

Design and Implementation of a SaaS-Based Helpdesk System Using a Multi-Tenant Architecture for Customer Complaint Management

[Rancang Bangun Sistem Helpdesk Berbasis SaaS dengan Arsitektur Multi-Tenant untuk Pengelolaan Keluhan Pelanggan]

Muhammad Syamsuddin¹⁾, Irwan Alnarus Kautsar^{*,2)}

¹⁾Program Studi Informatika, Universitas Muhammadiyah Sidoarjo, Indonesia

²⁾ Program Studi Informatika, Universitas Muhammadiyah Sidoarjo, Indonesia

*Email Penulis Korespondensi: irwan@umsida.ac.id

Abstract. *Customer complaint management is an important aspect of maintaining service quality and customer satisfaction. However, complaint handling processes are often conducted through multiple communication channels, making monitoring and documentation difficult. This study aims to design and develop a Software as a Service (SaaS)-based helpdesk system using a Multi-Tenant architecture for customer complaint management. The research employed the Research and Development (R&D) method with a Prototyping approach, while requirement analysis was conducted using the Income-Based Education (IBE) 1.0 framework. The resulting system implements a Single Database Multi-Schema architecture to provide data isolation among tenants. The developed prototype supports tenant management, ticket management, complaint tracking, product management, and automated email notifications. Black Box Testing results indicate that the implemented functionalities operate according to the defined requirements. This research contributes to the implementation of a SaaS Multi-Tenant model in customer complaint management systems.*

Keywords - Software as a Service, Multi-Tenant, Helpdesk, Ticket Management System, Subdomain Tenant

Abstrak. *Pengelolaan keluhan pelanggan merupakan aspek penting dalam menjaga kualitas layanan dan kepuasan pelanggan. Namun, proses penanganan keluhan sering kali dilakukan melalui berbagai saluran komunikasi, sehingga menyebabkan proses pemantauan dan dokumentasi menjadi sulit. Penelitian ini bertujuan untuk merancang dan mengembangkan sistem helpdesk berbasis Software as a Service (SaaS) dengan menggunakan arsitektur Multi-Tenant untuk pengelolaan keluhan pelanggan. Penelitian ini menggunakan metode Research and Development (R&D) dengan pendekatan Prototyping, sedangkan analisis kebutuhan dilakukan menggunakan framework Income-Based Education (IBE) 1.0. Sistem yang dihasilkan menerapkan arsitektur Single Database Multi-Schema untuk menyediakan isolasi data antar tenant. Prototipe yang dikembangkan mendukung pengelolaan tenant, manajemen tiket, pelacakan keluhan, pengelolaan produk, serta notifikasi email otomatis. Hasil pengujian menggunakan Black Box Testing menunjukkan bahwa fungsi-fungsi yang diimplementasikan berjalan sesuai dengan kebutuhan yang telah ditentukan. Penelitian ini memberikan kontribusi terhadap penerapan model SaaS Multi-Tenant dalam sistem pengelolaan keluhan pelanggan.*

Kata Kunci - Software as a Service, Multi-Tenant, Helpdesk, Ticket Management System, Subdomain Tenant

I. PENDAHULUAN

Transformasi digital telah terjadi di berbagai sektor bisnis, seperti manajemen layanan pelanggan, karena kemajuan teknologi informasi. Dalam menghadapi persaingan yang semakin kompetitif, kualitas pelayanan pelanggan menjadi salah satu faktor penting yang memengaruhi kepuasan pelanggan, loyalitas pengguna, dan citra organisasi. Proses penanganan keluhan pelanggan adalah komponen yang sangat penting dalam pelayanan pelanggan. Bisnis dapat menggunakan keluhan pelanggan sebagai sumber informasi untuk menemukan masalah layanan, mengevaluasi kualitas produk, dan terus meningkatkan efektivitas operasional. Akibatnya, diperlukan sistem pengelolaan keluhan yang terorganisir, tercatat, dan mudah dipantau oleh semua pihak yang terlibat [1]-[4].

Pada praktiknya, banyak organisasi masih mengelola keluhan pelanggan melalui berbagai media komunikasi yang terpisah, seperti surat elektronik (email), aplikasi perpesanan instan, telepon, maupun media sosial. Penggunaan banyak saluran komunikasi tanpa sistem terintegrasi sering menimbulkan berbagai permasalahan, antara lain kesulitan dalam mendokumentasikan laporan pelanggan, keterbatasan dalam memantau status penyelesaian masalah, duplikasi pekerjaan, serta risiko kehilangan informasi penting selama proses penanganan keluhan. Selain itu, pelanggan juga mengalami kesulitan untuk mengetahui perkembangan penyelesaian masalah yang telah dilaporkan karena tidak tersedia mekanisme pelacakan yang terpusat dan transparan. Kondisi ini dapat mengurangi tingkat kepuasan pelanggan dan menyebabkan layanan menjadi kurang efisien [5]-[8].

Sistem helpdesk berbasis ticketing adalah solusi populer untuk masalah ini. Sistem ini memungkinkan setiap keluhan pelanggan dicatat dalam bentuk tiket yang memiliki identitas unik sehingga proses penanganan dapat dilakukan secara terstruktur dan terdokumentasi. Melalui sistem ticketing, organisasi dapat melakukan distribusi tugas

kepada petugas yang bertanggung jawab, memantau progres penyelesaian masalah, menyimpan histori aktivitas penanganan, serta menghasilkan laporan yang dapat digunakan sebagai bahan evaluasi layanan. Oleh karena itu, sistem helpdesk tidak hanya mencatat keluhan tetapi juga membantu mengelola layanan pelanggan dan terus meningkatkan kualitas layanan [9], [10].

Seiring meningkatnya kebutuhan digitalisasi layanan, model pengembangan perangkat lunak berbasis *Software as a Service* (SaaS) semakin banyak diadopsi karena menawarkan fleksibilitas, efisiensi biaya, serta kemudahan pengelolaan sistem. Pada model SaaS, penyedia layanan dapat mengelola satu aplikasi yang digunakan oleh banyak organisasi secara bersamaan melalui jaringan internet tanpa memerlukan instalasi terpisah pada setiap organisasi pengguna. Pendekatan ini memungkinkan organisasi memperoleh akses terhadap layanan perangkat lunak dengan biaya implementasi yang lebih rendah dibandingkan pengembangan sistem secara mandiri. Selain itu, pembaruan sistem, pemeliharaan aplikasi, serta pengelolaan infrastruktur dapat dilakukan secara terpusat oleh penyedia layanan sehingga meningkatkan efisiensi operasional [14], [15].

Implementasi model SaaS umumnya didukung oleh arsitektur Multi-Tenant, yaitu pendekatan yang memungkinkan satu aplikasi melayani banyak organisasi atau tenant dalam lingkungan yang sama dengan tetap menjaga isolasi data masing-masing pengguna. Arsitektur Multi-Tenant memberikan berbagai keuntungan, seperti efisiensi penggunaan sumber daya komputasi, kemudahan pemeliharaan sistem, skalabilitas yang lebih baik, serta kemampuan melayani banyak pelanggan dalam satu platform terintegrasi [11]-[14]. Dengan adanya isolasi data yang baik, setiap tenant dapat menggunakan sistem secara independen tanpa memengaruhi data tenant lainnya. Karakteristik tersebut menjadikan arsitektur Multi-Tenant sebagai salah satu komponen penting dalam implementasi SaaS modern.

