untuk melakukan pembangunan ulang sistem secara menyeluruh.

Oleh karena itu, penelitian ini bertujuan untuk mengimplementasikan dan mengevaluasi integrasi fitur *chatbot* berbasis otomatisasi alur kerja dan sistem notifikasi *real-time* pada platform *helpdesk* perguruan tinggi yang telah berjalan, tanpa mengubah arsitektur inti sistem yang sudah ada. Pengembangan sistem dilakukan menggunakan metodologi *Software Development Life Cycle* (SDLC) dengan pendekatan *Agile Kanban* untuk mendukung penambahan fitur secara inkremental pada sistem warisan. Penelitian ini diharapkan memberikan bukti empiris mengenai efektivitas integrasi kedua teknologi tersebut dalam meningkatkan efisiensi operasional layanan dan kepuasan pengguna di lingkungan perguruan tinggi.

2. TINJAUAN PUSTAKA

Penerapan *chatbot* pada layanan *helpdesk* perguruan tinggi telah banyak dikaji sebagai cara mengurangi beban kerja staf dalam menjawab pertanyaan rutin. Sistem *chatbot* berbasis pemahaman *intent* dan model AI hybrid terbukti mampu menangani volume pertanyaan FAQ secara otomatis dengan tingkat akurasi yang tinggi, sehingga staf dapat lebih fokus menangani keluhan yang membutuhkan penilaian manusia [8], [9]. Meski demikian, penelitian-penelitian tersebut umumnya memperlakukan *chatbot* sebagai fitur berdiri sendiri, tanpa ket-

erkaitan langsung dengan mekanisme pelacakan status tiket pada sistem *helpdesk* yang menaunginya.

Notifikasi *real-time* melalui protokol *WebSocket* juga menjadi fokus penelitian lain untuk mempercepat distribusi informasi status kepada pengguna. Berbeda dengan *polling* HTTP yang membebani server dengan permintaan berulang, koneksi *full-duplex* pada *WebSocket* memungkinkan pembaruan status terkirim langsung ke antarmuka pengguna tanpa pemuatan ulang halaman [10], [11]. Penelitian-penelitian ini menempatkan notifikasi *real-time* sebagai peningkatan pada sistem *ticketing* yang sudah ada, tetapi tidak menyertakan lapisan interaksi otomatis pada sisi masukan pengguna.

Belum ditemukan penelitian yang menggabungkan otomatisasi *chatbot* dan notifikasi *real-time* berbasis *WebSocket* secara simultan pada satu sistem *helpdesk* warisan yang telah berjalan [12], [13]. Kompleksitas integrasi dua fitur pada arsitektur *legacy*, tanpa mengganggu operasional sistem yang sedang digunakan, membutuhkan pendekatan pengembangan yang memungkinkan pengujian dan penyesuaian bertahap. Agile Kanban dipilih karena visualisasi alur kerja pada papan Kanban dan pembatasan pekerjaan dalam proses memungkinkan setiap fitur diuji dan diintegrasikan secara inkremental, sehingga risiko gangguan terhadap sistem yang sudah berjalan dapat diminimalkan sejak tahap awal pengembangan.

