

Aplikasi Manajemen Kontrakan Berbasis Web dengan Metode Extreme Programming

Hoqi Agustian¹, Asep Budiman², Winda Apriandari³

^{1,2,3} Universitas Muhammadiyah, Sukabumi

Email : hoqia883@gmail.com

Abstrak

Pengelolaan kontrakan secara manual menimbulkan ketidakefisienan, memakan waktu, dan menyebabkan kesalahan administrasi. Penelitian ini mengembangkan sebuah aplikasi manajemen kontrakan berbasis web dengan menerapkan metode Extreme Programming (XP) yang mendukung pengembangan perangkat lunak secara iteratif, memungkinkan perubahan sesuai kebutuhan pengguna. Aplikasi ini dirancang untuk membantu pemilik properti mencatat penyewaan, melacak pembayaran, serta mengelola fasilitas kontrakan dengan lebih terstruktur dan efisien melalui platform online yang dapat diakses kapan dan di mana saja. Proses pengembangan aplikasi ini meliputi perencanaan kebutuhan pengguna, perancangan antarmuka yang ramah pengguna, pengkodean menggunakan teknik pair programming, serta pengujian berkelanjutan dengan metode Test-Driven Development (TDD). Selain itu, implementasi Laravel Sanctum pada aplikasi ini memungkinkan autentikasi berbasis token, sehingga keamanan data pengguna lebih terjamin dan terhindar dari akses yang tidak sah. Aplikasi ini dibangun menggunakan framework Laravel yang mendukung struktur modular dan pemisahan logika bisnis, sehingga mempermudah pemeliharaan dan penambahan fitur di masa mendatang. Hasil pengujian menunjukkan bahwa aplikasi ini mampu meningkatkan efisiensi dalam pengelolaan kontrakan serta memastikan perlindungan data pengguna. Dengan penerapan metode XP, aplikasi dapat terus dikembangkan secara berkelanjutan dan responsif terhadap kebutuhan yang dinamis, menjadikannya solusi yang fleksibel dan relevan bagi pemilik properti dalam mengelola kontrakan secara digital. Aplikasi ini diharapkan mampu memberi dampak positif dalam pengelolaan properti serta menjadi referensi dalam pengembangan perangkat lunak di bidang properti.

Kata Kunci : Manajemen kontrakan, aplikasi web, Extreme Programming, Laravel.

PENDAHULUAN

Pengelolaan kontrakan dalam sektor properti adalah kegiatan yang melibatkan berbagai aspek administrasi yang kompleks, memerlukan ketelitian dan ketepatan tinggi. Aktivitas seperti pencatatan informasi penyewaan, pemantauan pembayaran, pengaturan fasilitas, dan komunikasi antara pemilik serta penyewa merupakan elemen penting dalam menjaga kualitas manajemen properti dan meningkatkan kepuasan penyewa. Namun, jika dilakukan secara manual, proses ini cenderung memakan waktu yang lama, kurang efisien, dan rentan terhadap kesalahan administratif. Kesalahan semacam ini berpotensi menurunkan tingkat kepuasan penyewa dan bahkan menyebabkan kerugian finansial

bagi pemilik properti. Kemajuan teknologi di era digital membuka peluang untuk meningkatkan efektivitas dalam pengelolaan kontrakan melalui aplikasi berbasis web. Dengan aplikasi ini, pemilik, pengelola, dan penyewa dapat mengakses sistem kapan saja dan di mana saja, menciptakan transparansi dan efisiensi yang sulit dicapai dengan metode manual (Siswidiyanto et al., 2020).

Dalam pengembangan aplikasi perangkat lunak, dikenal berbagai metodologi dalam Software Development Life Cycle (SDLC) yang berfungsi untuk merencanakan, membuat, menguji, dan memelihara perangkat lunak. Beberapa model SDLC yang umum digunakan antara lain Waterfall, Agile, dan Incremental Model. Waterfall adalah model yang

memiliki struktur linier dengan tahap-tahap yang harus diselesaikan secara berurutan. Model ini cocok untuk proyek yang memiliki kebutuhan yang jelas sejak awal, tetapi kurang fleksibel dalam menghadapi perubahan. Di sisi lain, Agile adalah pendekatan yang lebih adaptif, di mana pengembangan perangkat lunak dilakukan dalam iterasi singkat dan memungkinkan perubahan kebutuhan yang berkelanjutan (Rizky & Sugiarti, 2022). Incremental Model berfokus pada pengembangan perangkat lunak secara bertahap, di mana produk disusun dan disempurnakan secara berkelanjutan. Untuk mencapai tujuan ini, dipilih metodologi *Extreme Programming* (XP), salah satu pendekatan dalam keluarga Agile. XP dianggap sangat cocok untuk proyek yang membutuhkan kemampuan beradaptasi tinggi terhadap perubahan kebutuhan pengguna. XP menekankan pada siklus pengembangan iteratif yang pendek, memungkinkan komunikasi intensif antara tim pengembang dan pengguna. Hal ini memungkinkan pengembang untuk merespons umpan balik pengguna dengan cepat, sehingga fitur-fitur yang dikembangkan selalu relevan dan sesuai dengan kebutuhan nyata di lapangan (Prabandanizwaransa et al., 2023). Metode XP juga mengutamakan interaksi langsung antara pengembang dan pengguna akhir, sehingga aplikasi dapat berkembang sesuai harapan pengguna serta memberikan manfaat praktis dalam pengelolaan property.

