

Implementasi Otomatisasi Tagihan Pembayaran Kos dengan Whatsapp Gateway

Az Zahraina Nur Habibah^{a,1*}, Rina Fiati^{b,2}, Esti Wijayanti^{b,3}

^aTeknik Informatika, Fakultas Teknik, Universitas Muria Kudus, Kudus, 59327, Indonesia
1202251022@std.umk.ac.id, 2rina.fiati@umk.ac.id, 3esti.wijayanti@umk.ac.id

Submission:27/01/2024, Revision: 30/05/2024, Accepted : 30/11/2024

Abstract

This research aims to develop and implement the “Smart Watch Tagihan Kos” application, which is specifically designed to simplify and enhance the process of managing and monitoring boarding house bills. The application provides an efficient and reliable platform for organizing essential data, including room information, resident details, and billing records, all presented through a user-friendly and intuitive interface. The software development methodology applied in this project is the waterfall method, which is a structured approach that follows distinct phases: requirement analysis, system design, implementation, system testing, and ongoing maintenance. In the system testing phase, the black box testing technique is employed, allowing for the evaluation of various functional aspects and pages within the application to ensure proper functionality. The findings of this research demonstrate that the “Smart Watch Tagihan Kos” system successfully offers an effective and practical solution that significantly meets the management needs of boarding house administrators, streamlining operations and improving the accuracy of billing processes.

Keywords: Management Application, Boarding House Bills, Smart watch, Waterfall Method, Black Box Testing

Abstrak

Penelitian ini bertujuan untuk mengembangkan dan mengimplementasikan aplikasi “Smart Watch Tagihan Kos” yang secara khusus dirancang untuk menyederhanakan dan meningkatkan proses pengelolaan dan pemantauan tagihan kos. Aplikasi ini menyediakan platform yang efisien dan andal untuk mengelola data penting, termasuk informasi kamar, detail penghuni, dan catatan tagihan, yang semuanya disajikan melalui antarmuka yang mudah digunakan dan intuitif. Metodologi pengembangan perangkat lunak yang diterapkan dalam proyek ini adalah metode waterfall, yaitu pendekatan terstruktur yang mengikuti fase-fase yang berbeda: analisis kebutuhan, desain sistem, implementasi, pengujian sistem, dan pemeliharaan berkelanjutan. Pada fase pengujian sistem, teknik pengujian black box digunakan, yang memungkinkan evaluasi berbagai aspek fungsional dan halaman dalam aplikasi untuk memastikan fungsionalitas yang tepat. Temuan penelitian ini menunjukkan bahwa sistem “Smart Watch Tagihan Kos” berhasil menawarkan solusi yang efektif dan praktis yang secara signifikan memenuhi kebutuhan manajemen administrator kos, merampingkan operasi dan meningkatkan akurasi proses penagihan.

Kata kunci: Aplikasi Manajemen, Tagihan Kost, Jam Tangan Pintar, Metode Waterfall, Pengujian Black Box

This is an open access article under the [CC BY-SA](#) license.



1. Pendahuluan

Dalam era ini, peran komputer telah berkembang menjadi lebih dari sekadar alat bantu untuk meningkatkan efisiensi dalam pekerjaan. Komputer juga menjadi fondasi utama dalam mengelola bisnis kos-kosan, bentuk tempat penginapan yang semakin populer di tengah masyarakat, terutama di perkotaan. Pemilik kos umumnya menetapkan harga sewa dengan berbagai fasilitas [1]. Mengelola usaha kos-kosan dengan jumlah anak kos yang beragam memerlukan sistem informasi yang efisien untuk mencatat data penghuni, tanggal jatuh tempo pembayaran, dan informasi lainnya [1], [2], [3], [4].

Pentingnya sistem manajemen dalam usaha kos-kosan semakin terlihat, terutama dengan pertumbuhan jumlah anak kos. Contoh usaha kos-kosan yang mengalami kesulitan dalam pencatatan data adalah Kos Zahra, yang berlokasi di Desa Bumirejo, Kecamatan Bae, Kabupaten Kudus. Pengamatan

menunjukkan bahwa Kos Zahra masih menggunakan pencatatan manual dengan mencatat data di buku. Metode ini memiliki beberapa kelemahan, seperti ketidakakuratan data, keterbatasan dalam pencarian informasi, dan risiko kehilangan data jika buku fisik rusak atau hilang.

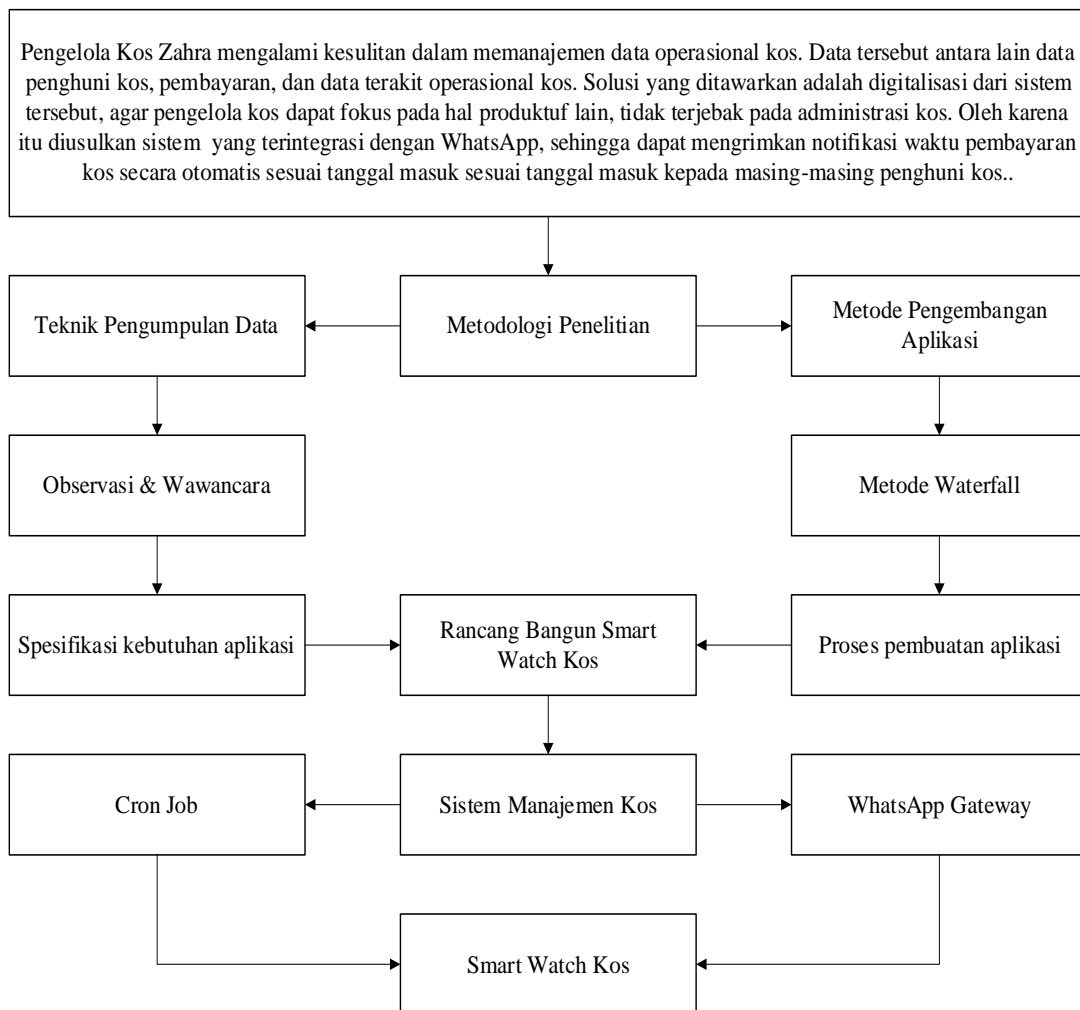
Dampaknya, muncul sejumlah konsekuensi negatif, termasuk kesulitan dalam penagihan pembayaran kos [3], ketidakakuratan catatan keuangan [5], dan ketidakefisienan waktu dalam mencari informasi [6]. Oleh karena itu, tujuan utama penelitian ini adalah mengembangkan sistem manajemen tagihan kos berbasis website yang diharapkan dapat mengatasi permasalahan tersebut. Dengan adanya sistem ini, pemilik kos diharapkan dapat mengelola data anak kos dengan lebih mudah dan efisien, menyediakan informasi yang lebih akurat, serta memfasilitasi penagihan uang kos melalui fitur WhatsApp tanpa harus mencari nomor handphone anak kos satu per satu [7].

