

Pengembangan Sistem Backoffice Manajemen Kos Berbasis Web Menggunakan Next.js dan Laravel REST API

Dhani Cahyo Kurnia Afandy^{*1}, Ramadhan Renaldy²

¹Informatika, Universitas PGRI Semarang, Kota Semarang, dhaniaja431@gmail.com

²Informatika, Universitas PGRI Semarang, Kota Semarang, ramadhanrenaldy@upgris.ac.id

Abstract

The increasing demand for boarding house (kos) rentals near campuses and business districts has not been matched by adequate management systems, as most operators still rely on manual record-keeping for tenant data, payment tracking, and room availability monitoring. This condition often leads to disorganized records, missed payment deadlines, and inefficient operational reporting. This study aims to design and develop a web-based backoffice management system for boarding houses, named Kost Manager, that integrates property management, tenant and contract administration, payment processing, and financial reporting into a single platform. The system was developed using the Prototype method, allowing iterative refinement based on user feedback, with Next.js used for the frontend and Laravel REST API for the backend within a monorepo architecture. Token-based authentication using Laravel Sanctum and Role-Based Access Control (RBAC) were implemented to restrict access between Admin and Staff roles. The results show that the system successfully manages rooms, room types, facilities, tenants, contracts, payments, expenses, documents, and financial and occupancy reports, with partial payment handling and contract renewal tracking. Testing results confirm that all major features function according to user requirements. Therefore, the developed system can serve as an integrated and structured solution to improve the efficiency of boarding house management operations.

Keywords: Backoffice, Boarding House Management, Next.js, Laravel REST API, Prototype

Abstrak

Tingginya permintaan hunian kos di sekitar kawasan kampus dan perkantoran belum diimbangi dengan sistem pengelolaan yang memadai, karena sebagian besar pengelola masih bergantung pada pencatatan manual untuk data penyewa, pemantauan pembayaran, dan ketersediaan kamar. Kondisi ini menyebabkan data tidak terorganisir, keterlambatan pemantauan tenggat pembayaran, serta pelaporan operasional yang kurang efisien. Penelitian ini bertujuan merancang dan mengembangkan sistem backoffice manajemen kos berbasis web bernama Kost Manager, yang mengintegrasikan pengelolaan properti, administrasi penyewa dan kontrak, pemrosesan pembayaran, serta pelaporan keuangan dalam satu platform terpusat. Sistem dikembangkan menggunakan metode Prototype yang memungkinkan penyempurnaan secara iteratif berdasarkan umpan balik pengguna, dengan Next.js sebagai frontend dan Laravel REST API sebagai backend dalam arsitektur monorepo. Autentikasi berbasis token menggunakan Laravel Sanctum dan Role-Based Access Control (RBAC) diterapkan untuk membatasi hak akses antara peran Admin dan Staff. Hasil penelitian menunjukkan bahwa sistem mampu mengelola kamar, tipe kamar, fasilitas, penyewa, kontrak, pembayaran, pengeluaran, dokumen, serta laporan keuangan dan okupansi, dengan dukungan pembayaran bertahap dan pelacakan riwayat perpanjangan kontrak. Hasil pengujian menunjukkan bahwa seluruh fitur utama berjalan sesuai kebutuhan pengguna. Dengan demikian, sistem yang dikembangkan dapat menjadi solusi terintegrasi dan terstruktur untuk meningkatkan efisiensi pengelolaan operasional kos.

Keywords: Backoffice, Manajemen Kos, Next.js, Laravel REST API, Prototype.

1. Pendahuluan

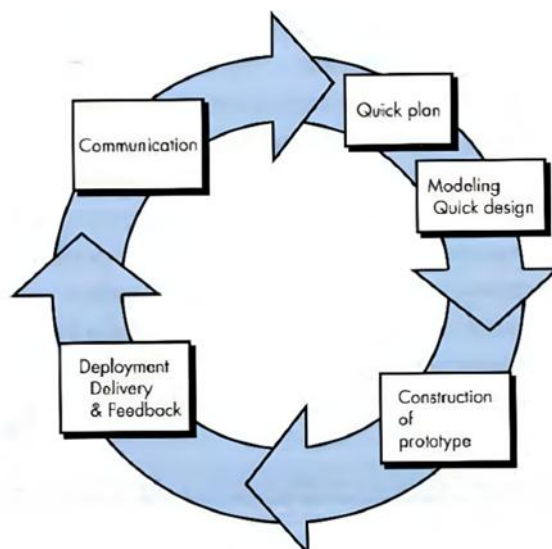
Perkembangan teknologi informasi mendorong berbagai sektor usaha beralih dari sistem pengelolaan konvensional menuju sistem berbasis digital, termasuk bisnis kos. Usaha kos yang berada di sekitar kawasan kampus, perkantoran, dan pusat aktivitas lainnya memberikan kemudahan bagi mahasiswa dan pekerja dari luar daerah yang membutuhkan hunian sementara, sehingga permintaan terhadap layanan ini terus meningkat dari tahun ke tahun [1]. Namun, banyak pengelola kos yang masih mengandalkan catatan manual yang sering menyebabkan berbagai masalah, seperti data penghuni yang tidak teratur, pencatatan pembayaran yang tidak sistematis, dan pengawasan kondisi kamar yang sulit dilakukan [2]. Ketergantungan pada cara manual juga membuat pengelola mudah lupa dengan batas waktu pembayaran atau masa berakhir kontrak penyewa karena tidak ada pemberitahuan otomatis, yang dapat menyebabkan kerugian uang dan menurunkan mutu layanan [3]. Pelaksanaan sistem informasi yang berbasis web dianggap mampu menyelesaikan masalah tersebut karena memungkinkan proses pencatatan, pengaturan, dan pelaporan data dilakukan dengan cara terpusat dan fleksibel [4], [5].