Beberapa penelitian terdahulu telah membahas pengembangan sistem helpdesk maupun implementasi arsitektur SaaS dan Multi-Tenant. Mahesa mengembangkan aplikasi helpdesk berbasis web yang menerapkan konsep ticketing untuk mendukung proses penanganan keluhan pengguna pada suatu organisasi [16]. Penelitian tersebut menunjukkan bahwa sistem helpdesk mampu membantu proses pencatatan dan pengelolaan keluhan secara lebih terstruktur dibandingkan metode konvensional. Namun, penelitian tersebut masih berfokus pada implementasi helpdesk dalam lingkungan organisasi tunggal dan belum mengadopsi konsep *Software as a Service* maupun arsitektur Multi-Tenant.

Di sisi lain, penelitian yang dilakukan oleh Aziza dkk. membahas analisis trade-off arsitektur Multi-Tenant sebagai dasar dalam pengembangan sistem berbasis SaaS [17]. Penelitian tersebut menjelaskan berbagai pertimbangan teknis dalam penerapan Multi-Tenant, termasuk aspek efisiensi sumber daya dan pengelolaan basis data. Meskipun demikian, penelitian tersebut tidak berfokus pada pengembangan sistem helpdesk maupun pengelolaan keluhan pelanggan sebagai domain implementasi. Penelitian lain yang dilakukan oleh Krishna membahas kajian literatur mengenai implementasi *Software as a Service* dari perspektif keamanan dan *multitenancy* [18]. Hasil penelitian tersebut menunjukkan bahwa Multi-Tenant merupakan salah satu karakteristik utama yang mendukung skalabilitas dan efisiensi layanan SaaS. Akan tetapi, penelitian tersebut masih bersifat konseptual dan belum mengimplementasikan arsitektur tersebut pada sistem helpdesk yang digunakan secara nyata oleh banyak organisasi.

Berdasarkan penelitian terdahulu, dapat dilihat bahwa penelitian terdahulu cenderung berfokus pada sistem helpdesk dan implementasi SaaS Multi-Tenant secara terpisah. Dalam kebanyakan kasus, penelitian sistem helpdesk tidak menggunakan konsep SaaS dan Multi-Tenant dalam pengelolaan keluhan pelanggan. masih ada celah penelitian atau gap penelitian karena belum ada penelitian yang mengintegrasikan sistem helpdesk, model *Software as a Service*, dan arsitektur Multi-Tenant ke dalam satu platform yang dimaksudkan untuk mendukung pengelolaan keluhan pelanggan secara terpusat [16]-[18].

Berdasarkan permasalahan dan kesenjangan penelitian tersebut, diperlukan pengembangan sistem yang mampu mengintegrasikan mekanisme pengelolaan keluhan pelanggan dengan model layanan SaaS dan arsitektur Multi-Tenant. Integrasi tersebut diharapkan dapat memberikan solusi yang tidak hanya mendukung proses pencatatan dan penanganan keluhan secara terstruktur, tetapi juga memungkinkan satu platform digunakan oleh banyak organisasi secara bersamaan dengan tetap menjaga keamanan dan isolasi data masing-masing tenant. Selain itu, sistem yang dikembangkan diharapkan mampu meningkatkan transparansi proses penanganan keluhan melalui mekanisme pelacakan tiket, histori aktivitas, serta notifikasi otomatis kepada pengguna.

Oleh karena itu, penelitian ini bertujuan untuk merancang dan membangun Sistem Helpdesk Berbasis SaaS dengan Arsitektur Multi-Tenant untuk Pengelolaan Keluhan Pelanggan. Penelitian dilakukan menggunakan metode *Research and Development* (R&D) dengan pendekatan *Prototyping*, sedangkan analisis kebutuhan dan perancangan sistem dilakukan menggunakan kerangka *Income-Based Education* (IBE) 1.0 yang terdiri dari *Cause-Effect-Solution* (CES) Framework, *Functional and Non-Functional* (FNF) Framework, serta pemodelan *Unified Modeling Language* (UML). Hasil penelitian diharapkan dapat memberikan kontribusi dalam penerapan konsep SaaS Multi-Tenant pada domain helpdesk sekaligus menjadi alternatif solusi bagi organisasi dalam meningkatkan kualitas pengelolaan layanan dan keluhan pelanggan.

II. METODE

Untuk menghasilkan sistem helpdesk berbasis Software as a Service (SaaS) dengan arsitektur Multi-Tenant, penelitian ini menggunakan metode *Research and Development* (R&D) dengan pendekatan prototipe. Metode ini dipilih karena fokus penelitian adalah mengembangkan produk perangkat lunak melalui tahapan analisis kebutuhan, perancangan, implementasi, dan pengujian sistem [19].

Penelitian dimulai dengan melihat proses bisnis pengelolaan keluhan pelanggan dan membaca penelitian tentang sistem helpdesk, SaaS, dan arsitektur multi-tenant. Hasil pengumpulan kebutuhan kemudian dianalisis menggunakan pendekatan *Rapid Prototyping Income-Based Education* (IBE) 1.0 framework untuk mendukung proses analisis secara terstruktur dan berorientasi pada pengguna (*user-centered*) yang terdiri dari *Cause-Effect-Solution* (CES) Framework untuk mengidentifikasi permasalahan dan solusi sistem, *Functional and Non-Functional* (FNF) Framework untuk menentukan kebutuhan fitur sistem, serta pemodelan *Unified Modeling Language* (UML) yang merupakan tahap perancangan sistem meliputi *Use Case Diagram*, *Sequence Diagram*, dan *Entity Relationship Diagram* (ERD) [20]-[22]. Penggunaan IBE 1.0 dalam penelitian ini didasari oleh kesesuaiannya dengan kondisi peneliti yang membutuhkan hasil analisis yang cepat namun tetap terstruktur, sejalan dengan prinsip *rapid prototyping* yang menjadi landasan kerangka tersebut [23].

Proses implementasi dilakukan dengan membuat prototype aplikasi berbasis web menggunakan Laravel sebagai backend, React dan Inertia.js sebagai antarmuka pengguna, PostgreSQL sebagai basis data, dan library pendukung untuk menerapkan Multi-Tenant dan manajemen hak akses pengguna.

Pada tahap terakhir, sistem diuji menggunakan metode *Black Box Testing* untuk memastikan bahwa setiap fungsi yang dibuat memenuhi persyaratan fungsional yang telah ditetapkan.

III. HASIL DAN PEMBAHASAN

A. Analisis Kebutuhan Sistem

Tujuan dari tahap analisis kebutuhan sistem adalah untuk menemukan masalah yang terjadi dalam proses pengelolaan keluhan pelanggan dan menentukan komponen sistem yang diperlukan untuk menyelesaikannya. Untuk melakukan analisis, observasi proses bisnis, wawancara dengan stakeholder, dan penelitian literatur terkait sistem helpdesk, Software as a Service (SaaS), dan arsitektur Multi-Tenant digunakan. Hasil analisis ini membentuk fondasi untuk proses perancangan dan pengembangan prototype sistem.

Analisis Permasalahan

Analisis permasalahan dilakukan menggunakan Cause-Effect-Solution (CES) Framework yang merupakan bagian dari metode IBE 1.0. Framework ini digunakan untuk mengidentifikasi hubungan antara penyebab permasalahan (*cause*), dampak yang ditimbulkan (*effect*), serta solusi yang dapat diterapkan (*solution*).