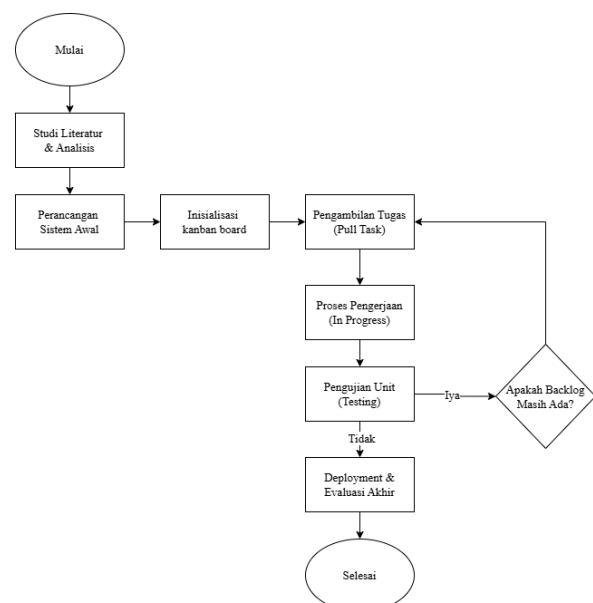
Penelitian ini menghubungkan langsung hasil klasifikasi *chatbot* dengan pemicu notifikasi *real-time*, sehingga perubahan status tiket yang dihasilkan dari interaksi *chatbot* dapat langsung terdorong ke antarmuka pengguna tanpa entri manual staf sebagai perantara. Pendekatan ini diterapkan pada sistem *legacy* yang telah beroperasi, tanpa mengganti struktur basis data atau arsitektur inti yang ada. Kontribusi penelitian ini terletak pada integrasi dua lapisan otomatisasi ke dalam satu alur kerja, bukan pada pengembangan salah satu teknologi dari nol.

Untuk menjawab kesenjangan tersebut, penelitian ini menerapkan metodologi Agile Kanban, yang memungkinkan integrasi *chatbot* dan notifikasi *real-time* diuji dan disesuaikan secara bertahap pada sistem yang sedang berjalan. Bagian metodologi selanjutnya menjelaskan tahapan penerapan Agile Kanban tersebut, mulai dari analisis kebutuhan hingga pengujian setiap iterasi fitur.

3. METODE PENELITIAN

Penelitian ini menggunakan pendekatan *mixed methods* dengan metodologi *Software Development Life Cycle* (SDLC) Agile Kanban untuk merancang dan mengimplementasikan *chatbot* generatif berbasis Gemini API yang terintegrasi dengan *workflow automation* n8n, serta sistem notifikasi *real-time* berbasis *WebSocket*. Agile Kanban dipilih karena mendukung pengembangan fitur secara inkremental dengan membatasi *Work-in-Progress* (WIP), sehingga setiap perubahan kebutuhan dapat diakomodasi tanpa mengganggu alur kerja tim maupun operasional sistem yang sudah berjalan.

3.1. Rancangan Penelitian



Gambar 1. Rancangan Penelitian

3.2 Studi Literatur dan Analisis

Tahap ini menjadi fondasi bagi keseluruhan proses pengembangan, mencakup observasi langsung terhadap sistem *helpdesk* yang sedang berjalan, wawancara semi terstruktur

dengan lima kelompok pemangku kepentingan di lingkungan UNSIKA, serta analisis kebutuhan pengguna dan kebutuhan sistem. Hasil dari tahap ini berupa rumusan kebutuhan fungsional dan non-fungsional yang menjadi acuan pada tahap perancangan berikutnya.

3.3 Perancangan Sistem Awal

Berdasarkan kebutuhan yang telah dirumuskan, dilakukan perancangan arsitektur sistem menggunakan pendekatan *layered architecture*. Perancangan ini dituangkan ke dalam tiga artefak pemodelan *Unified Modeling Language* (UML), yaitu *Use Case Diagram* untuk menggambarkan interaksi pengguna dengan sistem, *Activity Diagram* untuk memodelkan alur proses bisnis, dan *Entity Relationship Diagram* (ERD) untuk mendefinisikan struktur basis data beserta relasi antarentitas.

3.4 Inisialisasi Kanban Board

Setelah rancangan sistem awal selesai disusun, tim pengembang menginisialisasi papan *Kanban* pada platform Trello sebagai media visualisasi alur kerja. Papan ini terdiri atas lima kolom, yaitu *Backlog*, *To Do*, *In Progress*, *Testing*, dan *Done*, yang berfungsi sebagai acuan bersama bagi seluruh anggota tim dalam memantau progres pengerjaan.