Berbagai penelitian terdahulu telah mencoba untuk menemukan solusi terkait pengelolaan kos secara digital. Yulianto dan Prasetya telah menciptakan sistem penyewaan kamar kos yang menggunakan web, yang meliputi pemantauan ketersediaan kamar, pelacakan pembayaran, serta pengelolaan data penyewa dengan cara yang menyeluruh [5]. Purnia dkk. menerapkan metode Prototyping dalam pengembangan sistem informasi kost berbasis mobile yang terbukti efektif mendukung interaksi antara pemilik dan calon penyewa kos [6]. Namun, kedua studi itu masih hanya memusatkan perhatian pada salah satu sisi operasional, yaitu pada pemantauan ketersediaan kamar atau interaksi dasar antara pemilik dan penyewa, dan belum menggabungkan elemen manajemen kontrak, pengeluaran operasional, pelaporan keuangan dan okupansi, serta pembagian hak akses staf ke dalam satu platform terpusat. Kesenjangan yang sama juga terlihat pada penelitian lainnya di area manajemen kos yang berbasis web [1], [2] maupun berbasis mobile [3], yang umumnya belum mencakup seluruh siklus operasional kos dalam satu sistem yang terintegrasi.

Berdasarkan perbedaan yang ada, studi ini merancang dan membuat sistem manajemen kos yang berbasis web yang dinamakan Kost Manager. Sistem ini menggabungkan pengelolaan properti, administrasi penyewa dan kontrak, pengolahan pembayaran, pencatatan pengeluaran, serta laporan keuangan dalam satu platform yang terpusat. Sistem ini dibuat menggunakan Next.js sebagai antarmuka pengguna (frontend) dan Laravel REST API sebagai backend, dengan struktur yang memisahkan frontend dan backend, sehingga mendukung pengembangan yang modul dan pemeliharaan yang Panjang [7], [8]. Autentikasi yang menggunakan token dengan Laravel Sanctum dan model Pengendalian Akses Berdasarkan Peran (RBAC) diterapkan untuk membedakan hak akses antara Admin dan Staff [9], [10]. Pengembangan dilakukan dengan metode Prototype yang memungkinkan perbaikan sistem secara bertahap berdasarkan masukan dari pengguna [11]. Keunikan dari penelitian ini terletak pada integrasi menyeluruh dari semua aspek operasional kos, termasuk pembayaran bertahap dan pelacakan riwayat perpanjangan kontrak, yang belum ada di sistem-sistem lain yang dibahas dalam penelitian sebelumnya.

2. Metode

Penelitian ini dilakukan untuk merancang dan menciptakan sistem manajemen kos berbasis web yang mengumpulkan semua fungsi seperti pengelolaan properti, penyewa, kontrak, pembayaran, dan laporan keuangan dalam satu platform terpusat. Informasi dan kebutuhan sistem dikumpulkan melalui diskusi dengan pembimbing di Vrasmedia IT Solution serta pengamatan terhadap cara kerja pengelolaan kos yang sedang berlangsung. Pengembangan perangkat lunak dilakukan dengan metode Prototype karena cara ini sangat fleksibel dalam menghadapi perubahan kebutuhan yang belum sepenuhnya jelas sejak awal [11], sehingga dianggap lebih cocok dibandingkan model bertahap seperti Waterfall untuk konteks pengembangan sistem ini [6].



Gambar 2.14 Metode Pengembangan Sistem Prototype

Seperti yang ditunjukkan pada Gambar 2.1, proses pengembangan sistem menggunakan metode Prototype bersifat melingkar dan terdiri dari lima tahapan utama yang dilakukan secara berulang, yaitu Communication, Quick Plan, Modeling Quick Design, Construction of Prototype, serta Deployment, Delivery & Feedback [11]. Kelima tahap ini dijalankan secara berulang hingga prototipe yang dihasilkan memenuhi semua kebutuhan yang telah disepakati. Dalam penelitian ini, siklus iterasi dilakukan lebih dari satu kali; iterasi pertama digunakan untuk membangun dasar sistem seperti autentikasi dan pengelolaan properti dasar, sementara iterasi berikutnya digunakan untuk memperbaiki fitur yang lebih rumit seperti sistem pembayaran dan validasi pemesanan ganda. Rincian setiap tahap dijelaskan sebagai berikut.