Tabel 1. Hasil Analisis Menggunakan CES Framework

#Problem	Cause (Penyebab)	Effect (Dampak)	Solution (Solusi)
1	Perusahaan kecil-menengah tidak memiliki infrastruktur helpdesk sendiri karena biaya dan kompleksitasnya tinggi	Customer tidak mendapat kepastian apakah keluhan diterima dan ditangani, menurunkan kepercayaan terhadap layanan	Platform helpdesk berbasis SaaS siap pakai — client cukup mendaftar dan sistem langsung aktif tanpa infrastruktur mandiri
2	Keluhan customer masuk melalui channel tidak terstruktur (WhatsApp, email pribadi, telepon) tanpa standar pencatatan	Agent kewalahan atau salah arah karena tidak ada distribusi dan routing keluhan yang sistematis	Form keluhan terpusat per subdomain unik — customer lapor tanpa login, tiket dibuat otomatis dengan nomor unik
3	Tidak ada mekanisme routing keluhan ke tim yang tepat berdasarkan produk yang dikeluhkan	Data keluhan hilang dan tidak terdokumentasi sehingga tidak dapat dijadikan bahan evaluasi layanan	Routing notifikasi ke agent berdasarkan mapping produk-agent — keluhan langsung sampai ke penanggung jawab yang tepat

4	Customer tidak dapat memantau progres penanganan keluhan yang dilaporkan	Customer harus sering follow-up ke PIC	Semua data keluhan tercatat dan terlacak — customer dapat memantau progres tiket secara mandiri via nomor tiket
---	--	--	---

Berdasarkan hasil analisis, ditemukan bahwa proses pengelolaan keluhan pelanggan masih dilakukan melalui berbagai media komunikasi yang tidak terintegrasi sehingga menyebabkan kesulitan dalam pencatatan, distribusi, dan pemantauan status penyelesaian keluhan. Selain itu, pelanggan tidak memiliki mekanisme yang memungkinkan mereka memantau perkembangan laporan secara mandiri. Kondisi tersebut menyebabkan rendahnya transparansi layanan serta meningkatnya risiko kehilangan informasi penting terkait histori penanganan keluhan.

Untuk menyelesaikan masalah tersebut, disarankan untuk membangun sistem helpdesk berbasis SaaS dengan arsitektur Multi-Tenant. Sistem ini akan memiliki kemampuan untuk menyediakan layanan pengelolaan keluhan yang terpusat, tercatat, dan dapat digunakan oleh berbagai organisasi dalam satu platform.

Analisis Kebutuhan Fungsional

Analisis dilakukan untuk menentukan fungsi-fungsi yang harus tersedia pada sistem untuk mendukung proses bisnis yang telah diidentifikasi.

Tabel 2. Kebutuhan Fungsional Sistem

No	Kebutuhan Fungsional	Deskripsi terukur
F-01	Pengelolaan data tenant oleh Super Admin	Sistem menyediakan fitur bagi Super Admin untuk melihat, menambah, memperbarui, dan menghapus data tenant (client) yang terdaftar pada platform SaaS.
F-02	Pendaftaran tenant oleh Super Admin	Sistem memungkinkan Super Admin membuat tenant baru dengan menginput informasi dasar tenant seperti nama perusahaan, identitas tenant, dan konfigurasi awal lingkungan tenant.
F-03	Registrasi perusahaan oleh Admin Tenant	Sistem memungkinkan Admin Tenant melakukan registrasi perusahaan menggunakan email perusahaan dan menentukan subdomain unik dengan batas maksimal 10 karakter sebagai identitas akses tenant.
F-04	Pengelolaan data produk tenant	Admin Tenant dapat mengelola daftar produk atau layanan yang dimiliki oleh tenant dengan fitur CRUD (Create, Read, Update, Delete).
F-05	Pengelolaan data agent tenant	Sistem memungkinkan Admin Tenant menambahkan, mengubah, menghapus, dan mengirimkan undangan akun kepada agent (PIC) yang akan menangani tiket keluhan pelanggan.
F-06	Pemetaan produk dengan agent	Sistem menyediakan fitur mapping antara produk tertentu dengan agent yang bertanggung jawab sehingga tiket berdasarkan produk dapat diarahkan kepada agent terkait.
F-07	Pengiriman keluhan oleh customer tanpa login	Sistem menyediakan formulir pengaduan pada URL subdomain tenant yang dapat diakses customer tanpa autentikasi dengan data input minimal berupa email customer, produk terkait, dan deskripsi keluhan.
F-08	Pembuatan nomor tiket otomatis	Sistem menghasilkan nomor tiket unik secara otomatis setelah customer berhasil mengirimkan keluhan dan mengirimkan informasi tiket melalui email customer.
F-09	Pelacakan tiket mandiri oleh customer	Sistem menyediakan halaman self-service yang memungkinkan customer melihat status dan progres tiket menggunakan nomor tiket tanpa harus memiliki akun pengguna.

F-10	Notifikasi tiket baru kepada agent	Sistem mengirimkan notifikasi email secara otomatis kepada agent yang telah terhubung dengan produk terkait ketika terdapat tiket baru masuk.
F-11	Pengelolaan daftar tiket agent	Sistem menyediakan halaman daftar tiket bagi agent yang hanya menampilkan tiket sesuai tanggung jawab berdasarkan hasil pemetaan produk.
F-12	Pembaruan status dan progres tiket	Sistem memungkinkan agent memperbarui informasi tiket seperti status pengerjaan, catatan progres, dan informasi penyelesaian selama proses penanganan keluhan.
F-13	Penyelesaian tiket oleh agent	Sistem memungkinkan agent mengubah status tiket menjadi closed/resolved setelah keluhan pelanggan berhasil diselesaikan serta menyimpan riwayat perubahan status tiket.

Hasil analisis menunjukkan bahwa sistem memiliki empat aktor utama yaitu Superadmin, Client/Admin Tenant, Agent/PIC, dan Customer. Setiap aktor memiliki hak akses dan tanggung jawab yang berbeda sesuai dengan proses bisnis yang dijalankan. Kebutuhan fungsional yang dihasilkan mencakup pengelolaan tenant, pengelolaan produk, pengelolaan pengguna, pembuatan tiket keluhan, penanganan tiket, pelacakan tiket, notifikasi email, serta monitoring aktivitas sistem.

Analisis Kebutuhan Non-Fungsional

Tahap ini dilakukan untuk menentukan karakteristik kualitas yang harus dimiliki sistem.

Tabel 3. Kebutuhan Non-Fungsional Sistem

No	Kebutuhan Non Fungsional	Deskripsi Terukur
NF-01	Keamanan	Sistem menerapkan mekanisme autentikasi dan otorisasi berbasis peran (role-based access control) serta isolasi data antar tenant sehingga setiap tenant hanya dapat mengakses data miliknya sendiri.
NF-02	Skalabilitas	Sistem menggunakan arsitektur SaaS multi-tenant yang memungkinkan penambahan tenant baru tanpa perlu melakukan instalasi aplikasi atau konfigurasi infrastruktur secara terpisah.
NF-03	Ketersediaan Layanan	Sistem dapat menyediakan layanan helpdesk secara berkelanjutan melalui akses berbasis subdomain tenant dengan mekanisme penyimpanan dan pengelolaan data tiket yang terpusat.
NF-04	Kemudahan Penggunaan	Sistem menyediakan antarmuka yang mudah dipahami oleh pengguna dengan alur pelaporan keluhan yang sederhana serta fitur pelacakan tiket mandiri bagi customer menggunakan nomor tiket.