Pemilihan metode *Extreme Programming* (XP) dalam penelitian ini didasarkan pada kebutuhan fleksibilitas tinggi dalam pengembangan aplikasi manajemen kontrak. Aplikasi ini dirancang untuk memecahkan berbagai masalah dalam pengelolaan kontrak, seperti pencatatan manual yang sering memakan waktu, kesalahan administratif, dan keterbatasan sistem dalam menghadapi perubahan kebutuhan yang dinamis. Dalam konteks ini, XP menawarkan pendekatan yang tidak hanya fleksibel, tetapi juga memungkinkan kolaborasi intensif antara

pengembang dan pengguna akhir (Prabandanizwaransa et al., 2023). XP memanfaatkan iterasi pendek untuk memastikan setiap fitur yang dikembangkan dapat langsung diuji dan disesuaikan dengan kebutuhan nyata pengguna, menciptakan aplikasi yang selalu relevan dan fungsional. Perbandingan dengan metode lain seperti *Waterfall*, *Scrum*, dan *Rational Unified Process* (RUP) memperjelas keunggulan XP dalam konteks proyek ini. Metode *Waterfall*, misalnya, memiliki struktur linier dengan tahap-tahap yang harus diselesaikan secara berurutan, sehingga kurang fleksibel dalam menghadapi perubahan di tengah jalan. Dalam proyek pengembangan perangkat lunak yang kebutuhannya sering berubah, pendekatan *Waterfall* akan sulit menyesuaikan diri karena setiap perubahan di tahap akhir memerlukan modifikasi besar di tahap-tahap sebelumnya. Sebaliknya, XP memungkinkan tim untuk mengintegrasikan perubahan di setiap iterasi, sehingga aplikasi dapat berkembang seiring dengan kebutuhan yang terus berubah (Prabandanizwaransa et al., 2023). Metode *Scrum*, yang juga termasuk dalam keluarga Agile, memiliki kemiripan dengan XP dalam hal responsif terhadap perubahan. Namun, *Scrum* menggunakan pendekatan sprint dengan durasi yang sudah ditetapkan (biasanya dua hingga empat minggu), yang memerlukan perencanaan setiap sprint secara terstruktur. XP menawarkan fleksibilitas lebih tinggi dibandingkan *Scrum* karena tidak terikat pada durasi sprint yang tetap, sehingga tim pengembang dapat menyesuaikan waktu dan fokus pengembangan sesuai kebutuhan pengguna (Rizky & Sugiarti, 2022). Dengan XP, tim memiliki kebebasan untuk menyesuaikan iterasi dan langsung merespons masukan pengguna tanpa menunggu siklus sprint berikutnya, yang sangat bermanfaat dalam proyek yang membutuhkan penyesuaian cepat.

Selain itu, XP mendukung *Test-Driven Development* (TDD), yaitu proses di

mana pengujian terhadap setiap fitur dilakukan sebelum fitur tersebut diterapkan dalam aplikasi utama. TDD memastikan bahwa setiap komponen berfungsi optimal sebelum diterapkan, membantu menjaga kualitas aplikasi, dan meminimalkan risiko bug yang dapat mengganggu penggunaan. Metode ini juga memberikan keandalan dalam pengembangan iteratif, di mana setiap fitur diuji ketat sehingga meminimalkan risiko kesalahan. *Rational Unified Process* (RUP) juga mendukung pendekatan berbasis pengujian, tetapi cenderung memerlukan dokumentasi yang lebih banyak dan proses pengujian formal yang sering kali memperlambat respons tim terhadap perubahan kebutuhan (Rizky & Sugiarti, 2022). Keunggulan lainnya dari XP adalah adanya praktik *pair programming* dan *refactoring* yang menekankan kualitas kode. Dalam *pair programming*, dua pengembang bekerja bersama pada satu tugas yang sama, meningkatkan kualitas kode melalui diskusi dan evaluasi langsung di setiap langkah pengembangan (Prabandani et al., 2023). Praktik ini berbeda dengan metode lain yang biasanya membiarkan pengembangan dilakukan secara individu, yang mungkin tidak memiliki tingkat evaluasi langsung yang sama. *Pair programming dan refactoring* juga membantu menjaga keteraturan struktur kode, memudahkan pemeliharaan aplikasi di masa depan, dan memastikan bahwa aplikasi dapat terus berkembang tanpa mengalami penurunan kualitas seiring waktu.