3.5 Pengambilan Tugas (Pull Task)

Pertama mengambil tugas dari kolom *To Do* berdasarkan prinsip *pull-based workflow*, dengan prioritas ditentukan melalui metode *MoSCoW* (*Must have*, *Should have*, *Could have*, *Won't have*). Batas *Work in Progress* (WIP) diterapkan sebesar satu tugas aktif per pengembang untuk menjaga fokus dan mencegah tumpang tindih pekerjaan.

3.6 Proses Pengerjaan (In Progress)

Tugas yang telah ditarik dikerjakan melalui implementasi kode sesuai spesifikasi fungsional, integrasi dengan komponen sistem lain seperti RESTful API, basis data MySQL, komponen *frontend* Vue.js, dan *webhook* n8n, serta pengujian mandiri sebelum tugas dinyatakan selesai pada tahap ini.

3.7 Pengujian Unit (Testing)

Tugas yang telah selesai dikembangkan dipindahkan ke kolom *Testing* untuk menjalani serangkaian pengujian, meliputi *unit testing* menggunakan *framework* PHPUnit, *integration testing* antarmodul, dan *code review* oleh pengembang lain untuk memastikan kesesuaian dengan standar kualitas yang ditetapkan. Apabila masih terdapat tugas yang belum dikerjakan, alur kembali (*looping*) ke tahap Pengambilan Tugas (*Pull Task*), sehingga siklus pengambilan, pengerjaan, dan pengujian tugas terus berulang. Mekanisme ini mencerminkan prinsip pengembangan inkremental, di mana fitur *chatbot* dan notifikasi *real-time* tidak dibangun sekaligus dalam satu tahap besar, melainkan dipecah menjadi satuan tugas kecil yang dikerjakan dan diuji satu demi satu, sehingga risiko kegagalan sistem secara menyeluruh dapat diminimalkan dan penyesuaian kebutuhan dapat dilakukan tanpa mengganggu keseluruhan alur pengembangan.

3.8 Pengecekan Kondisi: Apakah Backlog Masih Ada?

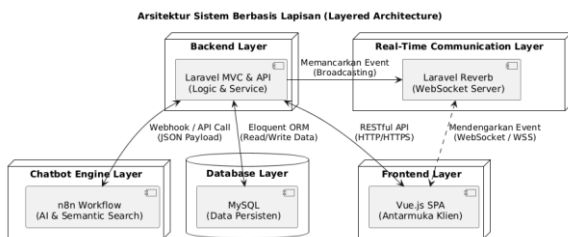
Tahap ini menjadi inti dari sifat iteratif metodologi *Agile Kanban* yang diterapkan dalam penelitian ini. Setelah satu tugas dinyatakan lulus pengujian dan dipindahkan ke kolom *Done*, alur pengembangan tidak langsung menuju tahap akhir, melainkan melakukan pengecekan terhadap kolom *Backlog*.

3.9 Deployment dan Evaluasi Akhir

Setelah seluruh tugas mencapai kolom *Done* dan membentuk fitur yang utuh, dilakukan *deployment* ke lingkungan produksi melalui persiapan *checklist*, *backup* basis data, dan eksekusi menggunakan *continuous deployment pipeline* untuk meminimalkan waktu henti sistem. Selanjutnya dilakukan *smoke test* untuk memverifikasi bahwa sistem berjalan normal, dilanjutkan dengan evaluasi kesesuaian terhadap kebutuhan fungsional dan non-fungsional, pengujian performa serta keamanan, dan pengumpulan umpan balik awal dari pengguna.