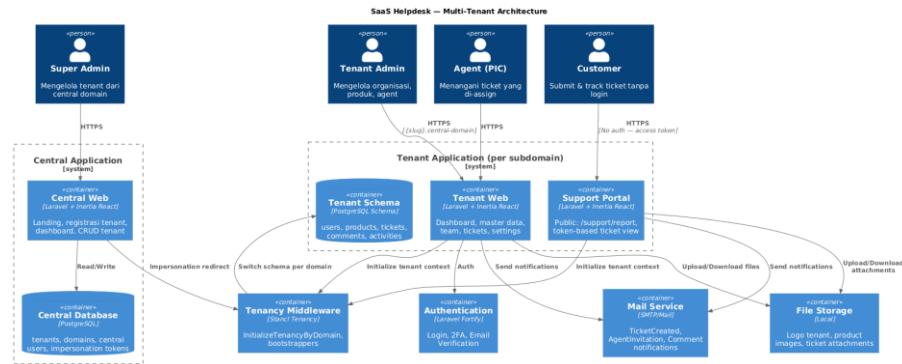
Hasil analisis menunjukkan bahwa aspek keamanan, skalabilitas, ketersediaan layanan, dan kemudahan penggunaan menjadi kebutuhan utama dalam pengembangan sistem. Oleh karena itu, sistem dirancang menggunakan arsitektur Multi-Tenant Single Database Multi-Schema yang mampu menjaga isolasi data antar tenant sekaligus mendukung efisiensi pengelolaan sumber daya aplikasi.

B. Perancangan Sistem

Hasil analisis kebutuhan sebelumnya digunakan sebagai dasar untuk memulai tahap perancangan. Tujuan dari tahap ini adalah untuk menghasilkan model sistem yang memenuhi kebutuhan pengguna dan mendukung implementasi konsep SaaS dan Multi-Tenant.

Arsitektur Sistem

Perancangan sistem menggunakan pendekatan arsitektur Software as a Service (SaaS) dengan konsep Multi-Tenant Single Database Multi-Schema.

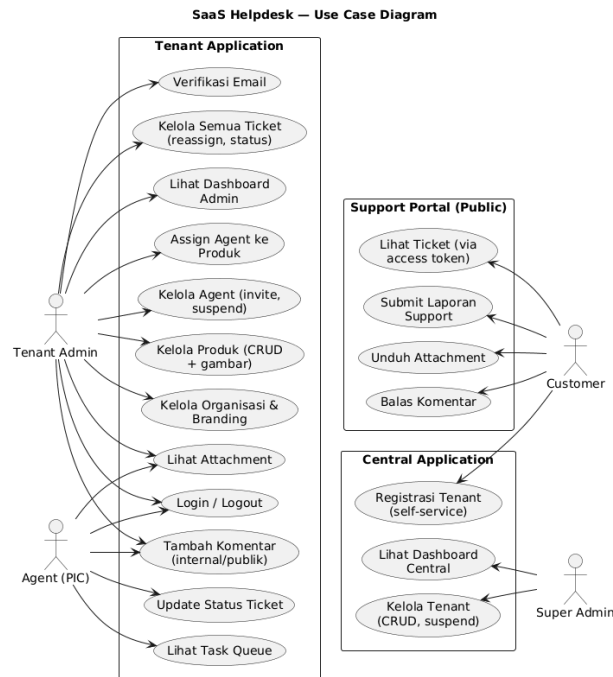


Gambar 1. Arsitektur Sistem

Arsitektur tersebut memungkinkan satu aplikasi digunakan oleh banyak organisasi secara bersamaan dengan tetap mempertahankan pemisahan data masing-masing tenant. Pada sisi central application, Superadmin bertanggung jawab terhadap pengelolaan tenant dan monitoring layanan. Sementara itu, setiap tenant memiliki schema basis data yang digunakan untuk menyimpan data operasional organisasi secara terpisah.

Use Case Diagram

Diagram yang digunakan untuk menggambarkan hubungan antara aktor dan fungsi sistem yang tersedia.

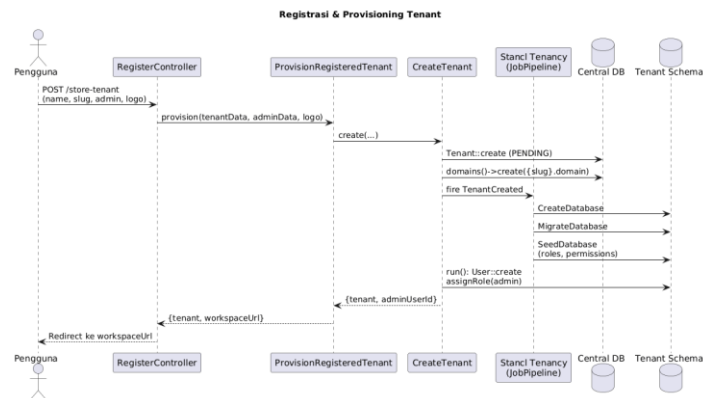


Gambar 2. Diagram Use Case

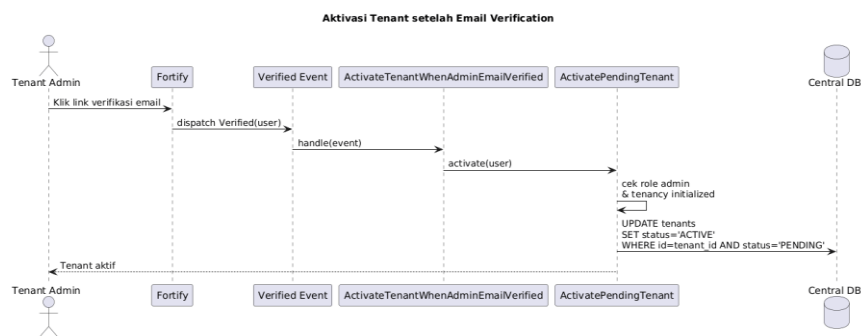
Diagram tersebut menunjukkan bahwa Customer dapat membuat dan melacak tiket, Agent/PIC dapat menangani tiket, Admin Tenant dapat mengelola operasional helpdesk, sedangkan Superadmin memiliki akses terhadap pengelolaan tenant dan monitoring sistem secara keseluruhan.

Sequence Diagram

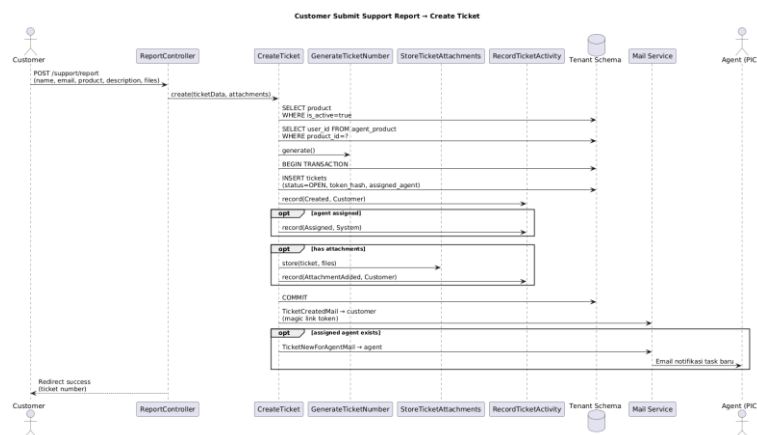
Diagram ini digunakan untuk menggambarkan cara komponen sistem berinteraksi satu sama lain dalam proses bisnis utama.