2. Metode Penelitian

Penelitian ini melibatkan sejumlah prosedur yang terstruktur untuk menghasilkan sistem manajemen tagihan kos berbasis website. Berikut adalah penjelasan lengkap tentang prosedur penelitian ini.

2.1. Perumusan Masalah

Proses dimulai dengan mengidentifikasi permasalahan yang dihadapi oleh usaha kos-kosan, terutama terkait dengan manajemen data tagihan, pencatatan pembayaran, dan efisiensi dalam penagihan. Perumusan masalah yang jelas membantu mengarahkan penelitian ke arah yang tepat untuk mencari solusi.



Gambar. 1 Kerangka Pikir

2.2. Studi Literatur

Tahap ini melibatkan peninjauan literatur terkait manajemen kos-kosan, sistem informasi manajemen, dan pengembangan aplikasi berbasis web. Analisis literatur membantu memahami konsep-konsep yang telah ada, teknologi yang relevan, dan solusi yang telah diimplementasikan pada masalah serupa.

Penelitian oleh [1] bertujuan untuk membangun e-kos, sebuah sistem informasi yang dapat membantu penyewa dalam melakukan booking kamar kos, pembayaran bulanan kos, dan pencatatan pengeluaran biaya pemeliharaan kos. Metode pendekatan sistem yang digunakan adalah object oriented dengan penelitian sistem menggunakan waterfall. Sistem informasi e-kos dapat membantu pengelola kos mendokumentasikan setiap proses pembayaran dengan baik dan mendata pengeluaran, termasuk menghasilkan laporan pengeluaran.

Penelitian oleh [7] bertujuan untuk mengetahui sistem informasi administrasi kos yang berjalan pada Kos London Residence and Lodging, serta merancang sistem informasi baru yang dapat mempermudah pengelolaan administrasi kos. Penelitian ini menggunakan metode deskriptif dan metode sekuensial linier. Hasil penelitian menunjukkan bahwa sistem informasi administrasi baru dapat mempermudah proses pendaftaran, perpanjangan sewa, pengecekan kerusakan fasilitas, dan laporan keuangan.

Penelitian oleh [8] bertujuan untuk membangun sistem informasi pengelolaan rumah kos Deo Garut berbasis web yang dapat mempermudah proses pendaftaran penghuni baru, transaksi bulanan, dan pelaporan keluhan penghuni. Penelitian ini menggunakan metode Unified Software Development Process (USDP). Hasil penelitian menunjukkan bahwa sistem informasi yang dibangun dapat memenuhi tujuan penelitian.

Penelitian oleh [9] bertujuan untuk membangun sistem informasi penyewaan kos secara online yang dapat mempermudah pemilik kos dan pencari kos. Penelitian ini menggunakan metode SDLC waterfall. Hasil penelitian menunjukkan bahwa OKOS dapat memenuhi tujuan penelitian dan dapat mempermudah pemilik kos dan pencari kos dalam melakukan transaksi penyewaan kos.

Penelitian oleh [3] bertujuan untuk membangun sistem informasi pembayaran biaya kos berbasis web yang dapat mempermudah proses pembayaran biaya kos, mengarsipkan data pembayaran, dan memberikan pelayanan yang akurat. Penelitian ini menggunakan metode waterfall. Hasil penelitian menunjukkan bahwa sistem informasi yang dibangun dapat memenuhi tujuan penelitian.

Penelitian ini membedakan diri dari penelitian-penelitian sebelumnya dalam bidang pengembangan sistem manajemen tagihan kos berbasis website. Beberapa penelitian sebelumnya telah menggali aspek-aspek tertentu terkait manajemen kos-kosan, seperti pencatatan data penghuni atau penagihan pembayaran. Namun, penelitian ini menyajikan kontribusi yang lebih fokus pada pengembangan sistem berbasis website yang mencakup sejumlah fitur penting, termasuk halaman login, dashboard admin, kelola data kamar, kelola data penghuni, kelola data tagihan, kelola pengguna sistem, dashboard pengguna, halaman data tagihan pengguna, dan otomatisasi pengiriman tagihan melalui WhatsApp Gateway. Dengan demikian, penelitian ini menitikberatkan pada keseluruhan proses manajemen kos-kosan dengan pendekatan yang lebih komprehensif, memberikan solusi yang lebih terintegrasi dan berorientasi pada kebutuhan pemilik kos serta kenyamanan penghuni.

2.3. Perancangan Sistem

Pada langkah ini, dilakukan perancangan sistem termasuk identifikasi kebutuhan fungsional dan non-fungsional. Desain mencakup perancangan antarmuka pengguna, struktur database, dan alur kerja aplikasi. Proses ini menghasilkan blueprint untuk implementasi sistem.

2.4. Pengembangan Aplikasi

Berdasarkan perancangan sistem, dilakukan pengkodean menggunakan bahasa pemrograman PHP dan HTML untuk logika backend dan frontend. Penggunaan XAMPP sebagai server lokal dan Visual Studio Code sebagai editor memfasilitasi pengembangan aplikasi secara efisien [10].