4. HASIL DAN PEMBAHASAN

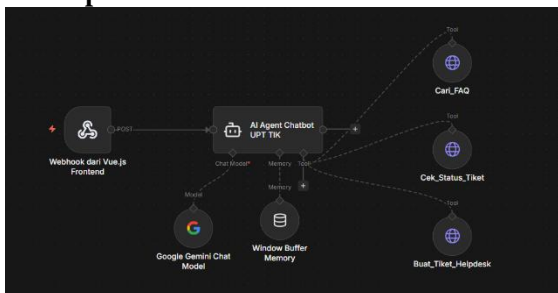
4.1. Implementasi Arsitektur Sistem



Gambar 2 Arsitektur Sistem Berbasis Lapisan

Sistem helpdesk ini dibangun mengikuti pendekatan *layered architecture*, dengan empat komponen yang masing-masing menjalankan tanggung jawab fungsional berbeda: lapisan antarmuka (*frontend*), lapisan logika bisnis (*backend*), mesin otomatisasi percakapan (*chatbot engine*), dan lapisan komunikasi asinkron (*real-time layer*). Pemisahan ini bukan sekadar preferensi desain. Pemrosesan bahasa alami dan transaksi data konvensional memiliki karakteristik beban komputasi yang berbeda, sehingga jika keduanya digabung dalam satu lapisan, salah satu proses akan membebani proses lainnya. Memisahkan keduanya menjaga performa sistem tetap stabil.

4.2. Implementasi Chatbot AI Berbasis N8N

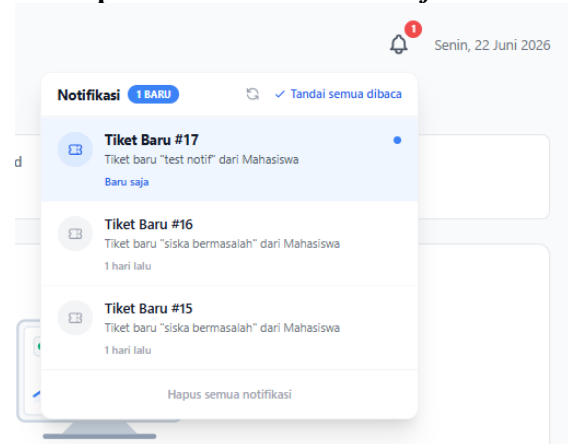


Gambar 3 Implementasi workflow pada n8n

Pemrosesan bahasa alami pada fitur chatbot ditempatkan pada server n8n yang berjalan terpisah dari peladen aplikasi utama, sehingga beban komputasi kecerdasan buatan tidak mengganggu kinerja layanan helpdesk. Ketika pengguna mengirimkan pertanyaan, backend meneruskan input tersebut ke endpoint webhook n8n dalam bentuk payload JSON, lalu menahan koneksi secara asinkron hingga jawaban selesai diproses, sehingga kredensial API kecerdasan buatan tidak pernah terekspos di sisi klien. Di dalam n8n, permintaan diproses oleh

node AI Agent yang ditenagai model bahasa Google Gemini dan dirangkai dalam arsitektur agen dinamis, bukan alur kerja linear konvensional. Komponen Window Buffer Memory merekam riwayat percakapan sebelumnya sehingga agen dapat mempertahankan konteks obrolan, bukan memperlakukan setiap pesan sebagai permintaan yang berdiri sendiri.

4.3. Implementasi Real Time Notification



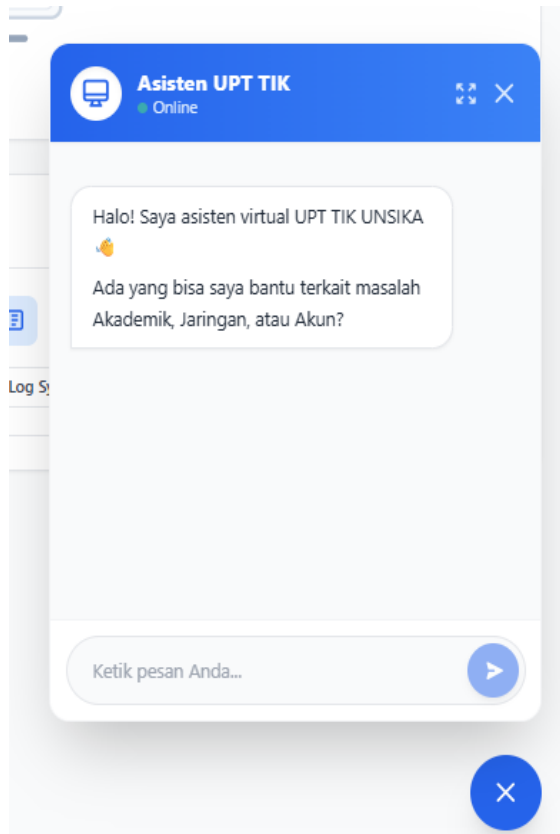
Gambar 4 Tampilan Real Time Notification