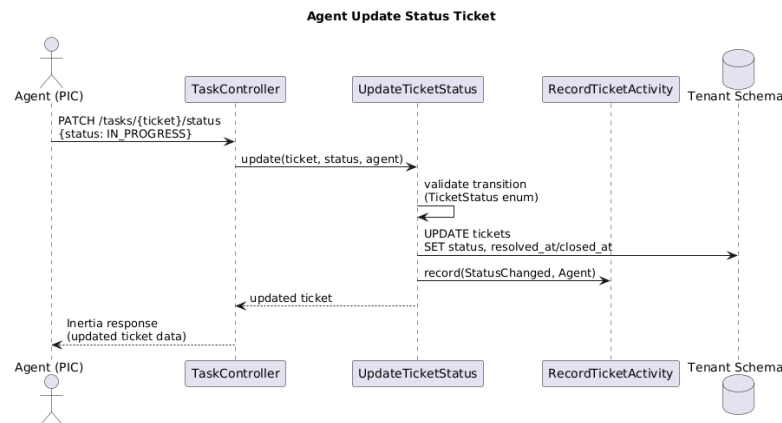
Gambar 3. Sequence Diagram - Registrasi dan Provisioning Tenant



Gambar 4. Sequence Diagram - Aktivasi Tenant



Gambar 5. Sequence Diagram - Submit Report

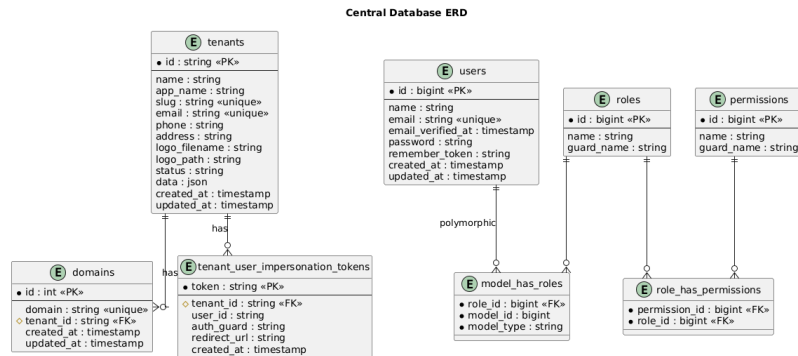


Gambar 6. Sequence Diagram - Update Status Tiket

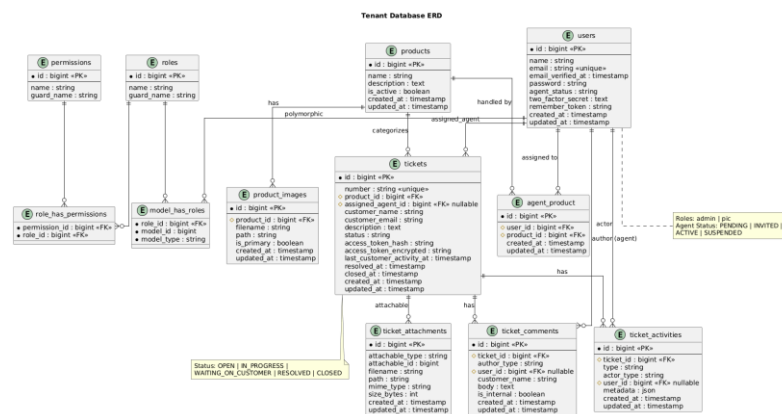
Diagram tersebut menggambarkan urutan komunikasi yang terjadi selama proses registrasi tenant, pembuatan tiket keluhan, dan penyelesaian tiket oleh Agent/PIC sehingga dapat menjadi acuan dalam tahap implementasi.

Perancangan Basis Data

Untuk memenuhi kebutuhan penyimpanan sistem, basis data dirancang.



Gambar 7. ERD Database Schema Central



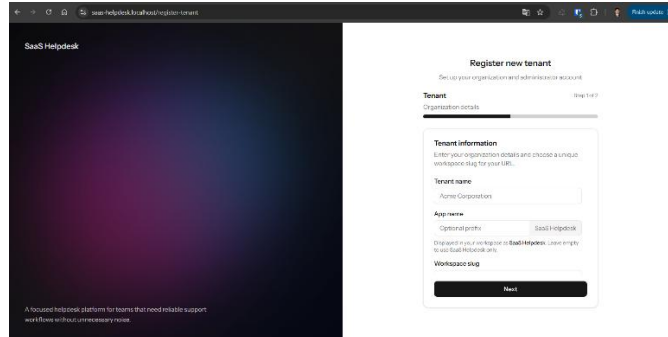
Gambar 8. ERD Database Schema Tenant

Basis data dibagi menjadi central schema dan tenant schema. Central schema digunakan untuk menyimpan informasi tenant dan data yang berkaitan dengan platform SaaS, sedangkan tenant schema digunakan untuk menyimpan data operasional masing-masing organisasi seperti produk, pengguna, tiket, komentar, dan histori aktivitas.

C. Implementasi

Tahap implementasi dimulai dengan membuat prototype sistem berdasarkan hasil perancangan yang telah dibuat. Ini dilakukan dengan menggunakan Laravel, React, Inertia.js, PostgreSQL, serta library pendukung Multi-Tenant dan manajemen hak akses.

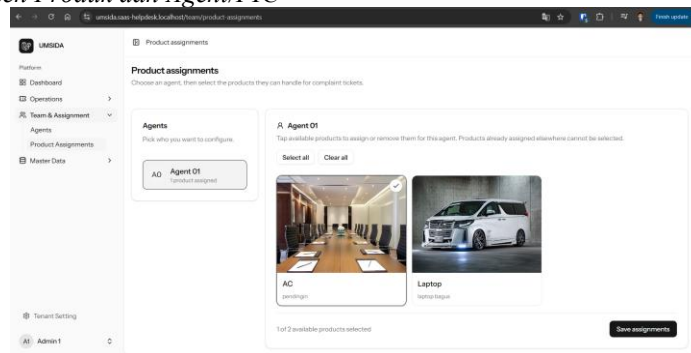
Implementasi Registrasi Tenant



Gambar 9. Halaman Registrasi Tenant

Prototype menyediakan mekanisme pelaporan keluhan yang dapat digunakan pelanggan tanpa harus melakukan registrasi akun. Pelanggan cukup mengisi identitas dasar, memilih produk yang mengalami kendala, dan menjelaskan permasalahan yang dihadapi. Setelah laporan berhasil dikirimkan, sistem secara otomatis menghasilkan nomor tiket sebagai identitas laporan.

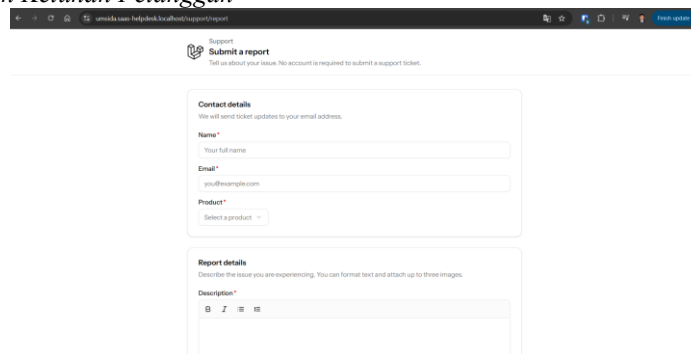
Implementasi Manajemen Produk dan Agent/PIC



Gambar 10. Halaman Asignasi Produk

Admin Tenant dapat mengelola daftar produk dan menentukan Agent/PIC yang bertanggung jawab terhadap masing-masing produk. Mekanisme ini memungkinkan distribusi tiket dilakukan secara lebih terstruktur sesuai dengan ruang lingkup tanggung jawab pengguna