2.5. Implementasi Cron Job

Untuk mengotomatisasi pengiriman notifikasi tagihan melalui WhatsApp Gateway, sistem cron job dikonfigurasi. Cron job menjadwalkan pengiriman notifikasi pada interval waktu tertentu, meningkatkan efisiensi dalam proses pengingat pembayaran [11].

2.6. Pengujian Aplikasi

Pengujian dilakukan menggunakan metode black box testing, melibatkan sejumlah uji fungsional pada berbagai halaman, termasuk halaman login, dashboard, kelola data kamar, pengguna, dan tagihan. Pengujian dilakukan oleh pengguna aplikasi (penghuni kos) untuk memastikan keamanan, kegunaan, dan kinerja aplikasi.

2.7. Analisis Hasil Pengujian

Hasil pengujian dievaluasi dengan mempertimbangkan skor dari setiap pertanyaan pada kuesioner, yang melibatkan aspek kecepatan respons, kemudahan penggunaan, keamanan privasi, dan lainnya. Analisis data pengujian membantu mengevaluasi kinerja aplikasi dan mendapatkan umpan balik pengguna [12].

2.8. Pemeliharaan dan Refaktor

Setelah mendapatkan umpan balik dari pengujian, dilakukan pemeliharaan sistem dengan membuat dokumentasi pada kode dan menulis manual book. Kode sistem juga direfaktor untuk memastikan keberlangsungan dan kemudahan pengembangan lebih lanjut [13].

2.9. Penyusunan Panduan Pengguna

Untuk mempermudah penggunaan aplikasi, disusun panduan pengguna yang berisi petunjuk langkah demi langkah untuk pengguna tipe admin dan penghuni kos. Panduan ini membantu pemahaman dan penerapan fungsi-fungsi aplikasi dengan lebih baik [14].

3. Hasil dan Pembahasan

3.1. Analisis Kebutuhan

Analisis kebutuhan pada penelitian ini akan merinci dua aspek penting, yaitu kebutuhan fungsional dan kebutuhan non-fungsional pada pengembangan sistem manajemen tagihan kos berbasis website. Kebutuhan fungsional akan mencakup spesifikasi detail terkait fitur-fitur yang diinginkan dan diperlukan oleh pemilik kos serta penghuni kos. Sedangkan, kebutuhan non-fungsional akan membahas aspek-aspek performa, keamanan, dan antarmuka pengguna yang perlu dipertimbangkan dalam merancang sistem ini. Spesifikasi komputer yang diperlukan untuk pengembangan sistem ini adalah:

- a. *Processor AMD Ryzen 7 5700U*
- b. *RAM 16 GB LPDDR4x*
- c. *SSD 512 GB SSD M.2 2280 PCIe*
- d. *Sistem operasi Windows 11 Home + OHS Pre-installed*
- e. *XAMPP 7.4.33*

Kebutuhan fungsional dalam penelitian ini merujuk pada fitur dan fungsi spesifik yang diharapkan terdapat dalam aplikasi manajemen tagihan kos berbasis *website*. Adapun kebutuhan fungsional aplikasi ini mencakup sejumlah aspek penting:

- a. Aplikasi diharapkan memiliki kemampuan untuk mencatat data anak kos secara sistematis, termasuk informasi identitas, data pembayaran, dan tanggal jatuh tempo.
- b. Sistem harus dapat memantau tanggal jatuh tempo pembayaran dan memberikan notifikasi kepada anak kos secara otomatis melalui fitur *WhatsApp*.
- c. Aplikasi juga diharapkan dapat memfasilitasi proses penagihan uang kos dengan efisien, termasuk memberikan informasi tagihan yang jelas kepada pemilik kos.
- d. Kemampuan untuk mengelola perubahan, seperti pergantian anak kos dengan membarui data dan menyesuaikan tagihan, juga menjadi kebutuhan fungsional penting.

Kebutuhan non-fungsional pada penelitian ini merujuk pada aspek-aspek yang tidak terkait langsung dengan fitur atau fungsi aplikasi, tetapi lebih kepada karakteristik pengoperasian dan kinerja sistem. Berikut adalah beberapa kebutuhan non fungsional aplikasi ini:

- a. Performa yang cepat dan responsif adalah kebutuhan non-fungsional utama, memastikan aplikasi berjalan efisien tanpa keterlambatan signifikan.

- b. Keamanan menjadi aspek non-fungsional yang krusial, menjamin perlindungan data penghuni kos-kosan dari potensi risiko keamanan.
- c. Antarmuka pengguna yang intuitif dan mudah digunakan adalah kebutuhan non-fungsional lainnya, mendukung penggunaan oleh pemilik kos dan anak kos tanpa memerlukan keterampilan teknis khusus.
- d. Keterjangkauan menjadi pertimbangan penting, dengan harapan biaya pengembangan dan pemeliharaan aplikasi tidak melampaui manfaat yang diberikan.

3.2. Perancangan Database

Setiap tabel memiliki peran dan fungsi khususnya dalam menyimpan dan mengorganisir data yang relevan [15]. Oleh karena itu, pemahaman yang baik tentang struktur dan hubungan antar tabel akan membuka jalan bagi pengembangan sistem yang efektif dan efisien.

Aplikasi hasil penelitian ini memanfaatkan beberapa tabel yang saling terkait untuk menyimpan data dengan struktur yang terorganisir. Hubungan kardinalitas ini dirancang dengan cermat untuk menciptakan struktur basis data yang sesuai dengan kebutuhan fungsional aplikasi, memastikan integritas data, dan mendukung operasional yang efisien. Ada beberapa tabel yang saling terhubung. Tabel-tabel tersebut diantaranya:

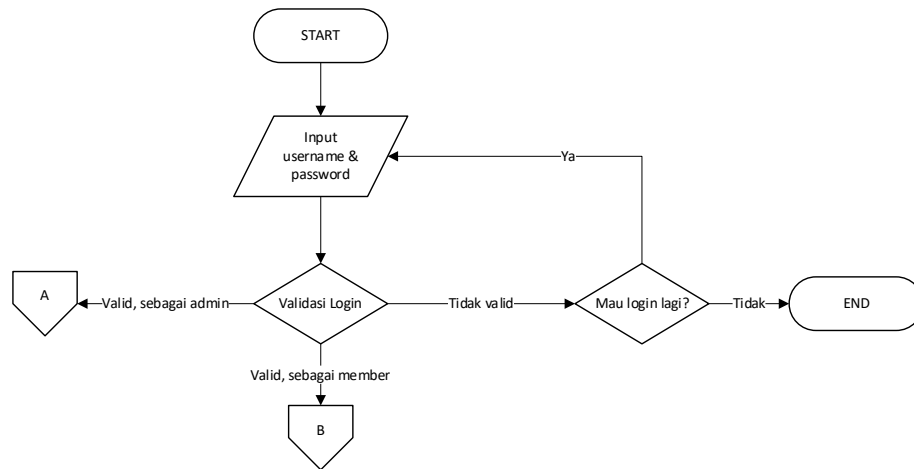
- a. Tabel *tb_penghuni* dan *tb_kamar*. Terdapat relasi banyak ke satu (M:1) antara *tb_penghuni* dan *tb_kamar*. Artinya, satu kamar (*tb_kamar*) dapat ditempati oleh banyak penghuni kos (*tb_penghuni*), tetapi setiap penghuni hanya dapat terdaftar pada satu kamar.
- b. Tabel *tb_penghuni* dan *tb_tagihan*. Relasi antara *tb_penghuni* dan *tb_tagihan* bersifat satu ke banyak (1:M). Setiap penghuni kos (*tb_penghuni*) dapat memiliki beberapa tagihan (*tb_tagihan*), tetapi setiap tagihan hanya terkait dengan satu penghuni.
- c. Tabel *tb_tagihan* dan *tb_bulan*. Terdapat relasi banyak ke satu (M:1) antara *tb_tagihan* dan *tb_bulan*. Hal ini mengindikasikan bahwa banyak tagihan (*tb_tagihan*) dapat terkait dengan satu bulan tertentu (*tb_bulan*), namun setiap bulan hanya dapat terdapat pada satu atau lebih tagihan.
- d. Tabel *tb_pengguna* dan *tb_tagihan*. Terdapat relasi banyak ke satu (M:1) antara *tb_pengguna* dan *tb_tagihan*. Artinya, satu pengguna sistem (*tb_pengguna*) dapat terkait dengan beberapa tagihan (*tb_tagihan*), tetapi setiap tagihan hanya terkait dengan satu pengguna.
- e. Tabel *tb_kamar* dan *tb_tagihan*. Relasi antara *tb_kamar* dan *tb_tagihan* bersifat satu ke banyak (1:M). Satu kamar (*tb_kamar*) dapat terkait dengan beberapa tagihan (*tb_tagihan*), tetapi setiap tagihan hanya terkait dengan satu kamar.

3.3. Perancangan Sistem

Pada sub-bab ini akan dipaparkan serangkaian langkah rinci dalam merancang sistem manajemen tagihan kos berbasis *website*. Pertama, akan disajikan *Flowchart* Proses *Login* untuk memberikan gambaran visual langkah-langkah dalam proses autentikasi pengguna. Selanjutnya, akan dibahas *Flowchart* Proses Admin dan *Flowchart* Proses *Member* untuk memperinci alur kerja masing-masing peran dalam sistem. *Use case diagram* akan memvisualisasikan interaksi antara aktor dan sistem, sedangkan *Class Diagram* akan merinci struktur kelas dan hubungan antar mereka. *Activity diagram* akan memperlihatkan serangkaian aktivitas dalam suatu proses, sedangkan *Sequence diagram* akan menggambarkan urutan pesan antar objek pada sistem.

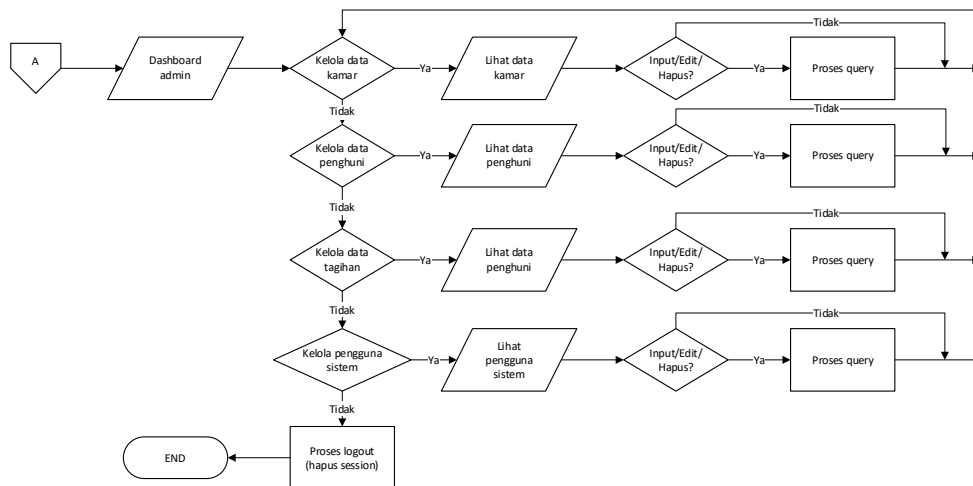
Flowchart proses *Login* dimulai dari *start*, di mana pengguna diminta untuk memasukkan *username* dan *password*. Setelah itu, sistem melakukan percabangan untuk memvalidasi *Login*. Jika *Login* valid sebagai admin, alur akan berlanjut ke *off-page connector* A yang mengarah ke halaman admin. Jika *Login* valid sebagai *member*, alur akan beralih ke *off-page connector* B yang mengarah ke halaman *member*. Namun, jika *Login* tidak valid, sistem akan memeriksa apakah pengguna ingin mencoba *Login* lagi atau tidak. Jika pengguna tidak ingin mencoba lagi, alur proses akan berakhir (*end*). Namun, jika pengguna ingin mencoba *Login* lagi, alur akan kembali ke langkah *input username* dan *password*, memberikan kesempatan untuk mencoba *Login* ulang.

Alur ini memberikan pandangan yang jelas tentang bagaimana sistem mengelola proses *Login*, dengan berbagai jalur berdasarkan hasil validasi dan keputusan pengguna.



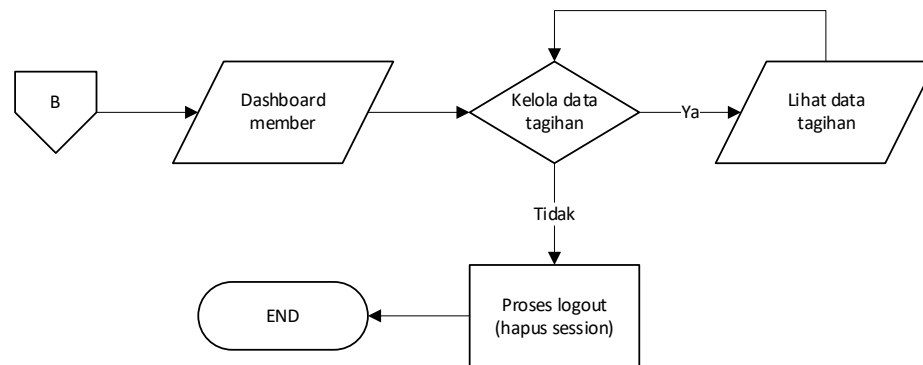
Gambar. 2 Flowchart proses login

Dalam *Flowchart* Proses Admin yang diawali dari *off-page connector* A, sistem pertama-tama akan menampilkan *Dashboard* admin setelah *Login*. Di *Dashboard* ini, admin memiliki opsi untuk mengelola beberapa aspek utama, yaitu data kamar, data penghuni, data tagihan, dan pengguna sistem. Admin dapat memilih salah satu dari opsi tersebut, dan jika tidak ada pilihan yang dipilih, maka pilihan terakhir adalah *logout*, yang akan mengakhiri sesi dan menuju langkah akhir (*end*). Jika admin memilih untuk mengelola data kamar, data penghuni, atau data tagihan, sistem akan mengarahkannya ke tampilan data masing-masing. Selanjutnya, jika admin melakukan *input*, edit, atau hapus data, *query* akan diproses sesuai dengan tindakan yang diambil. Namun, jika tidak ada tindakan tersebut, alur akan kembali ke *Dashboard* admin untuk memberikan fleksibilitas dan efisiensi dalam pengelolaan sistem.