Penelitian ini, berjudul "PENGEMBANGAN APLIKASI MANAJEMEN KONTRAKAN BERBASIS WEB DENGAN METODE EXTREME PROGRAMMING" Penelitian ini tidak hanya berfokus pada pengembangan teknis aplikasi, tetapi juga pada analisis dampak praktis dalam pengelolaan properti. Dengan dukungan teknologi yang memadai dan pendekatan XP yang fleksibel, aplikasi ini diharapkan mampu menjadi solusi manajemen

kontrakan yang modern, efektif, dan efisien. Melalui pendekatan inovatif ini, diharapkan dapat tercipta sebuah sistem yang tidak hanya memudahkan pengelolaan kontrakan, tetapi juga meningkatkan efisiensi dan kenyamanan bagi semua pihak yang terlibat.

METODE

Penelitian ini menggunakan metode *Extreme Programming (XP)* dalam pengembangan aplikasi manajemen kontrakan berbasis web. XP adalah salah satu metodologi dalam keluarga Agile yang berfokus pada siklus iteratif pendek, memungkinkan tim pengembang merespons kebutuhan yang terus berubah secara cepat dan efisien. Metode XP dipilih karena sesuai dengan kebutuhan pengembangan aplikasi yang memerlukan kolaborasi intensif antara tim pengembang dan pengguna serta fleksibilitas yang tinggi dalam pengembangan fitur.

XP mencakup beberapa tahapan utama: Planning, Design, Coding, Testing, dan Listening. Tiap tahap dijalankan dalam siklus yang berulang, sehingga hasil umpan balik dapat segera diimplementasikan dalam pengembangan berikutnya. Proses ini memastikan bahwa aplikasi yang dibangun selalu sesuai dengan kebutuhan dan ekspektasi pengguna.

3.1 Tahapan Pengembangan Extreme Programming

Tahapan dalam Extreme Programming melibatkan proses yang intensif dengan fokus pada kualitas dan adaptasi terhadap umpan balik pengguna. Berikut ini adalah penjelasan setiap tahapan yang dilakukan dalam penelitian ini:

3.1.1 Planning

Tahap perencanaan adalah langkah awal dalam XP yang bertujuan untuk memahami kebutuhan pengguna secara mendalam. Pada tahap ini, penulis mengumpulkan dan menganalisis informasi terkait kebutuhan fungsional dan non-fungsional aplikasi manajemen kontrakan. Beberapa kegiatan yang dilakukan pada tahap ini meliputi:

1. Pengumpulan Kebutuhan Pengguna
Melalui wawancara dan diskusi langsung dengan pemilik kontrakan, kebutuhan dasar dari aplikasi didokumentasikan. Pengguna memberikan informasi mengenai fitur-fitur utama yang dibutuhkan, seperti sistem pencatatan penyewaan, pelacakan pembayaran, dan pengaturan fasilitas yang tersedia dalam kontrakan
2. Penyusunan User Stories
Kebutuhan pengguna dituangkan dalam bentuk user stories, yang menggambarkan fungsi-fungsi spesifik yang diinginkan pengguna. Misalnya, sebuah user story untuk pemilik kontrakan adalah: "Sebagai pemilik, saya ingin dapat memantau pembayaran penyewa secara real-time agar tidak ada keterlambatan."
3. Prioritas Fitur
Berdasarkan diskusi dengan pengguna, setiap fitur diurutkan berdasarkan prioritas untuk memastikan pengembangan fitur yang paling mendesak di setiap iterasi. Proses ini memungkinkan pengembang fokus pada fitur yang memberikan nilai terbesar bagi pengguna pada tahap awal.

3.1.2 Design

Setelah perencanaan, tahap desain dilakukan untuk menyusun struktur aplikasi secara sederhana namun fungsional. Karena XP menekankan pada iterasi pendek, desain disusun secara modular agar mudah disesuaikan di kemudian hari. Elemen desain utama yang dirancang adalah sebagai berikut:

1. Desain Antarmuka Pengguna (UI)
UI aplikasi didesain untuk memudahkan pemilik kontrakan dalam mengakses fitur utama. Prototipe antarmuka seperti halaman utama, halaman pencatatan penyewaan, dan pelacakan pembayaran dikembangkan untuk mendapatkan umpan balik dari pengguna.
2. Arsitektur Sistem

Struktur aplikasi disusun agar mendukung modularitas, sehingga komponen-komponen utama dapat diperbarui tanpa mengganggu keseluruhan aplikasi. Misalnya, modul pencatatan penyewaan dibuat terpisah dari modul pelacakan pembayaran, sehingga memungkinkan pengembang melakukan *refactoring* tanpa kesulitan.