Real time notification pada sistem ini dibangun dengan arsitektur event-driven menggunakan protokol WebSocket, dengan Laravel Reverb berperan sebagai broadcast server self-hosted. Pendekatan ini menggantikan mekanisme HTTP polling konvensional yang mengharuskan klien mengirim permintaan berulang hanya untuk memeriksa apakah ada pembaruan data. Koneksi antara server dan peramban dijaga tetap terbuka sepanjang sesi pengguna, sehingga begitu status data berubah, server langsung mendorong pembaruan ke klien tanpa menunggu permintaan berikutnya. Notifikasi pun diterima dalam hitungan milidetik setelah event dipicu, jauh lebih cepat dibandingkan siklus polling konvensional yang umumnya berjalan setiap beberapa detik.

4.4. Hasil Antarmuka Pengguna

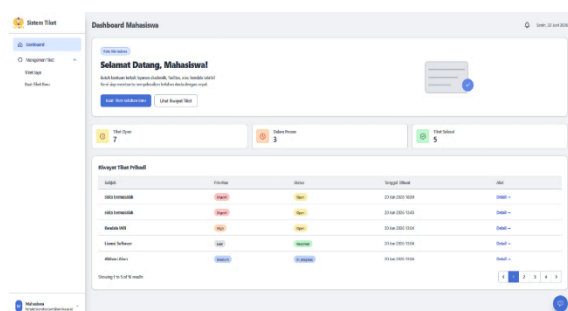
Implementasi antarmuka dilakukan untuk mempermudah pengguna dalam berinteraksi dengan layanan helpdesk secara mandiri dan real time. Widget chatbot digunakan untuk memberikan jawaban instan atas pertanyaan repetitif serta memfasilitasi pengajuan tiket baru langsung dari jendela percakapan. Selain itu, sistem menyediakan halaman dashboard Mahasiswa untuk memantau status tiket pelaporan pribadi

secara terpusat, serta dashboard UPA TIK yang memberikan kendali penuh atas penugasan dan penyelesaian tiket operasional.



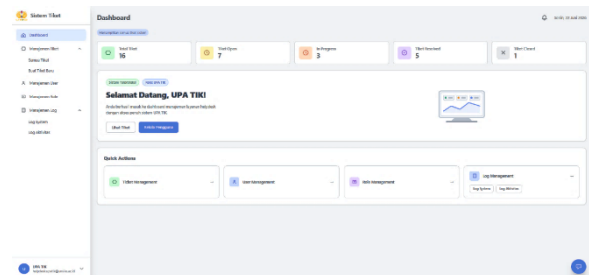
Gamabar 5 Tampilan Widget Chatbot

Widget chatbot ditampilkan dalam bentuk jendela *pop-up* dengan indikator status "Online" pada header, menandakan layanan asisten virtual UPT TIK sedang aktif melayani pengguna. Pesan sambutan otomatis langsung menyapa pengguna dan menawarkan bantuan seputar kendala akademik, jaringan, atau akun, sementara kolom input di bagian bawah memungkinkan pengguna mengetik pertanyaan tanpa perlu berpindah dari halaman yang sedang dibuka.



Gambar 6 Tampilan Dashboard User

Dashboard pada sisi mahasiswa dirancang dengan tampilan ringkas yang berfokus pada pemantauan status tiket pelaporan pribadi. Pengguna dapat melihat progres penanganan keluhannya secara langsung dari halaman ini tanpa memerlukan akses ke fungsionalitas manajerial yang lebih luas.