Gambar. 3 Flowchart proses admin

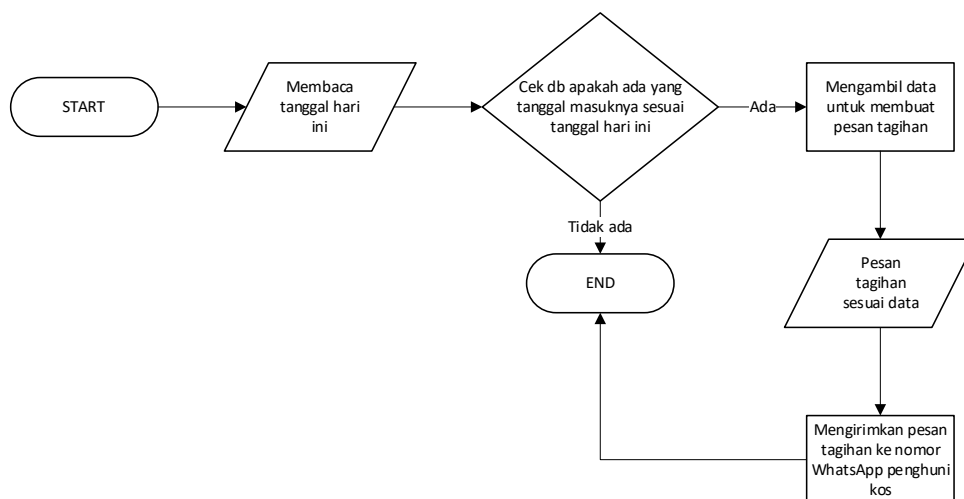
Flowchart proses untuk pengguna tipe *member* dimulai setelah berhasil *login* ke aplikasi. Setelah *login*, pengguna akan diarahkan ke halaman *dashboard* yang menyajikan informasi dalam bentuk *card*, termasuk jumlah tagihan yang sudah lunas dan yang belum dibayar. Dari halaman ini, pengguna memiliki dua opsi utama, yakni mengelola data tagihan atau melakukan *logout*. Apabila pengguna memilih untuk mengelola data tagihan, sistem akan membimbingnya melalui serangkaian langkah yang memungkinkan pengguna untuk melihat dan mengelola informasi tagihan mereka dengan mudah dan efisien.



Gambar. 4 Flowchart proses admin

Flowchart proses pengiriman notifikasi tagihan kos harian ini dimulai dengan “*Start*” pada *cron job* yang dijadwalkan untuk berjalan setiap hari. Langkah pertama, program melakukan “*Input*” dengan membaca tanggal saat ini. Selanjutnya, program melakukan “*seleksi kondisi*” dengan memeriksa database untuk melihat apakah ada penghuni kos yang tanggal masuknya bertepatan dengan tanggal hari ini.

Jika kondisi terpenuhi, berarti ada penghuni yang perlu dikirimkan tagihan; program kemudian mengambil “*data penghuni*” dan menggunakan data tersebut untuk “*membuat pesan tagihan*” yang telah disesuaikan dengan biaya kos harian mereka. Pesan tersebut menjadi “*Output*” yang kemudian dikirimkan ke “*nomor WhatsApp penghuni kos*” melalui “*WhatsApp Gateway*”, memastikan bahwa notifikasi tagihan sampai kepada mereka. Setelah pengiriman berhasil, program mencapai tahap “*End*”. Jika tidak ada penghuni yang memenuhi kriteria tanggal masuk pada hari tersebut, program tidak menghasilkan *output* apa pun dan langsung berakhir tanpa melakukan pengiriman notifikasi.



Gambar. 5 Flowchart proses pengingat otomatis

3.4. Implementasi

Proses implementasi dalam penelitian ini menitikberatkan pada penerapan kode program menggunakan XAMPP versi 7.4.33 dan Visual Studio Code. XAMPP, sebagai paket pengembangan web terintegrasi, menjadi pilihan utama karena menyediakan lingkungan yang mencakup Apache sebagai server web, MySQL sebagai sistem manajemen basis data, PHP sebagai bahasa pemrograman *server-side*, dan *Perl*. Keberadaan komponen-komponen ini secara bersama-sama menciptakan lingkungan pengembangan yang komprehensif dan memudahkan integrasi antara server web dan basis data.

Visual Studio Code dipilih sebagai editor kode sumber karena kemudahan penggunaan, kecepatan, dan dukungan untuk berbagai bahasa pemrograman. Editor ini mempermudah penulisan, pengelolaan, dan *debugging* kode aplikasi secara efisien. Pada tahap ini, tim pengembang menindaklanjuti desain sistem yang telah disiapkan sebelumnya. Proses penulisan kode melibatkan implementasi halaman-halaman kunci seperti

halaman *login*, *dashboard* admin, kelola data kamar, kelola data penghuni, kelola data tagihan, kelola pengguna sistem, *dashboard* pengguna, dan halaman data tagihan pengguna.

Setelah menyelesaikan tahap pengkodean, langkah berikutnya dalam penelitian ini adalah mengimplementasikan sistem *cron job* untuk otomatisasi pengiriman notifikasi tagihan kos melalui *WhatsApp Gateway*. *Cron job*, sebuah utilitas perangkat lunak pada sistem operasi *Unix*, berfungsi sebagai penjadwalan tugas yang dijalankan secara otomatis pada waktu yang telah ditentukan. Penggunaan *cron job* dalam penelitian ini difokuskan pada pengiriman notifikasi tagihan kos kepada penghuni dengan interval waktu tertentu sesuai jatuh tempo pembayaran.

Konfigurasi *cron job* dilakukan untuk mengatur waktu dan frekuensi pengiriman notifikasi. Sistem diatur untuk secara terjadwal mengirimkan pesan notifikasi tagihan kepada penghuni melalui *WhatsApp Gateway* sesuai dengan waktu yang telah ditentukan dalam *cron job*. Dengan implementasi *cron job* ini, proses notifikasi pembayaran dapat dilakukan secara berkala dan otomatis, mengeliminasi keterlambatan dan meningkatkan efisiensi dalam memberikan informasi kepada penghuni kos.