i. Communication

Tahap ini merupakan tahap awal berupa komunikasi dan diskusi antara peneliti dengan pembimbing di Vrasmedia IT Solution, disertai pengamatan terhadap proses bisnis pengelolaan kos yang sedang berjalan. Dari tahap ini diperoleh kebutuhan sistem yang dibagi menjadi dua kategori. Kebutuhan fungsional meliputi pengelolaan aset, penyewa, kontrak, pembayaran, biaya, dokumen, serta laporan keuangan dan okupansi. Kebutuhan non-fungsional mencakup keamanan akses menggunakan token, pembagian hak akses berdasarkan peran (Admin dan Staff), kecepatan respons antarmuka, serta kompatibilitas dengan browser web yang terbaru.

j. Quick Plan

Berdasarkan kebutuhan yang telah diidentifikasi pada tahap Communication, disusun rencana pengembangan secara cepat berupa penentuan ruang lingkup dan prioritas fitur pada setiap iterasi. Iterasi pertama difokuskan pada fitur dasar seperti autentikasi dan pengelolaan properti, sedangkan iterasi berikutnya diprioritaskan untuk fitur yang lebih kompleks seperti sistem pembayaran bertahap dan validasi pemesanan ganda.

k. Modeling Quick Design

Struktur dan proses kerja sistem dirancang menggunakan UML (Unified Modeling Language), yang meliputi diagram use case untuk menggambarkan kebutuhan fungsional, dan diagram kelas untuk menggambarkan penyusunan data sistem [9], [12]. Desain basis data relasional juga dilakukan dengan menerapkan arsitektur multi-tenant, di mana setiap entitas data dipisahkan berdasarkan profil perusahaan yang mengelola, sebagai panduan dalam proses pelaksanaan [13].

l. Construction of Prototype

Sistem dibangun dengan memisahkan sisi frontend dan backend secara penuh, sesuai dengan hasil pemodelan pada tahap sebelumnya. Backend dikembangkan

menggunakan Laravel yang menerapkan arsitektur Model-View-Controller (MVC) untuk mendukung pengembangan API yang terstruktur dan efisien [7], sedangkan sisi frontend dikembangkan menggunakan Next.js yang menyediakan sistem routing otomatis berdasarkan struktur folder, serta pengambilan data asinkron untuk mendukung aplikasi satu halaman (Single Page Application) [8]. Kedua sisi berkomunikasi melalui REST API, dengan autentikasi berbasis token menggunakan Laravel Sanctum untuk melindungi endpoint yang bersifat privat [9], serta penerapan model Role-Based Access Control (RBAC) untuk membedakan hak akses antara peran Admin dan Staff [10], [14].

m. Deployment, Delivery & Feedback

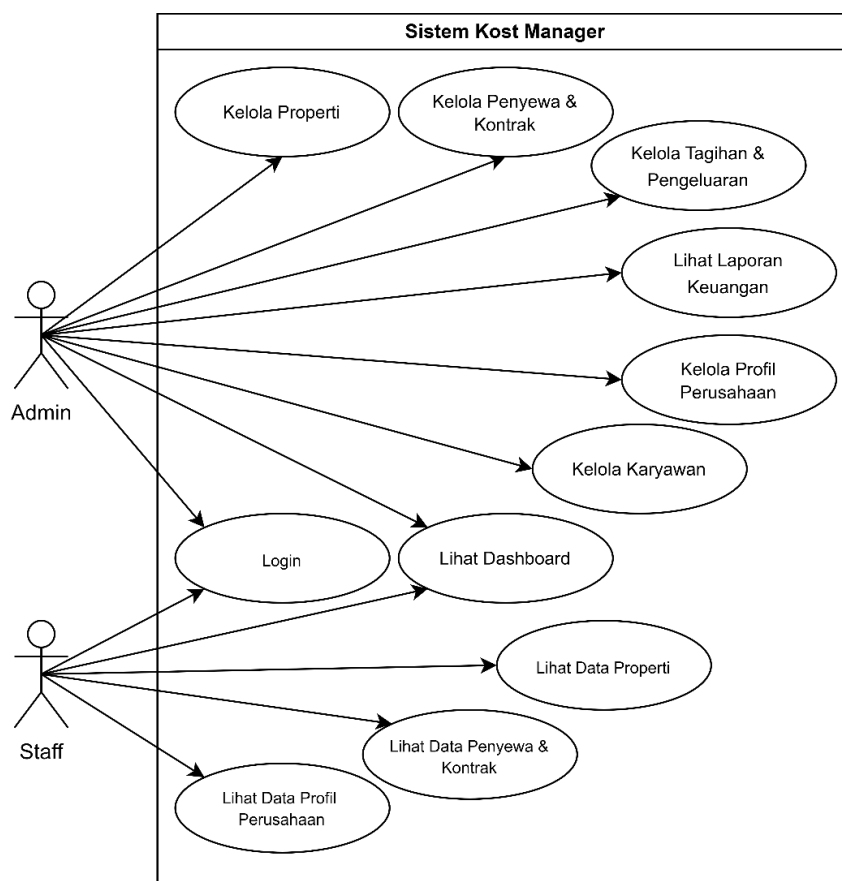
Prototipe yang telah dibangun diserahkan kepada pembimbing dan calon pengguna untuk diujicobakan secara langsung. Pengujian dilakukan menggunakan metode black-box terhadap fungsi utama sistem, meliputi autentikasi, pengelolaan properti, penyewa, kontrak, pembayaran, pengeluaran, dokumen, serta laporan keuangan dan okupansi, untuk memastikan setiap fitur berjalan sesuai kebutuhan pengguna dan spesifikasi yang telah ditentukan. Umpan balik yang diperoleh dari proses ini digunakan sebagai dasar perbaikan pada iterasi berikutnya, hingga prototipe dianggap memenuhi seluruh kebutuhan yang telah disepakati.