Gambar 7 Tampilan Dashboard Admin

Dashboard administrator menyajikan visualisasi data yang lebih komprehensif, mencakup ringkasan statistik volume tiket dan grafik distribusi status penanganan secara menyeluruh. Tata letak ini memberi staf UPA TIK gambaran cepat mengenai beban kerja operasional, sehingga penugasan dan eskalasi tiket dapat diprioritaskan berdasarkan data yang termonitor secara *real time*.

4.5. Hasil Pengujian Sistem

Pengujian sistem dilakukan dalam dua tahap: validasi fungsionalitas teknis dan pengukuran tingkat penerimaan pengguna akhir terhadap fitur *chatbot* serta *real time notification* yang telah diimplementasikan.

4.5.1. Blackbox Testing

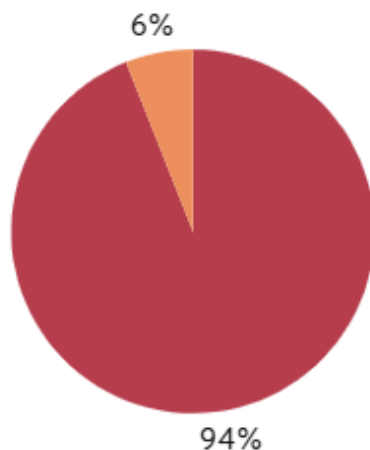
Validasi teknis dilakukan lewat *Alpha Testing* berbasis *blackbox*, yang menguji kesesuaian antara input pengguna dan respons sistem tanpa menelusuri struktur kode di baliknya. Skenario yang diuji mencakup perubahan status tiket oleh administrator, kemunculan *pop-up* notifikasi secara instan di sisi klien, hingga penambahan otomatis pada indikator jumlah notifikasi belum dibaca. Seluruh skenario berjalan sesuai harapan tanpa ditemukan *bug*, yang mengonfirmasi bahwa alur komunikasi antara *backend*, *chatbot engine*, dan lapisan *real-time* bekerja sebagaimana dirancang.

Tabel 1 Hasil *Blackbox Testing*

No	Aspek Pengujian	Jumlah Test Case	Status
1	Chatbot FAQ	5	Berhasil
2	Pembuatan Tiket melalui Chatbot	4	Berhasil
3	Real-Time Notification	6	Berhasil
4	Pembaruan Status Tiket	5	Berhasil
5	Integrasi Sistem	4	Berhasil
Total		24	Berhasil

4.5.2. User Acceptance Testing (UAT)

Penerimaan pengguna diukur lewat *Beta Testing* yang menggabungkan wawancara kualitatif dengan pengindeksan skor kuesioner secara kuantitatif. Mahasiswa memberikan indeks penerimaan sebesar 75%, masuk kategori "Diterima", terutama karena *chatbot* dinilai cukup akurat menjawab pertanyaan repetitif seputar FAQ. Administrator memberikan indeks penerimaan yang lebih tinggi, 94,3%, masuk kategori "Sangat Diterima", sebab notifikasi instan mempermudah pelacakan dan eskalasi tiket dibandingkan mekanisme pelaporan konvensional