Halaman *Dashboard* Admin dalam aplikasi hasil penelitian menyajikan gambaran menyeluruh tentang status dan performa kos-kosan. Pada halaman ini, admin dapat dengan cepat memantau berbagai informasi kunci yang relevan dengan manajemen kos, yang disajikan dalam bentuk *card*. Halaman *Dashboard* Admin bertujuan untuk memberikan pemahaman yang jelas dan langsung terkait dengan perkembangan dan keberlangsungan kos-kosan, memungkinkan admin untuk mengambil keputusan berbasis data dengan lebih akurat dan cepat.



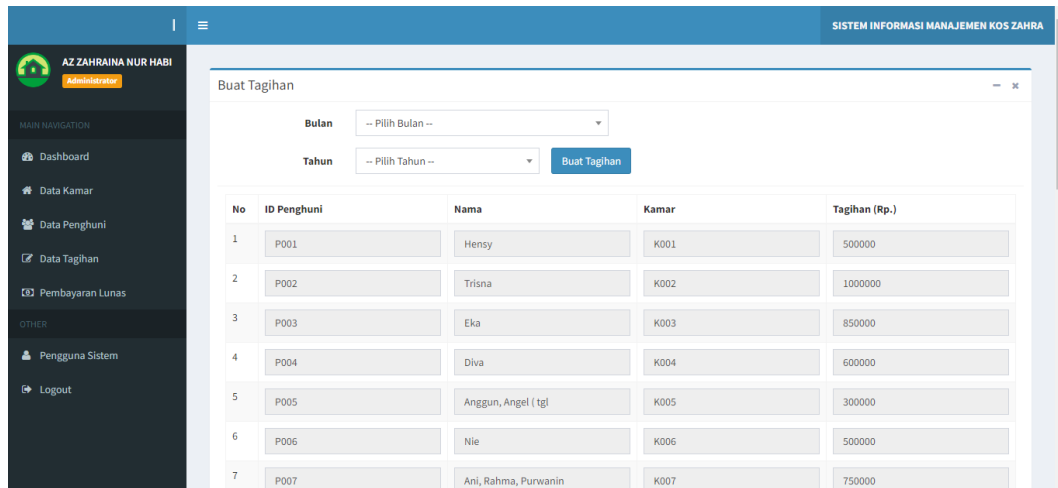
Gambar. 6 Halaman dashboard admin

Halaman Kelola Data Kamar pada aplikasi ini memberikan akses kepada admin untuk mengelola informasi terkait kamar-kamar yang ada dalam kos. Pada halaman ini, admin dapat melihat, menambahkan, mengedit, atau menghapus data kamar secara efisien. Data kamar mencakup beberapa aspek penting seperti *ID* kamar, lantai tempat kamar tersebut berada, kapasitas penghuni, fasilitas yang disediakan, dan tarif sewa kamar. Halaman manajemen kamar admin dalam proyek ini menampilkan daftar semua kamar yang ada dalam sistem. Bagian atas halaman memiliki tombol untuk menambahkan data kamar baru, yang mengarah ke halaman tambah kamar. Ada juga beberapa tombol untuk mengatur tampilan kotak informasi, seperti tombol untuk collapse dan menghapus kotak.

Halaman Kelola Data Penghuni pada aplikasi ini dirancang untuk memberikan kemudahan bagi admin dalam mengelola informasi terkait penghuni kos. Pada halaman ini, admin dapat mengakses dan mengelola biodata penghuni, yang meliputi informasi seperti nama, alamat, dan nomor telepon. Selain itu, terdapat kolom yang menyimpan data kode kamar yang dihuni oleh setiap penghuni. Keberadaan kolom ini memungkinkan admin untuk dengan cepat mengetahui lokasi kamar penghuni tersebut, memberikan informasi yang lebih terperinci terkait penempatan setiap penghuni dalam kos.

Halaman Kelola Data Tagihan pada aplikasi ini memberikan fungsionalitas kepada admin untuk dengan mudah mengakses dan mengelola informasi terkait tagihan kos. Pada halaman ini, admin memiliki kemampuan

untuk melihat seluruh daftar tagihan, baik yang sudah lunas maupun yang belum. Selain itu, admin juga dapat membuat tagihan baru sesuai dengan rentang waktu tertentu. Informasi tagihan melibatkan detail terkait setiap penghuni, termasuk jumlah yang harus dibayarkan, status pembayaran, serta tanggal jatuh tempo.

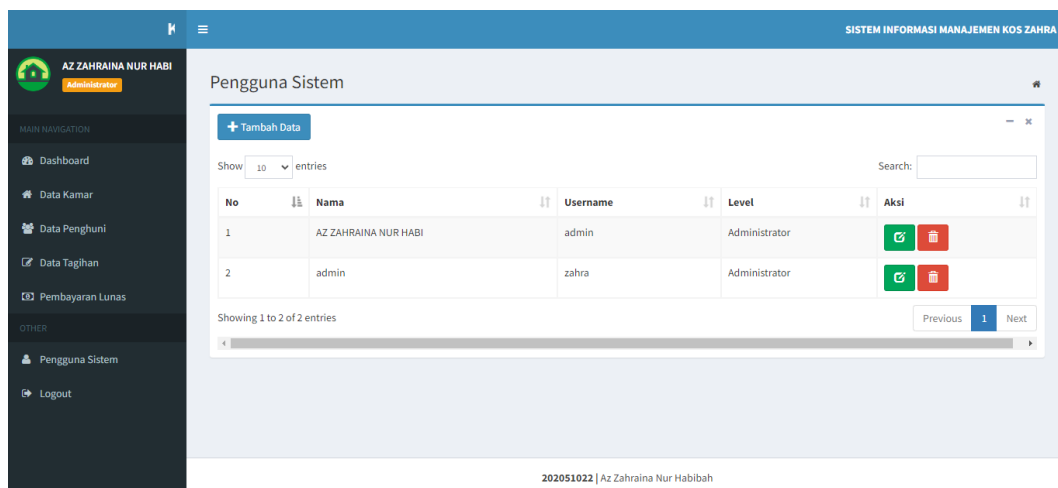






No	ID Penghuni	Nama	Kamar	Tagihan (Rp.)
1	P001	Hensy	K001	500000
2	P002	Trisna	K002	1000000
3	P003	Eka	K003	850000
4	P004	Divya	K004	600000
5	P005	Anggun, Angel (tgl)	K005	300000
6	P006	Nie	K006	500000
7	P007	Ani, Rahma, Purwanin	K007	750000

Gambar. 7 Halaman kelola data tagihan

Halaman Kelola Pengguna Sistem memberikan admin kemampuan untuk mengelola data pengguna dengan peran tipe admin pada aplikasi ini. Admin dapat dengan mudah menambah, mengubah, atau menghapus data pengguna sistem yang memiliki hak akses tertentu. Informasi yang dapat dikelola meliputi *username*, *password*, dan peran pengguna dalam sistem. Pada halaman ini, admin dapat menjalankan fungsi administratif terkait manajemen pengguna, memastikan bahwa setiap pengguna sistem memiliki hak akses dan kredensial yang sesuai. Dengan adanya Halaman Kelola Pengguna Sistem, admin dapat dengan efektif mengatur dan memelihara struktur pengguna sistem, meningkatkan keamanan dan kontrol dalam operasional aplikasi.