3. Hasil dan Pembahasan

3.1. Penyajian Hasil

Sistem backoffice manajemen kos yang dikembangkan dirancang untuk mendukung pengelolaan operasional kos secara terstruktur dan terpusat. Sistem melibatkan dua aktor utama, yaitu Admin dan Staff. Admin memiliki akses penuh terhadap seluruh fitur pengelolaan, sedangkan Staff memiliki akses baca terbatas pada modul-modul tertentu sesuai dengan kebutuhan operasionalnya.

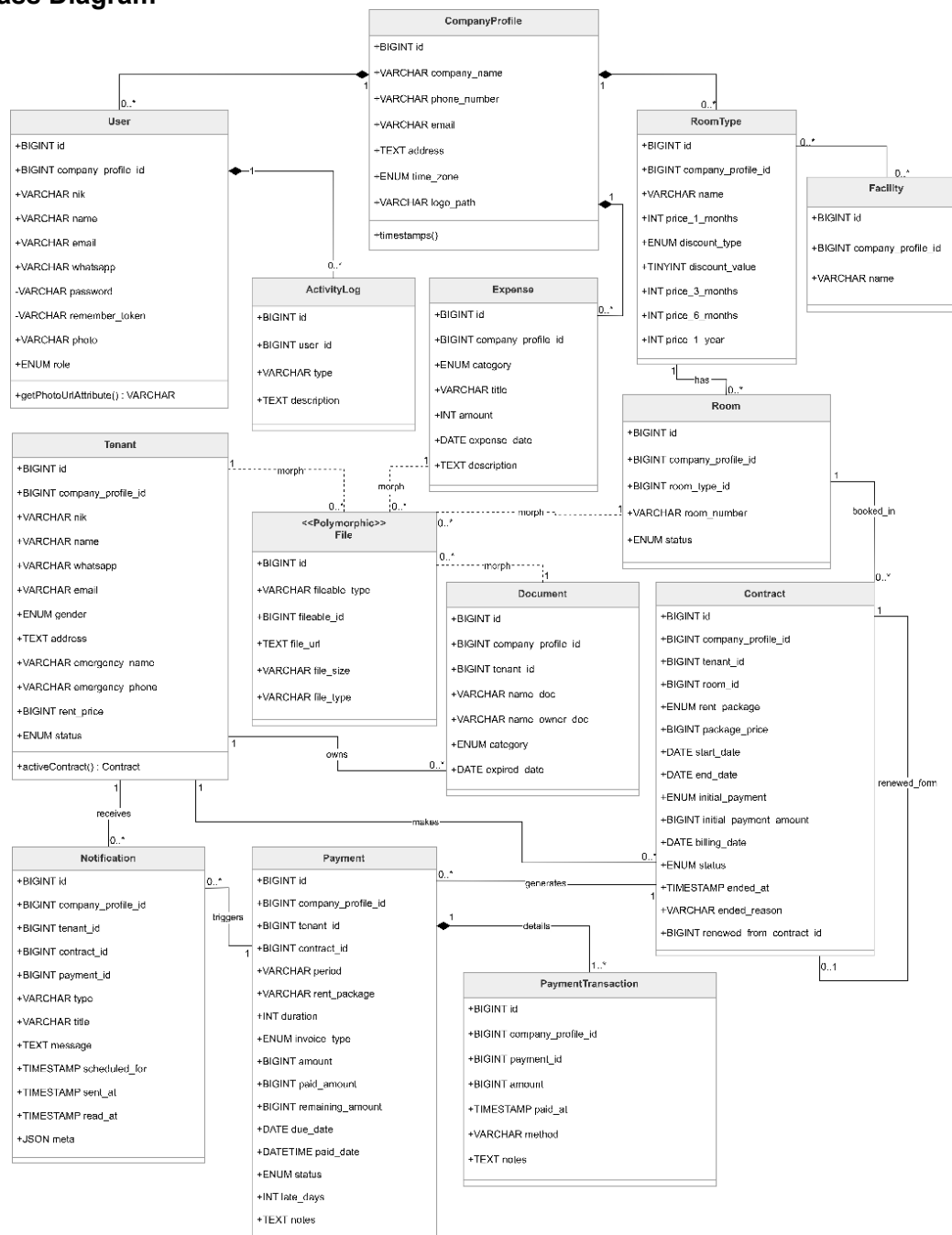
3.1.17 Use Case Diagram



Gambar 3.65 Use Case Diagram

Gambar 3.1 menggambarkan interaksi antara aktor dengan fungsionalitas yang tersedia pada sistem. Admin memiliki akses terhadap delapan use case utama, yaitu Login, Lihat Dashboard, Kelola Properti, Kelola Penyewa & Kontrak, Kelola Tagihan & Pengeluaran, Lihat Laporan Keuangan, Kelola Profil Perusahaan, dan Kelola Karyawan. Staff memiliki akses yang lebih terbatas, yaitu Login, Lihat Dashboard, Lihat Data Properti, Lihat Data Penyewa & Kontrak, serta Lihat Data Profil Perusahaan, tanpa kewenangan untuk melakukan perubahan data. Pembagian use case ini mencerminkan penerapan kontrol akses berbasis peran yang membedakan kewenangan operasional antara kedua aktor.