3. Diagram Sederhana
Diagram seperti use case dan flowchart digunakan untuk mengilustrasikan alur utama aplikasi. Diagram yang sederhana ini membantu menjaga fleksibilitas desain sehingga tim pengembang dapat melakukan perubahan yang diperlukan berdasarkan umpan balik pengguna tanpa harus merombak desain secara besar-besaran.

3.1.3 Coding

Tahap pengkodean adalah inti dari XP, di mana pengkodean dilakukan dengan praktik terbaik dalam XP seperti *pair programming* dan *refactoring*. Kegiatan utama pada tahap ini meliputi:

1. Pair Programming
Dalam metode ini, dua pengembang bekerja secara bersamaan untuk menulis kode, sehingga kualitas kode dapat terus terjaga. Pair programming membantu mengidentifikasi kesalahan lebih awal dan memungkinkan diskusi untuk menyelesaikan masalah yang muncul selama pengkodean.
2. Refactoring
Setelah fungsi dasar selesai dikembangkan, *proses refactoring* dilakukan untuk menyederhanakan dan mengoptimalkan struktur kode. Dengan *refactoring*, struktur aplikasi tetap terjaga dan mudah dipelihara, bahkan ketika aplikasi bertambah besar dan kompleks.
3. Penulisan Kode yang Mudah Dibaca
Setiap kode ditulis dengan mengikuti pedoman standar agar mudah dipahami oleh anggota tim pengembang lainnya. Praktik ini memastikan bahwa kode dapat diperbaiki dan dikembangkan

lebih lanjut di masa depan tanpa menimbulkan kebingungan.

3.1.4 *Testing*

Dalam penelitian ini, **User Acceptance Testing (UAT)** dipilih sebagai metode evaluasi utama. UAT adalah tahap pengujian perangkat lunak yang melibatkan pengguna akhir secara langsung untuk memastikan bahwa fitur-fitur yang dikembangkan sesuai dengan kebutuhan dan ekspektasi mereka. Dalam konteks aplikasi manajemen kontrakan, UAT dilakukan untuk memvalidasi fungsi utama, seperti pencatatan penyewaan, pelacakan pembayaran, dan pengelolaan fasilitas, agar sesuai dengan kebutuhan praktis pengguna. Tahapan UAT dimulai setelah semua pengujian internal, seperti unit testing dan integrasi, telah selesai. Pengguna menguji aplikasi dalam lingkungan yang mirip dengan kondisi sebenarnya, memberikan umpan balik berdasarkan pengalaman mereka saat menggunakan aplikasi. Hasil dari pengujian ini sangat penting karena memberikan wawasan langsung tentang sejauh mana aplikasi mampu menyelesaikan masalah yang dihadapi oleh pengguna.

Keunggulan UAT terletak pada fokusnya terhadap pengalaman pengguna akhir. Tidak seperti pendekatan lain yang lebih teknis, seperti Test-Driven Development (TDD), UAT menempatkan pengguna sebagai pusat evaluasi, sehingga pengembang dapat memastikan bahwa aplikasi benar-benar memberikan solusi yang sesuai dengan kebutuhan praktis. Masukan dari UAT digunakan untuk melakukan penyesuaian dan perbaikan sebelum aplikasi dirilis secara resmi. Metode ini dipilih karena sifatnya yang langsung menghubungkan hasil pengembangan dengan kebutuhan pengguna, sehingga memastikan aplikasi yang dihasilkan tidak hanya berfungsi dengan baik secara teknis, tetapi juga relevan dan efektif digunakan dalam pengelolaan kontrakan.

3.1.5 **Listening (Feedback)**

Listening atau mendengarkan umpan balik adalah tahap penting dalam XP yang membantu aplikasi berkembang sesuai kebutuhan pengguna. Setelah setiap iterasi, pengguna memberikan feedback mengenai fitur yang telah dikembangkan. Feedback ini digunakan sebagai bahan evaluasi untuk menentukan prioritas fitur di iterasi berikutnya.

1. **Evaluasi Pengalaman Pengguna**
Pengalaman pengguna dievaluasi berdasarkan kemudahan penggunaan dan efektivitas fitur. Berdasarkan hasil evaluasi ini, pengembang dapat memperbaiki antarmuka atau menyesuaikan fungsi fitur.
2. **Revisi Berdasarkan Feedback**
Feedback dari pengguna langsung diterapkan pada iterasi berikutnya. Proses ini memastikan aplikasi tetap relevan dan bermanfaat bagi pengguna karena selalu dikembangkan berdasarkan masukan pengguna.