Bagian atas halaman memiliki tombol untuk menambahkan data pengguna baru, yang mengarah ke halaman `add_pengguna`. Ada juga beberapa tombol untuk mengatur tampilan kotak informasi, seperti tombol untuk mengcollapse dan menghapus kotak. Bagian utama halaman adalah tabel yang menampilkan detail setiap pengguna. Kolom tabel mencakup nomor urut, nama, username, level, dan aksi yang dapat dilakukan terhadap pengguna tersebut.

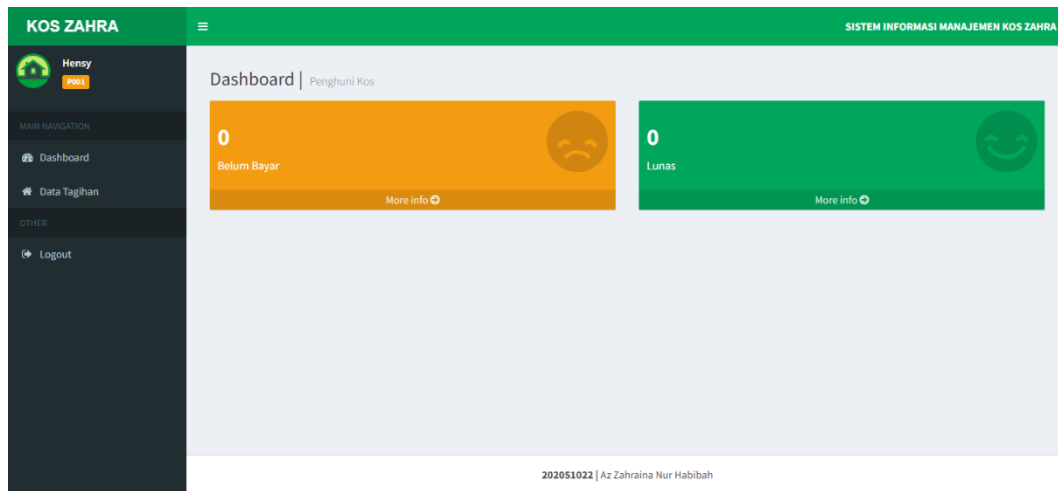


No	Nama	Username	Level	Aksi
1	AZ ZAHRAINA NUR HABI	admin	Administrator	 
2	admin	zahra	Administrator	 

Gambar. 8 Halaman kelola pengguna sistem

Halaman *Dashboard* Pengguna dirancang untuk memberikan pengguna akses cepat dan informatif terkait tagihan yang harus mereka bayar. Pada halaman ini, pengguna dapat melihat jumlah tagihan mereka dalam bentuk *card* yang mencakup informasi penting seperti total tagihan yang harus dibayarkan.

Jika proses login berhasil pengguna akan dialihkan ke halaman `aplication.php`. Ini ditunjukkan oleh kode `window.location = 'aplication.php'`; Jika proses login gagal, pengguna akan tetap berada di halaman `member.php`. Ini ditunjukkan oleh kode `window.location = 'member.php'`; Kode ini juga menggunakan library SweetAlert (`Swal.fire`) untuk menampilkan pesan popup kepada pengguna tentang hasil proses login - apakah berhasil atau gagal. Pesan popup akan berisi judul 'Login Berhasil' atau 'Login Gagal' dan ikon yang sesuai.

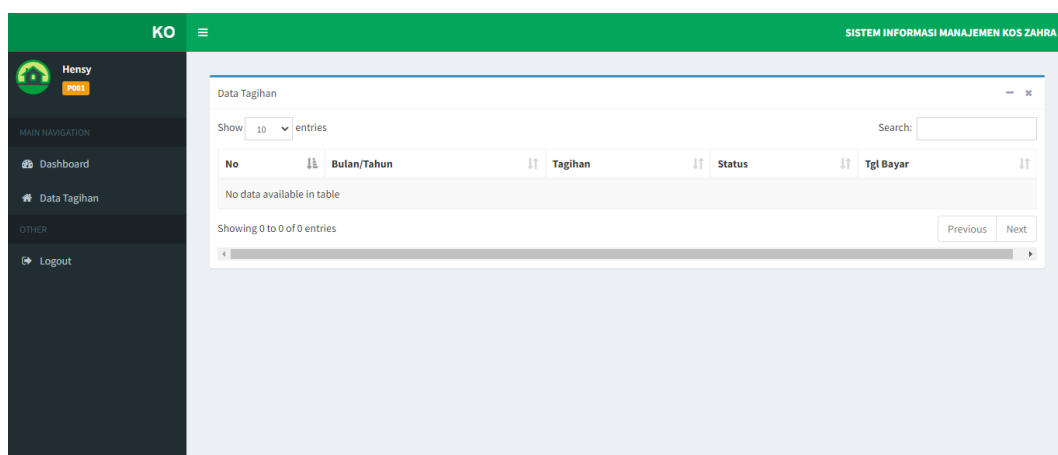


Gambar. 9 Halaman dashboard pengguna

Halaman Tagihan dirancang khusus untuk memberikan pengguna tipe penghuni akses rinci terkait tagihan mereka. Pada halaman ini, penghuni dapat melihat detail tagihan secara lengkap, termasuk rincian biaya dan tanggal jatuh tempo. Informasi tersebut disajikan secara terstruktur, memungkinkan pengguna untuk dengan mudah memahami komponen-komponen tagihan dan memastikan bahwa tidak ada informasi yang terlewat.

Bagian atas halaman ini menampilkan sebuah kotak informasi yang berisikan data tagihan, yang dapat dengan mudah diatur melalui beberapa tombol fungsionalitas. Terdapat tombol-tombol khusus yang memungkinkan pengguna untuk mengcollapse atau menyembunyikan kotak informasi tersebut, serta tombol untuk menghapus kotak jika diperlukan. Bagian ini dirancang untuk memberikan fleksibilitas dan kemudahan penggunaan, memastikan bahwa informasi yang ditampilkan dapat disesuaikan dengan preferensi pengguna.

Bagian utama halaman menampilkan sebuah tabel yang merinci setiap tagihan secara lebih rinci. Tabel ini memiliki beberapa kolom, termasuk nomor urut, bulan/tahun tagihan, jumlah tagihan, status pembayaran, dan tanggal pembayaran. Struktur tabel ini dirancang untuk memberikan tampilan yang terorganisir dan informatif bagi pengguna, memudahkan mereka untuk memahami dan memantau setiap tagihan dengan jelas.