3.2. Class Diagram

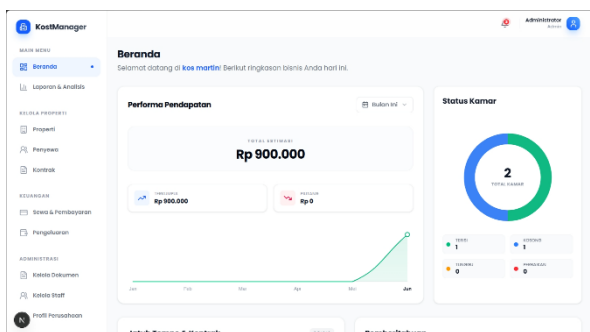


Gambar 3.66 Class Diagram

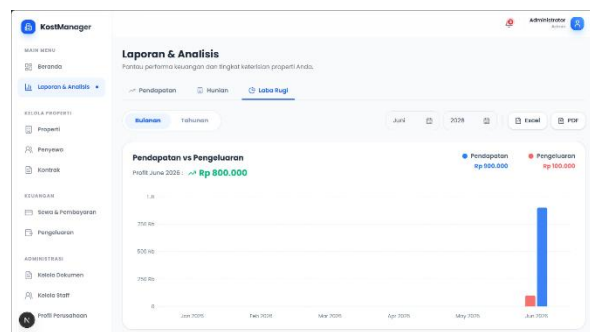
Gambar 3.2 menggambarkan struktur data dan relasi antarentitas yang digunakan dalam sistem, yang terdiri atas 14 kelas utama. Kelas CompanyProfile berperan sebagai

entitas pusat yang merepresentasikan profil perusahaan pengelola kos, dengan hampir seluruh kelas lain terhubung kepadanya untuk mendukung arsitektur multi-tenant. Kelas User merepresentasikan pengguna sistem yang dapat melakukan proses login, dengan atribut peran (role) bertipe ENUM yang membedakan kewenangan Admin dan Staff sesuai penerapan Role-Based Access Control (RBAC). Kelompok kelas properti, yaitu RoomType dan Room, mengelola data tipe kamar dan unit kamar yang tersedia. Kelompok kelas administrasi penyewa, yaitu Tenant, Contract, dan Document, mengelola data penyewa, perjanjian sewa, serta dokumen pendukung, dengan relasi self-referencing pada Contract untuk melacak riwayat perpanjangan kontrak. Kelompok kelas keuangan, yaitu Payment, PaymentTransaction, dan Expense, mendukung mekanisme tagihan, pencatatan transaksi pembayaran secara bertahap, serta pengeluaran operasional. Kelas File menerapkan relasi polimorfik yang memungkinkan satu struktur data digunakan oleh berbagai entitas, seperti Room, Tenant, Document, dan Expense, untuk menyimpan berkas digital, sementara kelas Notification mendukung fungsi pengingat tagihan kepada pengguna.