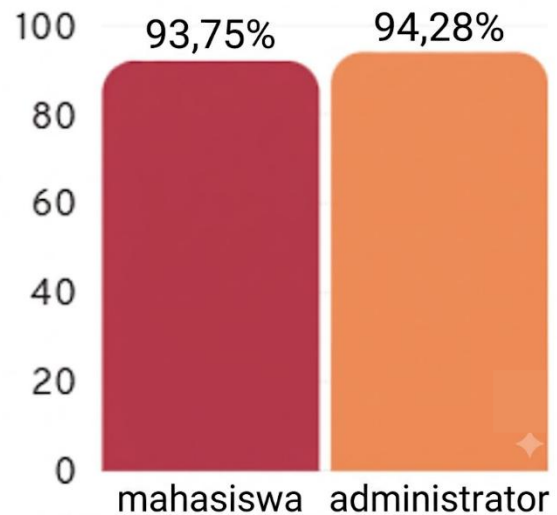


Gambar 8 Persentase Penerimaan Pengguna

Secara agregat, kedua kelompok responden memberikan penilaian positif sebesar 94%, sedangkan 6% sisanya berada pada kategori netral dan sebagian besar berasal dari penilaian mahasiswa yang masih menyesuaikan diri dengan *chatbot*. Hasil ini menunjukkan sistem yang

dikembangkan telah diterima secara luas, baik dari sisi kemudahan pelaporan maupun efisiensi penanganan tiket.

Diagram batang di atas membandingkan tingkat penerimaan antara dua kelompok responden,



Gambar 9 Presentase Penerimaan Pengguna dan Administrator

yaitu mahasiswa dengan skor 93,75% dan administrator dengan skor 94,28%. Selisih keduanya tidak signifikan, menunjukkan bahwa sistem helpdesk diterima secara merata baik oleh pengguna akhir yang melaporkan tiket maupun administrator yang mengelola dan menindaklanjuti tiket tersebut.

5. KESIMPULAN

Berdasarkan proses analisis kebutuhan, perancangan, implementasi, hingga pengujian yang telah dilakukan terhadap platform helpdesk UPT TIK, penelitian ini sampai pada beberapa simpulan. Arsitektur AI Agent yang dibangun di atas platform n8n berhasil mengotomatisasi penanganan keluhan tingkat dasar (Tier 1 Support). Kombinasi model bahasa besar, Window Buffer Memory, dan instrumen pencarian semantik memungkinkan chatbot memahami konteks percakapan secara natural, menjawab pertanyaan FAQ, serta memfasilitasi pembuatan dan pelacakan tiket tanpa intervensi staf. Fungsionalitas real-time notification, yang dibangun menggunakan Laravel Reverb di atas protokol WebSocket, berhasil menggantikan mekanisme HTTP polling yang selama ini membebani peladen; pembaruan status tiket

kini dapat didorong langsung ke antarmuka klien secara asinkron dan instan. Pada aspek proses pengembangan, pendekatan Agile dengan metode Kanban terbukti efektif mengendalikan siklus rekayasa perangkat lunak. Batasan pengerjaan (Work in Progress Limit) mencegah penumpukan beban kerja, dan seluruh spesifikasi utama hasil analisis MoSCoW terealisasi sesuai target waktu iterasi. Validasi kualitas perangkat lunak turut mengonfirmasi hasil ini: pengujian Alpha berbasis Blackbox Testing tidak menemukan cacat logika pada integrasi webhook maupun broadcasting, sedangkan pengujian Beta berbasis User Acceptance Testing mencatat tingkat penerimaan pengguna pada kategori "Sangat Diterima", dengan skor 93,75% dari mahasiswa dan 94,28% dari Pimpinan dan Staf UPT TIK.

UCAPAN TERIMA KASIH

Penulis mengucapkan terima kasih yang sebesar-besarnya kepada Bapak Jajam Haerul Jaman, dan Bapak Iqbal Maulana atas segala bimbingan, arahan, serta motivasi yang diberikan selama proses penelitian dan penulisan artikel jurnal ini. Ucapan terima kasih juga ditujukan kepada seluruh pihak yang telah membantu dan memberikan dukungan, baik secara langsung maupun tidak langsung, sehingga penelitian ini dapat diselesaikan dengan baik. Semoga segala kebaikan dan dukungan yang telah diberikan mendapatkan balasan yang setimpal dari Tuhan Yang Maha Esa.