Gambar. 10 Halaman tagihan

3.5. Pengujian Aplikasi

Pengujian merupakan tahap kritis dalam pengembangan aplikasi “Smart Watch Tagihan Kos,” yang melibatkan serangkaian tes untuk memastikan bahwa setiap fitur dan halaman dalam sistem berfungsi sesuai dengan spesifikasi yang telah ditentukan. Pada subbab ini, akan dibahas hasil pengujian untuk setiap komponen utama aplikasi, mulai dari Halaman *Login* hingga Pengujian Otomatisasi Pengiriman Tagihan. Tes ini mencakup verifikasi fungsionalitas, pengujian antarmuka pengguna, serta uji otomatisasi untuk memastikan bahwa seluruh sistem berjalan dengan stabil dan sesuai dengan kebutuhan.

4. Kesimpulan

Kesimpulan yang dapat diambil adalah bahwa sistem ini mampu memberikan solusi efektif untuk mengelola data kamar, data penghuni, dan data tagihan, serta menyajikannya melalui antarmuka yang mudah digunakan. Melalui proses implementasi dan pengujian, aplikasi ini telah diuji secara menyeluruh dan memberikan hasil yang memuaskan.

5. Daftar Pustaka

- [1] M. D. Rahmatya, D. E. S. Simangunsong, dan M. F. Wicaksono, “e-Kos sebagai Sistem Informasi Pengelolaan Kos pada Mazasi’s House,” *J. Teknol. dan Inf.*, vol. 12, no. 2, hal. 176–190, 2022, doi: 10.34010/jati.v12i2.8027.
- [2] M. M. Mur *et al.*, “Metode Extreme Programming Dalam Membangun Aplikasi Kos-Kosan Di Kota Bandar Lampung Berbasis Web,” *J. Cendekia*, vol. XVIII, no. 2013, hal. 377–383, 2019.
- [3] Q. Hubbairah, “Pembuatan Sistem Informasi Pembayaran Biaya Kos Berbasis Web,” *J. Online Mhs. Bid. Tek. dan Sains Fak. Tek. Univ. Riau*, vol. 7, hal. 1–7, 2020, [Daring]. Tersedia pada: <https://jom.unri.ac.id/index.php/JOMFTEKNIK/article/view/28342>
- [4] Y. D. Arimbi, D. Kartinah, dan A. N. W. Della, “Rancangan Sistem Informasi Kost Putri Malika Berbasis Website Menggunakan Framework Laravel Dan Mysql,” *J. Ilm. Multidisiplin*, vol. 1, no. 03, hal. 93–103, 2022, doi: 10.56127/jukim.v1i03.201.
- [5] N. P. E. T. Jayanti, I. G. P. K. Juliharta, dan B. P. W. Nirmala, “Implementasi Sistem Informasi Penjualan dan Penerimaan Kas berbasis Web pada UD. Indo Bintang Walet,” *J. Teknol. Inf. dan Komput.*, vol. 7, no. 1, hal. 76–88, 2021.
- [6] Fitri Zuyina Nur Azizah, Adine Pramesti Gitaswari, Dwiana Maryamas Sofa, Rexana Widyadhana Augusta, dan Zulva Nur Fadila, “Sistem Informasi Pengelolaan Data Kos Annisa Menggunakan Framework Code Igniter 4,” *J. Inform. dan Multimed.*, vol. 14, no. 2, hal. 7–14, 2023, doi: 10.33795/jim.v14i2.467.
- [7] M. I. Aziz dan R. Fauzan, “Sistem Informasi Administrasi pada Kos London Residence and Lodging,” *Elibrary.Unikom.Ac.Id*, 2022, [Daring]. Tersedia pada: https://elibrary.unikom.ac.id/1639/13/UNIKOM_MUHAMMAD_IRFANUL_AZIZ_Artikel.pdf
- [8] R. Setiawan, A. D. Supriatna, dan A. H. Kusuma, “Rancang Bangun Sistem Informasi Pengelolaan Rumah Kos Deo Garut Berbasis Web,” *J. Algoritma*, vol. 17, no. 2, hal. 368–377, 2021, doi: 10.33364/algoritma/v.17-2.368.
- [9] D. K. P. Wahyudi dan S. R. Wicaksono, “Sistem Informasi Penyewaan Kos Dengan Fitur Google Maps API,” *J. Innov. Inf. Technol. Appl.*, vol. 3, no. 2, hal. 122–131, 2021, doi: 10.35970/jnita.v3i2.774.
- [10] M. O. Fitri, “Awebserver Sebagai Alternatif Pengganti Xampp Pada Platform Android,” *Teknosains Media Inf. Sains Dan Teknol.*, vol. 15, no. 2, hal. 245, 2021, doi: 10.24252/teknosains.v15i2.20028.
- [11] F. W. Christanto dan R. Rudiyanto, “Cron Job Technique pada Integrasi WLAN Controller Device dan Google Maps API Berbasis Website dalam Jaringan Indonesia Wifi,” *Matrix J. Manaj. Teknol. dan Inform.*, vol. 10, no. 2, hal. 50–57, 2020, doi: 10.31940/matrix.v10i2.1477.
- [12] M. Sholeh, R. Y. Rachmawati, dan E. N. Cahyo, “Penerapan Regresi Linear Ganda Untuk Memprediksi Hasil Nilai Kuesioner Mahasiswa Dengan Menggunakan Python,” *J. Din. Inform.*, vol. 11, no. 1, hal. 13–24, 2022, [Daring]. Tersedia pada: <https://jdi.upy.ac.id/index.php/jdi/article/view/124>
- [13] R. Y. Endra, Y. Aprilinda, Y. Y. Dharmawan, dan W. Ramadhan, “Analisis Perbandingan Bahasa

-
- Pemrograman PHP Laravel dengan PHP Native pada Pengembangan Website,” *Expert J. Manaj. Sist. Inf. dan Teknol.*, vol. 11, no. 1, hal. 48, 2021, doi: 10.36448/expert.v11i1.2012.
- [14] N. Rahmawati dan H. Mulyono, “Panduan Mudah Belajar Framework Laravel,” *PT Alex Media Komputindo*, vol. 1, no. 2, hal. 104–116, 2018, [Daring]. Tersedia pada: <http://jurnalmsi.stikom-db.ac.id/index.php/jurnalmsi/article/view/44/38>
- [15] K. Christiono dan H. Sama, “Studi Komparasi Database Management System Antara Maria Db Dan Postgresql Terhadap Efisiensi Penggunaan Sumber Daya Komputer,” *Conf. Business, Soc. Sci. Innov. Technol.*, vol. 1, hal. 573–579, 2020, [Daring]. Tersedia pada: <http://journal.uib.ac.id/index.php/cbssit>