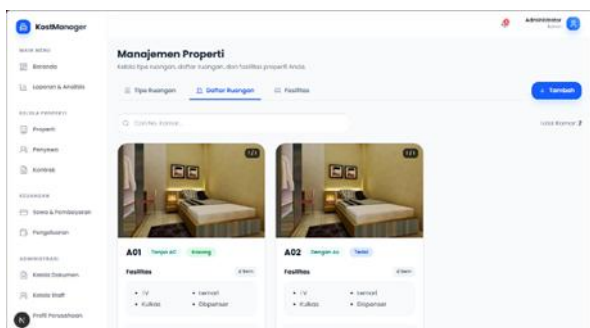
3.1.18 Implementasi Sistem



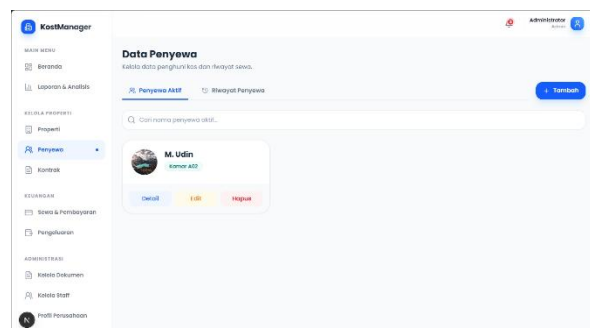
Gambar 3.67 Halaman Beranda



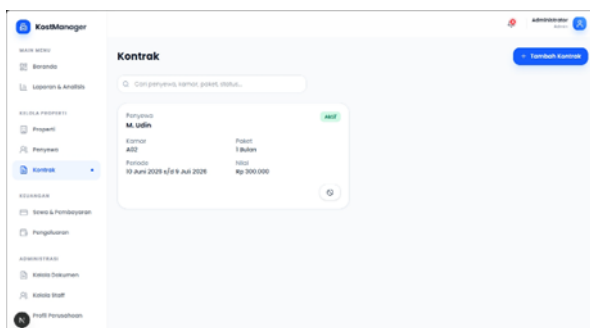
Gambar 3.68 Halaman Laporan & Analisis



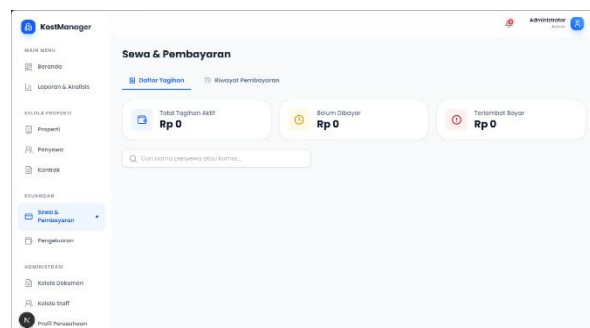
Gambar 3.69 Halaman Properti



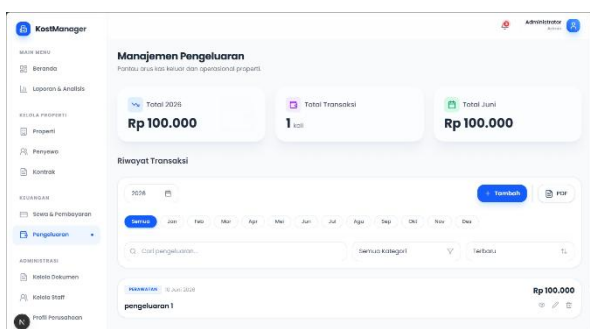
Gambar 3.70 Halaman Penyewa



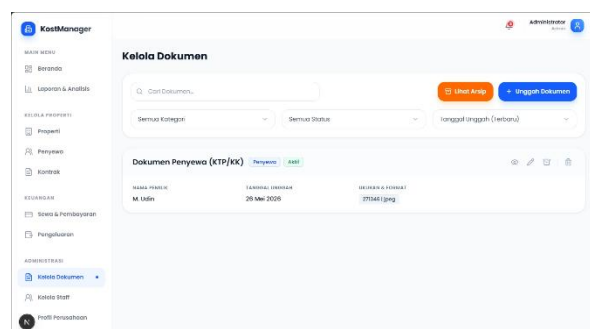
Gambar 3.71 Halaman Kontrak



Gambar 3.72 Halaman Sewa & Pembayaran



Gambar 3.73 Halaman Pengeluaran



Gambar 3.74 Halaman Kelola Dokumen

Gambar 3.3–3.10 menunjukkan penerapan fungsi utama dari sistem. Di halaman Beranda terdapat ringkasan informasi operasional seperti perkiraan pendapatan, jumlah kamar yang tersedia, dan waktu jatuh tempo kontrak. Halaman Laporan & Analisis menyajikan laporan tentang pendapatan, hunian, dan laba rugi lengkap dengan grafik serta fitur untuk mengekspor ke PDF. Halaman Properti digunakan untuk mengatur kamar dan fasilitas yang ada, sedangkan halaman Penyewa digunakan untuk mengelola informasi penghuni. Halaman Kontrak mencatat detail perjanjian sewa, sementara halaman Sewa & Pembayaran digunakan untuk memantau tagihan dan status pembayaran. Halaman Pengeluaran berfungsi untuk mencatat seluruh biaya

operasional, dan halaman Kelola Dokumen berperan sebagai tempat menyimpan berkas administratif penyewa. Dari hasil penerapan, semua fitur utama sistem telah berfungsi sesuai dengan kebutuhan pengguna dan mendukung pengelolaan operasional kos dengan baik.

3.3. Pembahasan

Hasil dari penerapan menunjukkan bahwa sistem manajemen kos yang dibuat dapat menyatukan semua aspek operasional kos dalam satu platform yang terpusat. Ini termasuk pengelolaan properti, administrasi sewa dan kontrak, pemrosesan pembayaran, pencatatan pengeluaran, serta pelaporan keuangan dan okupansi. Dengan adanya integrasi ini, para pengelola dapat lebih mudah memantau kondisi operasional secara menyeluruh tanpa harus berpindah-pindah antara berbagai sistem atau dokumen yang terpisah.