DAFTAR PUSTAKA

- [1] L. Mondial, D. Deha, S. Fil, and M. I. Kom, "Kata Kunci: Efisiensi administrasi, Sekretaris, Teknologi informasi," 2024.
- [2] G. Muhammad Fahmi Prasetio, Apriade Voutama, "Implementasi Sistem Pakar Dengan Metode Certainty Factor Dan Forward Chaining Pada Helpdesk UPT TIK Unsika," vol. 8, no. 13, pp. 289–302, 2022.
- [3] Deden Moh Alfiansyah, Wiilys, Lila Setiyani, Devi Fajar Wati, and Dedih, "Pengembangan Chatbot Berbasis Web untuk Layanan Informasi di Horizon University," *bit-Tech*, vol. 7, no. 3, pp. 1068–1077, 2025, doi: 10.32877/bt.v7i3.2318.
- [4] A. Lubis and I. Sumartono, "Implementasi Layanan Akademik Berbasis Chatbot untuk Meningkatkan Interaksi Mahasiswa," *Media Online*, vol. 3, no. 5, pp. 397–403, 2023, [Online]. Available: <https://djournals.com/resolusi>
- [5] I. G. Susrama, M. Diyasa, G. S. Budiwitjaksono, and A. Masrifah, "Push Notification Using the Websocket in the Application of Sistem Informasi Uji Kompetensi Online (Situk) Version 2," vol. 2022, pp. 154–163, 2022, doi: 10.11594/nstp.2022.2426.
- [6] S. N. Saifullah, Hendri Yawan, Nining Syafitri, "INTEGRATING AI CHATBOT INTO LEARNING MANAGEMENT SYSTEM: Email: hendriyawan@usn.ac.id Universitas Dayanu Ikhsanuddin, Indonesia, 2 Universitas Sembilanbelas November Kolaka Abstract," vol. 5, no. 2, pp. 1346–1359, 2024.
- [7] H. Abu Bakar, R. Razali, and D. I. Jambari, "A Qualitative Study of Legacy Systems Modernisation for Citizen-Centric Digital Government," *Sustainability (Switzerland)*, vol. 14, no. 17, 2022, doi: 10.3390/su141710951.
- [8] P. D. Larasati, A. Irawan, S. Anwar, M. F. Mulya, M. A. Dewi, and I. Nurfatima, "Chatbot helpdesk design for digital customer service," vol. 1, no. 3, pp. 138–145, 2022.
- [9] M. Anbiya, N. U. R. Islam, B. Warsito, and O. K. Y. D. W. I. Nurhayati, "AI-DRIVEN CHATBOT IMPLEMENTATION FOR ENHANCING CUSTOMER SERVICE IN HIGHER EDUCATION: A CASE STUDY FROM UNIVERSITAS NEGERI SEMARANG," vol. 102, no. 14, pp. 5690–5701, 2024.
- [10] M. Singh, "Integrating Artificial Intelligence with Legacy Systems: A Systematic Analysis of Challenges and Strategic Considerations," *European Journal of Computer Science and Information Technology*, vol. 13, no. 32, pp. 38–45, 2025, doi: 10.37745/ejcsit.2013/vol13n323845.
- [11] M. N. Wahyuddin and D. P. Putri, "Designing a helpdesk ticketing system to improve the efficiency of IT support services on an XYZ company-based website using the Laravel framework," vol. 14, no. 1, pp. 9–24, 2024.
- [12] K. S. Sarat Dyuthi, "Configuring Real-Time Event Processing of Api Gateway with Aws and Websocket Api's," *Journal of Informatics Education and Research*, vol. 4, no. 3, pp. 3324–3337, 2024, doi: 10.52783/jier.v4i3.1711.
- [13] Rudi Hartono, "Penerapan Kanban Model Sebagai Metode Perancangan Sistem Informasi (Studi Kasus: Pemetaan Sekolah SMA/K/MA Kota Tasikmalaya)," *Jurnal Petik*, vol. 8, no. 1, pp. 27–34, 2022, [Online]. Available: <https://journal.institutpendidikan.ac.id/index.php/petik/article/view/1247>