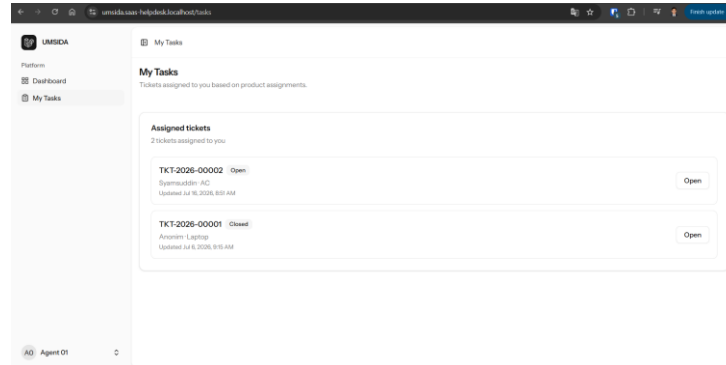
Implementasi Pelaporan Keluhan Pelanggan



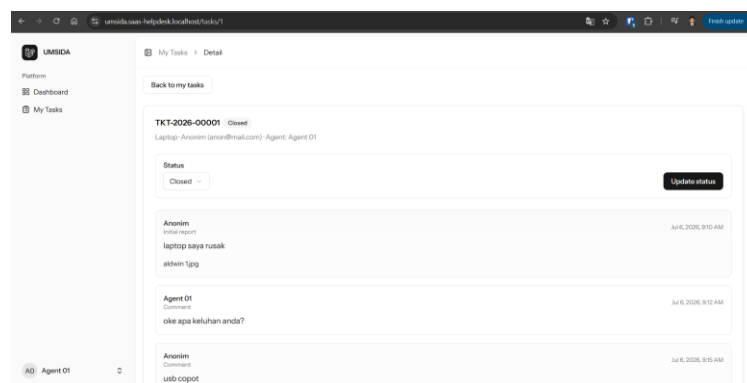
Gambar 11. Halaman Portal Pelaporan

Prototype menyediakan mekanisme pelaporan keluhan yang dapat digunakan pelanggan tanpa harus melakukan registrasi akun. Pelanggan cukup mengisi identitas dasar, memilih produk yang mengalami kendala, dan menjelaskan permasalahan yang dihadapi. Setelah laporan berhasil dikirimkan, sistem secara otomatis menghasilkan nomor tiket sebagai identitas laporan.

Implementasi Pengelolaan Tiket



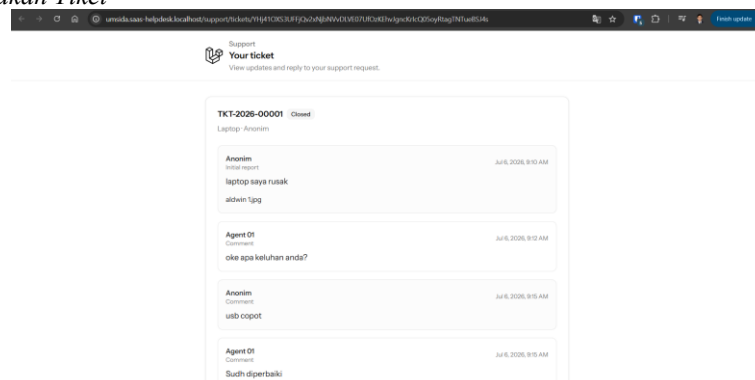
Gambar 12. Halaman My Tasks



Gambar 13. Halaman Detail Tiket – Untuk update Status dan Komen

Sistem menyediakan fitur pengelolaan tiket yang memungkinkan Agent/PIC melakukan penerimaan, pembaruan status, pemberian tanggapan, serta penyelesaian tiket. Seluruh aktivitas yang dilakukan tercatat dalam histori sehingga dapat digunakan sebagai bahan evaluasi layanan.

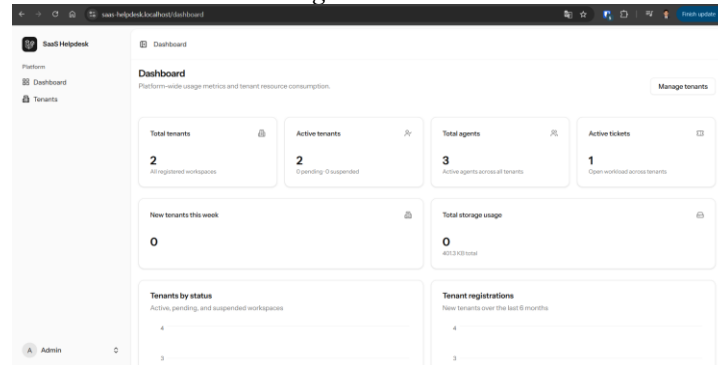
Implementasi Pelacakan Tiket



Gambar 14. Halaman Pelacakan Tiket

Sistem menyediakan fitur pelacakan tiket yang memungkinkan pelanggan memantau perkembangan laporan menggunakan nomor tiket dan alamat email yang digunakan saat pelaporan. Fitur ini meningkatkan transparansi layanan serta mengurangi kebutuhan komunikasi manual antara pelanggan dan organisasi.

Implementasi Pengelolaan Tenant dan Monitoring Sistem



Gambar 15. Halaman Dashboard Superadmin

Pada sisi penyedia layanan SaaS, sistem menyediakan fitur pengelolaan tenant, monitoring penggunaan sistem, serta impersonasi tenant untuk kebutuhan dukungan teknis. Fitur tersebut memungkinkan proses administrasi dan pemeliharaan sistem dilakukan secara terpusat.

D. Evaluasi

Tahap ini dilakukan untuk memastikan bahwa prototype yang dibuat memenuhi persyaratan pengguna dan bekerja sesuai dengan spesifikasi.

Hasil Pengujian Black Box Testing

Pengujian dilakukan dengan fokus pada fungsi sistem dari sudut pandang pengguna.

Tabel 4. Hasil Black Box Testing

No	Tes Kasus	Hasil yang diharapkan	Status
BB-01	Pendaftaran organisasi baru	Organisasi dan akun admin berhasil dibuat; data tidak valid atau duplikat ditolak	Success
BB-02	Aktivasi organisasi setelah email admin diverifikasi	Organisasi aktif hanya setelah admin memverifikasi email; status lain tidak berubah sembarangan	Success
BB-03	Masuk ke aplikasi di domain pusat dan organisasi	Pengguna bisa login di domain yang tepat; akun belum terverifikasi tidak bisa masuk beranda	Success
BB-04	Login dan keluar dari akun (sisi pusat)	Login berhasil dengan kredensial benar, gagal jika salah, dilindungi rate limit, dan logout membersihkan sesi	Success
BB-05	Konfirmasi alamat email pengguna	Email terverifikasi hanya dengan tautan valid; pengguna belum verified tidak mengakses dashboard	Success
BB-06	Mengatur ulang kata sandi	Pengguna dapat meminta dan mengatur ulang kata sandi dengan token valid; token salah ditolak	Success
BB-07	Pelaporan masalah oleh pelanggan (tanpa login)	Laporan masuk menjadi tiket, agen mendapat notifikasi, pelanggan bisa cek dan balas lewat tautan rahasia	Success
BB-09	Penanganan tiket oleh admin dan agen	Admin melihat semua tiket; agen hanya tiketnya; status, balasan, dan reassign berjalan sesuai aturan	Success
BB-08	Pengelolaan anggota tim dukungan	Admin dapat menambah, mengundang, menangguhkan, dan mengaktifkan agen; tanpa izin ditolak	Success