3.2 **Teknik Pengumpulan Data**

Untuk memahami kebutuhan pengguna dan memastikan aplikasi sesuai dengan ekspektasi, penelitian ini menggunakan beberapa metode pengumpulan data, yang terdiri dari data primer dan sekunder.

1. **Data Primer**
Penulis melakukan pengamatan terhadap pengelolaan kontrakan secara langsung untuk memahami alur kerja dan kebutuhan pengguna.
 - a. **Observasi**
Penulis melakukan pengamatan langsung terhadap proses pengelolaan kontrakan, seperti pencatatan penyewaan, pelacakan pembayaran, dan komunikasi dengan penyewa. Observasi ini bertujuan untuk memahami tantangan yang dihadapi oleh pemilik kontrakan.
 - b. **Wawancara**
Wawancara terstruktur dilakukan dengan pemilik kontrakan untuk mendapatkan wawasan mendalam

mengenai kebutuhan dan harapan mereka terhadap aplikasi manajemen kontrakan. Informasi yang diperoleh digunakan untuk menyusun user stories dan menetapkan prioritas fitur aplikasi.

2. Data Sekunder

Referensi diambil dari jurnal, buku, artikel ilmiah, dan penelitian sebelumnya yang relevan dengan topik pengelolaan properti berbasis web dan metodologi Extreme Programming (XP). Data ini memberikan landasan teori dan mendukung analisis yang dilakukan dalam penelitian.

3.3 Perangkat Penelitian

Untuk mendukung pelaksanaan penelitian ini, beberapa perangkat keras dan perangkat lunak digunakan dalam pengembangan aplikasi manajemen kontrakan berbasis web. Berikut adalah daftar perangkat yang digunakan:

1. *Laravel*

Framework PHP ini digunakan sebagai kerangka utama untuk mengembangkan aplikasi berbasis web, khususnya untuk menangani backend, otentikasi, dan manajemen data. *Laravel* menyediakan berbagai fitur bawaan yang mempermudah pengelolaan data, keamanan, dan pengembangan yang efisien.

2. *MySQL*

MySQL dipilih sebagai sistem manajemen basis data relasional untuk menyimpan berbagai informasi terkait manajemen kontrakan, seperti data penyewaan, transaksi pembayaran, dan data pengguna. *MySQL* memiliki keunggulan dalam kecepatan akses dan kemudahan integrasi dengan *Laravel*.

3. *Blade Template Engine*

Blade adalah template engine bawaan dari *Laravel* yang digunakan untuk mengembangkan antarmuka pengguna (UI) yang dinamis dan terstruktur. *Blade* memungkinkan pemisahan logika aplikasi dari tampilan, sehingga memudahkan proses pengembangan dan pemeliharaan aplikasi.

4. *Laravel Sanctum*

Sanctum digunakan untuk otentikasi berbasis token, sehingga memastikan keamanan data pengguna. Dengan *Sanctum*, sistem dapat mengontrol akses pengguna ke halaman dan data tertentu yang membutuhkan otentikasi, menjaga keamanan informasi sensitif.

5. *Visual Studio Code*

Visual Studio Code (VS Code) adalah editor kode yang digunakan untuk menulis, mengelola, dan menguji kode selama pengembangan aplikasi. *VS Code* dipilih karena mendukung berbagai ekstensi yang meningkatkan produktivitas pengembangan, seperti fitur auto-completion dan debugging.

HASIL DAN PEMBAHASAN

Untuk memudahkan pemahaman dan pembacaan, hasil penelitian dideskripsikan terlebih dahulu, dilanjutkan bagian pembahasan. Subjudul hasil dan subjudul pembahasan disajikan terpisah. Bagian ini harus menjadi bagian yang paling banyak, minimum 60% dari keseluruhan badan artikel.

Hasil

Tahap analisis bertujuan untuk memahami kebutuhan sistem dan mendefinisikan fitur utama yang akan dikembangkan pada aplikasi manajemen kontrakan berbasis web. Analisis ini dilakukan dengan mengacu pada data struktur tabel database dan kode aplikasi *Laravel* yang telah diberikan, serta kebutuhan fungsional pengguna.

1.1.1 Analisis Kebutuhan Sistem

Analisis kebutuhan sistem dilakukan untuk mengidentifikasi kebutuhan utama yang harus dipenuhi oleh aplikasi manajemen kontrakan berbasis web. Hasil analisis ini menjadi dasar dalam proses perancangan dan implementasi aplikasi agar sesuai dengan kebutuhan pengguna dan tujuan pengembangan. Berikut ini adalah rincian kebutuhan sistem yang telah dianalisis.

1. Manajemen Unit Kontrakan

a. Penambahan Unit Kontrakan

Admin dapat menambahkan unit kontrakan baru dengan memasukkan informasi lengkap, termasuk nama unit, deskripsi, harga sewa, status (tersedia atau tersewa), dan unggahan gambar. Hal ini memudahkan admin dalam memperbarui data unit secara langsung tanpa menggunakan metode manual.