Penelitian ini memiliki kesamaan dengan penelitian yang dilakukan oleh Yulianto dan Prasetya [5], yang mengembangkan sistem sewa kamar berbasis web untuk mendukung pemantauan ketersediaan kamar dan pelacakan pembayaran. Demikian pula, penelitian Purnia dkk [6] juga telah memperlihatkan bahwa penerapan metode Prototyping pada sistem informasi kost berbasis mobile mampu mendukung proses interaksi antara pemilik dan penyewa secara lebih efisien. Namun, kedua penelitian tersebut masih berfokus pada satu sisi operasional, yaitu pemantauan ketersediaan kamar atau interaksi dasar pemilik-penyewa, dan belum mengintegrasikan aspek manajemen kontrak, pengeluaran operasional, serta pelaporan keuangan dan okupansi ke dalam satu sistem yang sama. Sistem yang dikembangkan dalam penelitian ini memperluas cakupan tersebut dengan menambahkan modul manajemen kontrak yang dapat membantu dalam melacak riwayat perpanjangan sewa, mencatat pengeluaran operasional, serta menggabungkan laporan laba rugi dan okupansi secara otomatis berdasarkan data pembayaran dan pengeluaran, sejalan dengan pendekatan otomatisasi pelaporan keuangan berbasis web yang juga terbukti efektif pada penelitian sejenis [15].

Di luar aspek integrasi, ada beberapa fitur teknis yang membuat sistem ini berbeda dari cara pengelolaan kos biasanya. Pertama, sistem ini menerapkan kontrol akses berdasarkan peran (RBAC) pada backend menggunakan middleware yang membatasi akses endpoint sesuai peran, dan pada frontend dengan penyesuaian antarmuka yang menyembunyikan menu dan aksi yang tidak relevan, sesuai dengan prinsip keamanan yang diungkapkan dalam penelitian Mary dan Febriyani [10] serta Nasich dkk. [14]. Kedua, sistem ini mendukung pembayaran bertahap dengan memisahkan entitas tagihan dan transaksi individual, sehingga satu tagihan dapat dilunasi dalam beberapa kali pembayaran tanpa kehilangan catatan. Ketiga, pelacakan sejarah perpanjangan kontrak dilakukan melalui hubungan saling merujuk pada entitas kontrak, yang memungkinkan pengelola untuk mengikuti urutan masa sewa seorang penyewa tanpa kehilangan data kontrak sebelumnya.

Walaupun demikian, ada beberapa batasan yang perlu diakui. Saat ini, sistem belum terhubung dengan layanan notifikasi eksternal seperti WhatsApp, sehingga pengiriman pengingat tagihan masih bergantung pada pemantauan manual melalui antarmuka sistem. Di samping itu, pengujian dilakukan dalam lingkungan localhost, sehingga perilaku sistem dalam kondisi jaringan publik atau saat beban pengguna yang lebih tinggi belum dapat dipastikan.

Berdasarkan evaluasi fungsional terhadap seluruh modul utama, setiap fitur berfungsi sesuai dengan kebutuhan masing-masing peran pengguna. Admin memiliki kemampuan untuk mengelola seluruh operasional kos, sementara Staff dapat mengakses informasi yang relevan tanpa hak untuk mengubah data. Temuan ini menunjukkan bahwa sistem yang dikembangkan dapat memenuhi tujuan penelitian, yaitu menyediakan solusi backoffice manajemen kos yang lebih terintegrasi dibandingkan dengan pendekatan-pendekatan sebelumnya.

4. Kesimpulan

Penelitian ini berhasil merancang dan mengembangkan sistem backoffice manajemen kos berbasis web yang mengintegrasikan pengelolaan properti, penyewa, kontrak, pembayaran, pengeluaran, dokumen, serta pelaporan keuangan dan okupansi dalam satu platform terpusat. Sistem dikembangkan menggunakan metode Prototype dengan arsitektur frontend-backend terpisah, memanfaatkan Next.js pada sisi antarmuka pengguna dan Laravel REST API pada sisi logika bisnis, sehingga mendukung pengembangan yang lebih modular dan mudah dipelihara.

Penerapan autentikasi berbasis token menggunakan Laravel Sanctum serta kontrol akses berbasis peran (RBAC) pada dua lapisan, yaitu sisi backend dan frontend, berhasil membedakan kewenangan antara Admin dan Staff secara konsisten. Selain itu, mekanisme pembayaran bertahap dan pelacakan riwayat perpanjangan kontrak merupakan aspek yang belum ditemukan pada penelitian-penelitian sebelumnya di bidang yang sama. Hasil pengujian fungsional menunjukkan bahwa seluruh fitur utama berjalan sesuai dengan kebutuhan pengguna.

Pengembangan lanjutan dapat dilakukan dengan mengintegrasikan notifikasi otomatis melalui WhatsApp untuk mengingatkan pengelola dan penyewa terkait tagihan yang mendekati atau melewati tenggat waktu, sehingga proses pengingat tidak lagi bergantung pada pemantauan manual. Selain itu, pengujian pada lingkungan deployment nyata perlu dilakukan untuk memvalidasi performa sistem di luar kondisi localhost.