BB-09	Katalog produk yang didukung	Admin dapat menambah, mengubah, dan menghapus produk beserta gambar; batas dan izin ditegakkan	Success
BB-10	Penugasan produk ke masing-masing agen	Produk dapat dipetakan ke agen tanpa bentrok; produk yang sudah diambil agen lain tidak bisa digandakan	Success
BB-11	Ringkasan kinerja di beranda organisasi	Admin dan agen mendapat ringkasan sesuai perannya; filter periode mengubah angka yang relevan	Success
BB-12	Pengaturan identitas organisasi	Admin dapat mengubah nama dan logo; pengguna lain atau tamu tidak bisa mengakses	Success
BB-13	Hak akses di dalam organisasi	Peran dan izin terpasang saat organisasi dibuat; admin pertama mendapat peran yang benar	Success
BB-14	Hak akses di panel pusat	Peran pusat tersedia; super admin memiliki seluruh izin yang dibutuhkan	Success
BB-15	Menambah organisasi dari panel pusat	Operator pusat dapat membuat organisasi baru; tamu tidak diizinkan	Success
BB-16	Mengelola organisasi dari panel pusat	Operator dapat mengubah, menghapus, dan masuk sebagai admin organisasi (impersonasi)	Success
BB-17	Mengakhiri sesi masuk sebagai pengguna lain	Operator dapat kembali ke panel pusat dengan aman; tautan kembali yang tidak valid diabaikan	Success
BB-18	Ringkasan operasional di panel pusat	Operator melihat metrik dashboard; tamu tidak bisa masuk	Success
BB-19	Menjelajahi menu panel pusat	Pengguna yang sudah login dapat membuka halaman daftar organisasi; tamu ditolak	Success
BB-20	Mengubah data profil akun	Pengguna dapat memperbarui profil atau menghapus akun dengan konfirmasi kata sandi	Success
BB-21	Keamanan akun (kata sandi)	Pengguna dapat mengganti kata sandi jika kata sandi lama benar; halaman keamanan tampil sesuai pengaturan	Success
BB-22	Nama tampilan aplikasi per organisasi	Nama aplikasi mengikuti prefiks organisasi atau nama pusat jika prefiks kosong	Success

Hasil pengujian menunjukkan bahwa seluruh fungsi utama sistem dapat berjalan sesuai dengan skenario pengujian yang telah ditetapkan. Tidak ditemukan kegagalan fungsi pada proses registrasi tenant, pengelolaan produk, pengelolaan pengguna, pelaporan keluhan, penanganan tiket, maupun pelacakan tiket.

Analisis Kesesuaian Sistem terhadap Kebutuhan

Berdasarkan hasil pengujian, seluruh kebutuhan fungsional yang diperoleh pada tahap analisis berhasil diimplementasikan ke dalam prototype sistem. Selain itu, kebutuhan non-fungsional seperti isolasi data tenant, kemudahan akses pelanggan, dan pengelolaan layanan secara terpusat juga berhasil diwujudkan melalui penerapan arsitektur Multi-Tenant Single Database Multi-Schema.

E. Pembahasan

Analisis Pemenuhan Kebutuhan Pengelolaan Keluhan Pelanggan

Hasil implementasi menunjukkan bahwa sistem mampu mengatasi permasalahan utama yang ditemukan pada tahap analisis kebutuhan. Proses pelaporan keluhan yang sebelumnya dilakukan melalui berbagai media komunikasi berhasil dipusatkan dalam satu platform sehingga memudahkan proses pencatatan, distribusi, dan monitoring tiket.

Analisis Implementasi SaaS Multi-Tenant

Penerapan arsitektur SaaS Multi-Tenant memungkinkan sistem digunakan oleh banyak organisasi dalam satu platform yang sama. Pendekatan Single Database Multi-Schema memberikan keseimbangan antara efisiensi penggunaan sumber daya dan isolasi data tenant sehingga sesuai untuk implementasi layanan helpdesk berbasis SaaS.

Analisis Penggunaan Framework IBE 1.0

Penggunaan framework IBE 1.0 membantu proses identifikasi kebutuhan secara sistematis melalui CES Framework dan FNF Framework. Hasil analisis dapat digunakan sebagai dasar untuk membuat rancangan sistem yang sesuai dengan tujuan penelitian dan kebutuhan pengguna.

Perbandingan dengan Studi Sebelumnya

Tabel 5. Perbandingan Penelitian

No	Peneliti / Tahun	Topik	Helpdesk	Multi-Tenant	SaaS	Isolasi Subdomain
1	Mahesa (2024) [16]	Helpdesk Ticketing Berbasis Web di BSN	✓	✗	✗	✗
2	UNINDRA (2023) [24]	Perancangan Ticketing Helpdesk PT. Arthatech	✓	✗	✗	✗
3	Aziza dkk. (2024) [17]	Analisis Arsitektur Multi-Tenant SaaS	✗	✓	✓	✗
4	Krishna (2021) [18]	SLR SaaS Security & Multitenancy	✗	✓	✓	✗
5	Penelitian ini (2026)	SaaS Helpdesk Multi-Tenant	✓	✓	✓	✓

Berdasarkan hasil perbandingan, penelitian ini tidak hanya mengimplementasikan sistem helpdesk, tetapi juga mengintegrasikan konsep SaaS dan Multi-Tenant dalam satu platform. Kombinasi ketiga aspek tersebut menjadi pembeda utama dibandingkan penelitian terdahulu yang umumnya hanya berfokus pada salah satu aspek.

Kontribusi Penelitian

Kontribusi utama penelitian ini adalah menghasilkan prototype sistem helpdesk berbasis SaaS dengan arsitektur Multi-Tenant yang mendukung pengelolaan keluhan pelanggan secara terpusat, terstruktur, dan scalable. Selain itu, penelitian ini menunjukkan bahwa framework IBE 1.0 dapat digunakan sebagai pendekatan analisis kebutuhan yang efektif dalam pengembangan sistem berbasis layanan (*service-oriented system*) dengan karakteristik multi-organisasi.

VII. SIMPULAN

Penelitian ini berhasil merancang dan membangun sistem helpdesk berbasis *Software as a Service* (SaaS) dengan arsitektur Multi-Tenant menggunakan pendekatan Single Database Multi-Schema untuk mendukung pengelolaan keluhan pelanggan pada berbagai organisasi dalam satu platform terpusat. Hasil analisis kebutuhan menggunakan framework IBE 1.0 yang terdiri dari CES Framework dan FNF Framework menghasilkan kebutuhan sistem yang selanjutnya digunakan sebagai dasar dalam proses perancangan dan pengembangan prototype.

Implementasi sistem menunjukkan bahwa layanan pengelolaan keluhan pelanggan dapat dilakukan secara lebih terstruktur melalui mekanisme ticketing, pengelolaan produk, pengelolaan Agent/PIC, pelacakan status tiket, notifikasi email, serta monitoring aktivitas layanan. Penerapan arsitektur Multi-Tenant memungkinkan setiap organisasi menggunakan sistem yang sama dengan tetap mempertahankan isolasi data antar tenant, sehingga mendukung efisiensi pengelolaan aplikasi tanpa mengurangi aspek keamanan data.

Hasil pengujian *Black Box Testing* menunjukkan bahwa seluruh fungsi utama sistem berjalan sesuai dengan kebutuhan yang telah ditentukan. Dengan demikian, tujuan penelitian untuk menghasilkan sistem helpdesk berbasis SaaS dengan arsitektur Multi-Tenant yang mampu mendukung pengelolaan keluhan pelanggan secara terpusat, terstruktur, dan scalable telah tercapai.