- b. Pengeditan Unit Kontrakan
Admin dapat mengedit data unit yang sudah ada, termasuk mengubah nama unit, deskripsi, harga, status, dan mengganti gambar jika diperlukan. Fitur ini berguna untuk memperbarui informasi unit agar tetap relevan, seperti saat terjadi perubahan harga sewa atau renovasi fasilitas.
- c. Penghapusan Unit Kontrakan
Admin dapat menghapus unit kontrakan yang sudah tidak aktif atau tidak tersedia. Data unit yang dihapus tidak akan muncul di halaman pengguna atau dashboard admin. Fitur ini memastikan database tetap terorganisir dengan hanya menyimpan data unit yang aktif.
- d. Pengelolaan Status Unit
Admin dapat mengubah status unit dari "tersedia" menjadi "tersewa" atau sebaliknya. Fitur ini memastikan data status unit selalu terkini, sehingga pengguna dapat melihat ketersediaan unit secara real-time.

2. Manajemen Pengguna

- a. Penambahan Pengguna Baru
Admin dapat menambahkan pengguna baru dengan memasukkan informasi seperti nama, email, dan peran pengguna (admin atau user). Fitur ini memastikan hanya pengguna yang terdaftar yang dapat mengakses sistem.
- b. Pengeditan Data Pengguna
Admin dapat memperbarui informasi pengguna yang sudah ada, seperti nama dan email. Fitur ini memastikan data pengguna selalu akurat dan terkini.
- c. Penghapusan Data Pengguna

Admin dapat menghapus data pengguna yang tidak lagi aktif atau relevan. Fitur ini membantu menjaga keamanan dan memastikan database tetap terorganisir.

1.2 Perencanaan (Planning)

Tahap perencanaan bertujuan untuk memahami kebutuhan pengguna dan merancang fitur utama aplikasi yang sesuai dengan kebutuhan mereka. Berikut adalah rincian proses yang dilakukan dalam tahap ini:

1.2.1 Pengumpulan Kebutuhan

Pengumpulan kebutuhan dilakukan melalui wawancara dan diskusi langsung dengan pemilik kontrakan, yang merupakan calon pengguna aplikasi. Tujuannya adalah untuk mengidentifikasi masalah yang sering mereka hadapi dan menentukan fitur utama yang diperlukan dalam sistem. Beberapa kebutuhan yang berhasil diidentifikasi meliputi:

1. Efisiensi dalam pengelolaan unit kontrakan, seperti pencatatan dan pelacakan status unit.
2. Transparansi dalam informasi penyewaan dan pembayaran.

Hasil Pengumpulan Kebutuhan :

1. Sistem harus menyediakan fitur untuk menambah, mengedit, dan menghapus data unit kontrakan.
2. Informasi ketersediaan unit kontrakan harus tersedia secara real-time.

1.2.2 Penyusunan User Stories

User stories digunakan untuk menggambarkan kebutuhan pengguna dalam bentuk skenario sederhana. Setiap *user story* mendefinisikan satu fungsi spesifik dari aplikasi yang dirancang untuk memenuhi kebutuhan pengguna. Berikut adalah beberapa *user stories* yang disusun berdasarkan kebutuhan:

1. Sebagai admin, saya ingin dapat menambahkan unit kontrakan baru agar data kontrakan dapat dikelola dengan lebih mudah.
2. Sebagai admin, saya ingin mengubah informasi unit kontrakan, termasuk harga dan status, agar data selalu relevan dan terkini.

3. Sebagai admin, saya ingin menghapus unit kontrakan yang tidak aktif agar database tetap terorganisir.
4. Sebagai pengguna, saya ingin melihat daftar unit kontrakan yang tersedia agar saya dapat memilih unit sesuai kebutuhan.
5. Sebagai pengguna, saya ingin melihat informasi detail unit, termasuk fasilitas dan harga, untuk membantu saya membuat keputusan.

1.2.3 Prioritasi Fitur

Setelah user stories disusun, fitur-fitur yang akan dikembangkan diurutkan berdasarkan tingkat kepentingan dan dampaknya terhadap sistem. Fitur dengan prioritas tinggi dikembangkan lebih awal untuk memberikan nilai terbesar bagi pengguna dalam waktu singkat. Berikut prioritas nya:

1. CRUD untuk unit kontrakan (penambahan, pengeditan, penghapusan, dan pengelolaan status).
2. Sistem pencatatan penyewaan.
3. Tampilan daftar unit kontrakan untuk pengguna.