5. Referensi

- [1] Ie Chen and Tony, "DESAIN PERANCANGAN APLIKASI MANAJEMEN KOS BERBASIS WEB PADA KOS 2A," *Jurnal Ilmu Komputer dan Sistem Informasi*, vol. 14, no. 1, Jan. 2026, doi: 10.24912/qgkycc67.
- [2] M. Husni Syamaidzar *et al.*, "ANALISIS DAN PERANCANGAN SISTEM INFORMASI MANAJEMEN KOS BERBASIS WEB MENGGUNAKAN METODE WATERFALL," *JMA*, vol. 3, pp. 3031–5220, 2025, doi: 10.62281.
- [3] R. Laksana Bhakti, I. Nuryasin, and B. Setio Wiyono, "PENGEMBANGAN SISTEM MANAJEMEN KOS BERBASIS APLIKASI MOBILE PADA KOS SIDORAME12," *JATI (Jurnal Mahasiswa Teknik Informatika)*, vol. 9, no. 3, pp. 5259–5266, May 2025, doi: 10.36040/jati.v9i3.13708.
- [4] I. Hasridayyana, L. Ahmad, and R. Junaidi, "Sistem Informasi Manajemen Kos-Kosan Menggunakan Metode Content Management System Di Kota Banda Aceh," *Jurnal Sistem Komputer (SISKOM)*, vol. 3, no. 2, pp. 84–94, Aug. 2023, doi: 10.35870/siskom.v3i2.797.
- [5] A. K. Yulianto and N. I. Prasetya, "PENGEMBANGAN SISTEM SEWA KAMAR KOS PADA "KOS YANI"," *Melek IT: Information Technology Journal*, vol. 10, no. 2, pp. 101–110, Dec. 2024, doi: 10.30742/melekitjournal.v10i2.340.
- [6] D. S. Purnia, R. Ratningsih, M. Surahman, and W. Agustin, "Implementasi Metode Prototyping Pada Rancang Marketplace Rumah Kost Berbasis Mobile," *EVOLUSI: Jurnal Sains dan Manajemen*, vol. 9, no. 1, Apr. 2021, doi: 10.31294/evolusi.v9i1.10145.
- [7] F. P. E. Putra, R. W. Efendi, A. B. Tamam, and W. A. Pramadi, "Trends and Best Practices in API-Based Web Development Using Laravel and React," *Brilliance:*

- Research of Artificial Intelligence*, vol. 5, no. 1, pp. 223–233, Jun. 2025, doi: 10.47709/brilliance.v5i1.5971.
- [8] Andreas Natanael Irawan and A. Permana Sanusi, “Development of a Web-Based ERP System Using Next.js and Laravel: A Prototyping SDLC Approach,” *SMATIKA JURNAL*, vol. 16, no. 01, pp. 88–97, Mar. 2026, doi: 10.32664/smatika.v16i01.2174.
- [9] Aldino Dewa Ndaru, Mujahidin Aljawari, and Muh Yudha Setiawan, “Implementasi REST API Pengelolaan Data Penduduk Multi-Desa Berbasis Laravel dengan Autentikasi Sanctum,” *Merkurius : Jurnal Riset Sistem Informasi dan Teknik Informatika*, vol. 3, no. 4, pp. 325–335, Jul. 2025, doi: 10.61132/merkurius.v3i4.997.
- [10] T. Mary and N. Febriyani, “Peningkatan Keamanan Sistem Informasi Berbasis Laravel 12 dengan Rate Limiting dan Role-Based Access Control (RBAC),” *Jurnal Teknologi Dan Sistem Informasi Bisnis*, vol. 7, no. 3, pp. 473–481, Jul. 2025, doi: 10.47233/jteksis.v7i3.1976.
- [11] Aldy Nifratama, Tri Suratno, and Daniel Arsa, “Analisis dan Evaluasi Pengujian pada Penerapan Metode Prototype dalam Software Engineering,” *Jurnal Ilmiah Media Sisfo*, vol. 18, no. 1, pp. 128–138, Apr. 2024, doi: 10.33998/mediasisfo.2024.18.1.1649.
- [12] S. Ramdany, “Penerapan UML Class Diagram dalam Perancangan Sistem Informasi Perpustakaan Berbasis Web,” *Journal of Industrial and Engineering System*, vol. 5, no. 1, Jul. 2024, doi: 10.31599/2e9afp31.
- [13] S. Nugraha and D. W. Chandra, “Peningkatan Keamanan Database Pada Layanan Azure Melalui Metode Multi-Tenant Dengan Pendekatan Separate Database,” *Jurnal Pendidikan dan Teknologi Indonesia*, vol. 3, no. 6, pp. 233–240, Jun. 2023, doi: 10.52436/1.jpti.293.
- [14] A. K. Nasich, S. A. Wicaksono, and M. C. Saputra, “Implementasi Role Based Access Control (RBAC) dalam Sistem Informasi Manajemen Pelanggan dan Pembayaran Air Berbasis Web (studi pada PT Tirta Wangi Sejahtera),” *Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer*, vol. 9, no. 9, pp. 2548–964, 2025, [Online]. Available: <http://j-ptiik.ub.ac.id>
- [15] S. N. Sarfiah, A. P. Nilasari, R. Retnosari, and R. Abidin, “Perancangan Aplikasi Laporan Keuangan Berbasis Web Untuk Pelaku UMKM,” *Jati: Jurnal Akuntansi Terapan Indonesia*, vol. 6, no. 1, pp. 81–95, May 2023, doi: 10.18196/jati.v6i1.18034.