Penelitian selanjutnya dapat difokuskan pada implementasi mekanisme otomatisasi distribusi tiket menggunakan pendekatan kecerdasan buatan atau machine learning, integrasi dengan berbagai kanal komunikasi seperti WhatsApp dan media sosial, serta pengukuran kualitas layanan menggunakan metode *User Acceptance Testing* (UAT) dan model evaluasi kepuasan pengguna yang lebih komprehensif.

REFERENSI

- [1] A. H. Nurhikmah, "Upaya transformasi digital dalam meningkatkan kepuasan nasabah melalui strategi pemasaran online (studi pada generasi milenial)," *Jurnal Media Akademik*, vol. 5, no. 2, pp. 45–56, Nov. 2024.
- [2] R. A. M. Ramadhani, "Pengaruh transformasi layanan digital dan kualitas pelayanan terhadap kepuasan pelanggan pada PT PLN (Persero) Unit Induk Distribusi Sumatera Selatan, Jambi dan Bengkulu," Skripsi, Fakultas Ekonomi, Universitas Sriwijaya, Palembang, Indonesia, 2025.
- [3] D. G. Siagian, A. Siregar, dan R. Simanjuntak, "Peningkatan kualitas pelayanan pelanggan melalui digitalisasi sistem penanganan keluhan di PT PLN (Persero) ULP Helvetia Medan," *Al-Khidma: Jurnal Manajemen Pelayanan Publik*, vol. 3, no. 1, pp. 11–22, 2025.
- [4] S. Manurung dan L. Simbolon, "Peran komunikasi digital dalam peningkatan kualitas pelayanan," *Jurnal Loyalitas Mahasiswa Indonesia (JLMI)*, vol. 2, no. 3, pp. 77–86, 2024.
- [5] A. Kusuma dan L. Yulianto, "Sistem monitoring dan pengelolaan data keluhan pelanggan berbasis website," *Pustaka AI: Jurnal Sistem Informasi dan Sains Data*, vol. 4, no. 1, pp. 23–34, 2024.
- [6] A. B. Kusuma dan L. Yulianto, "Sistem informasi keluhan pelanggan berbasis website," *J-Com (Journal of Computer)*, vol. 3, no. 1, pp. 15–24, Mar. 2021.
- [7] I. K. Imansyah, R. R. Setiawan, dan D. S. Sudrajat, "Rancang bangun aplikasi layanan pengaduan pelanggan berbasis web: studi kasus Imasnet Darmaraja," *Jurnal Ilmu Manajemen dan Teknologi (JIMT)*, vol. 2, no. 1, pp. 31–42, Jan. 2025.
- [8] R. A. Pratama, "Sistem informasi e-complaint berbasis website pada PT XYZ," Skripsi, Program Studi Teknik Informatika, Universitas Bina Darma, Palembang, Indonesia, 2024.
- [9] C. Mery, "Rancang bangun sistem informasi helpdesk," *Jurnal Citra Sistem Informasi*, vol. 4, no. 2, pp. 89–98, Jul. 2020.
- [10] H. S. Prasetyo dan E. P. Sari, "Sistem informasi helpdesk dalam tata kelola teknologi informasi pada Diskominfo dan SP," *Jurnal Teknologi Informasi (JuTI)*, vol. 7, no. 3, pp. 101–112, Aug. 2023.
- [11] S. Gupta dan R. Kumar, "Multi-tenant architectures in modern cloud computing," *International Journal of Scientific Research in Computer Science, Engineering and Information Technology*, vol. 10, no. 1, pp. 236–245, 2025.
- [12] S. Gupta dan R. Kumar, "Multi-tenant architecture: a comprehensive framework for building scalable SaaS applications," *International Journal of Scientific Research in Computer Science, Engineering and Information Technology*, vol. 9, no. 6, pp. 1151–1162, 2024.
- [13] N. Hattab, "Modular models for systems based on multi-tenant services: a multi-level Petri-net-based approach," *Journal of King Saud University – Computer and Information Sciences*, vol. 35, no. 6, pp. 854–869, 2023.
- [14] M. A. Khan dan S. R. Patel, "Multi-tenant SaaS architectures: design principles and security considerations," *Quest Journals: Journal of Software Engineering and Simulation*, vol. 6, no. 5, pp. 28–41, 2023.
- [15] J. Smith dan L. Brown, "SaaS modernization 2026: re-architecting legacy products for multi-tenant environments," *Cloud Architecture Laboratory, Technical Report*, 2026.
- [16] M. R. Mahesa, "Pengembangan Aplikasi Helpdesk Ticketing System Berbasis Web di Badan Standardisasi Nasional," Skripsi, STT Terpadu Nurul Fikri, 2024. [Online]. Available: <https://repository.nurulfikri.ac.id/eprint/536/>
- [17] Aziza et al., "Analisis Trade-Off Arsitektur Multi-Tenant Sebagai Dasar Justifikasi Strategi Basis Data SaaS," *Progresif: Jurnal Ilmiah Komputer*, STMIK Banjarbaru, 2024. [Online]. Available: <https://ojs.stmik-banjarbaru.ac.id/index.php/progresif/article/view/3590>
- [18] A. Krishna, "Systematic Literature Review of Software as a Service (SaaS) in View of Security and Multitenancy," *Int. J. Res. Appl. Sci. Eng. Technol.*, vol. 9, no. 9, pp. 1742–1751, 2021, doi: 10.22214/ijraset.2021.38205.
- [19] Sugiyono, *Metode Penelitian Kuantitatif, Kualitatif, dan R&D*, Edisi ke-2. Bandung: Alfabeta, 2022.
- [20] I. A. Kautsar and M. Ruslianor Maika, "The use of User-centered Design Canvas for Rapid Prototyping," *J. Phys. Conf. Ser.*, vol. 1764, no. 1, p. 012175, Feb. 2021, doi: 10.1088/1742-6596/1764/1/012175.

- [21] I. A. Kautsar and M. R. Maika, "Platform Design Canvas Adaptation for Rapid Prototyping and Project-based Learning amid Covid-19 Pandemic," in *2022 IEEE World Engineering Education Conference (EDUNINE)*, Santos, Brazil: IEEE, Mar. 2022, pp. 1–6. doi: 10.1109/EDUNINE53672.2022.9782390.
- [22] I. A. Kautsar, M. H. Hamdi, R. S. H. F. Aziz, and M. R. Maika, "User-centered Approach and Low-Code Framework for Prototyping and Income-based Education," in *2023 7th International Conference on New Media Studies (CONMEDIA)*, Bali, Indonesia: IEEE, Dec. 2023, pp. 1–6. doi: 10.1109/CONMEDIA60526.2023.10428254.
- [23] I. A. Kautsar, M. R. Maika, A. N. Budiman, A. B. Setyawan, and J. Y. Awali, "Microservice Based Architecture: The Development of Rapid Prototyping Supportive Tools for Project Based Learning," in *2023 IEEE World Engineering Education Conference (EDUNINE)*, Bogota, Colombia: IEEE, Mar. 2023, pp. 1–6. doi: 10.1109/EDUNINE57531.2023.10102884
- [24] Tim Penulis UNINDRA, "Perancangan Sistem Ticketing Helpdesk pada PT. Arthatech Selaras," JRKT Universitas Indraprasta, 2023. [Online]. Available: <https://jim.unindra.ac.id/index.php/JRKT/article/download/6485/849>

Conflict of Interest Statement:

The author declares that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.