1.3 Desain (Design)

Tahap desain dalam Extreme Programming (XP) berfokus pada penyusunan arsitektur sistem, perancangan antarmuka pengguna, dan pembuatan diagram alur kerja. XP menekankan desain yang sederhana namun fleksibel untuk memudahkan adaptasi terhadap perubahan kebutuhan pengguna. Berikut adalah komponen utama dalam tahap desain:

1.3.1 Desain Antarmuka Pengguna (UI)

Antarmuka pengguna dirancang agar intuitif dan mudah digunakan oleh admin maupun pengguna. Prototipe UI difokuskan pada tampilan yang minimalis namun fungsional, dengan mempertimbangkan kenyamanan navigasi dan aksesibilitas fitur utama.

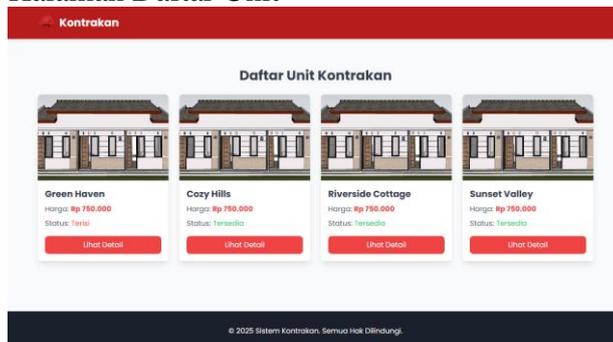
1. Dashboard Admin



Gambar 1. Dashboard Admin

- a. Menampilkan ringkasan data, seperti jumlah unit tersedia, unit tersewa, dan pengguna terdaftar.
- b. Menu navigasi untuk mengelola unit kontrakan, pengguna, dan laporan transaksi.

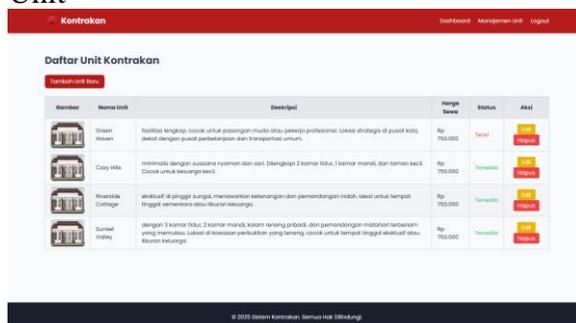
2. Halaman Daftar Unit



Gambar 2. Halaman Daftar Unit

- a. Menampilkan informasi unit seperti nama, deskripsi singkat, harga, status, dan gambar.
- b. Filter dan pencarian unit berdasarkan nama, status, atau rentang harga.

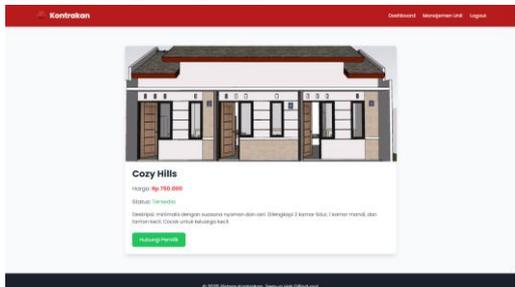
3. Form Penambahan dan Pengeditan Unit



Gambar 3. Form Penambahan dan Pengeditan Unit

- a. Form sederhana untuk memasukkan data unit, seperti nama, deskripsi, harga, status, dan unggahan gambar.
- b. Validasi data untuk memastikan informasi yang dimasukkan sesuai format yang diinginkan.

4. Halaman Detail Unit



Gambar 4 Halaman Detail Unit

- a. Informasi lengkap unit, termasuk deskripsi, fasilitas, harga, dan status ketersediaan.
- b. Tombol tindakan untuk penyewaan atau pengelolaan unit (khusus admin).

1.3.2 Arsitektur Aplikasi

Aplikasi dirancang menggunakan arsitektur Model-View-Controller (MVC), dengan kerangka kerja Laravel. Pendekatan ini memastikan struktur kode yang terorganisir dan memudahkan pengembangan fitur baru tanpa mengganggu fungsi yang sudah ada.

1. Model

- a. Mewakili data inti dari aplikasi, seperti User, Unit, dan Transaction.
- b. Mengelola relasi antar tabel dalam database menggunakan fitur ORM Eloquent Laravel.

2. View

- a. Bertanggung jawab untuk menampilkan data kepada pengguna dalam format HTML yang dinamis.
- b. Blade Template Laravel digunakan untuk menciptakan antarmuka pengguna yang efisien dan konsisten.

3. Controller

- a. Mengatur logika aplikasi, seperti menangani permintaan HTTP dan

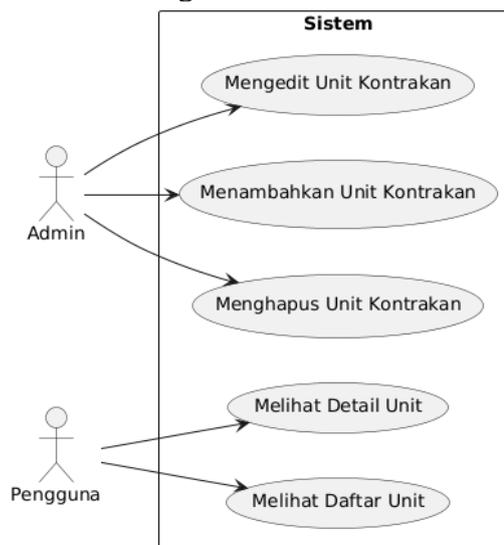
mengarahkan data antara model dan view.

- b. Contoh: UnitController untuk mengelola unit kontrakan, dan UserController untuk manajemen pengguna.

1.3.3 Diagram Alur Kerja

Untuk memvisualisasikan proses utama dalam sistem, dibuat beberapa diagram sederhana:

1. Use Case Diagram



Gambar 5. Use Case Diagram

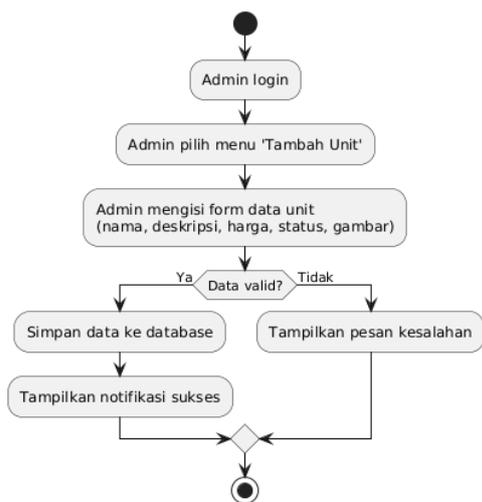
Menggambarkan interaksi antara aktor (admin dan pengguna) dengan fitur utama aplikasi.

- a. Admin: Menambahkan, mengedit, menghapus unit; mengelola pengguna; dan melihat laporan transaksi.
- b. Pengguna: Melihat daftar unit dan detail unit.

2. Flowchart

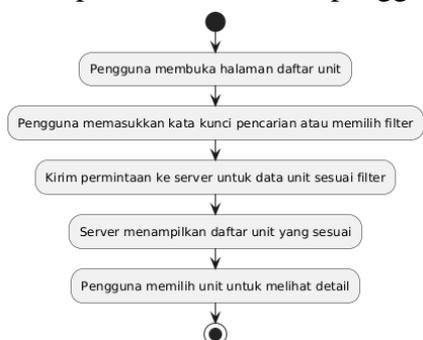
Menunjukkan alur proses dalam sistem, seperti:

- a. Proses penambahan unit oleh admin.



Gambar 6. Proses penambahan unit oleh admin.

b. Proses pencarian unit oleh pengguna.



Gambar 7. Proses pencarian unit oleh pengguna.

3. Entity Relationship Diagram (ERD)
Menjelaskan relasi antar tabel dalam database, seperti:

1.4.1 Implementasi Fitur Utama

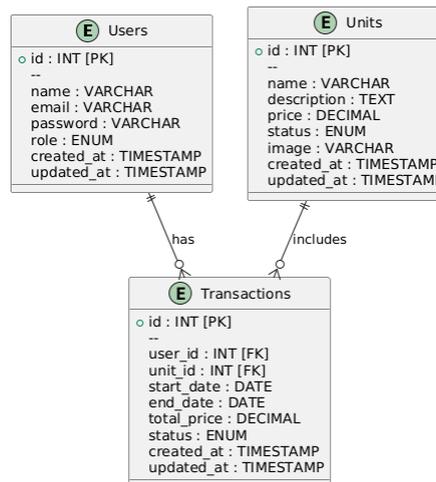
Fitur CRUD (Create, Read, Update, Delete) dikembangkan untuk mengelola data unit kontrakan. Berikut adalah implementasi masing-masing fungsi:

1. Create

Admin dapat menambahkan unit baru melalui form input. Data seperti nama, deskripsi, harga, status, dan gambar unit disimpan ke tabel units. kode di UnitController:

```

public function store(Request $request)
{
    $request->validate([
        'name' => 'required|string|max:255',
        'description' => 'nullable|string',
        'price' => 'required|numeric',
        'status' => 'required|in:available,occupied',
        'image' =>
        'nullable|image|mimes:jpeg,png,jpg|max:2048',
    ]);
}
    
```



Gambar 8. ERD

- a. Tabel users: Menyimpan data pengguna.
- b. Tabel units: Menyimpan data unit kontrakan.

1.4 Pengkodean (Coding)

Tahap pengkodean adalah inti dari proses pengembangan aplikasi dalam Extreme Programming (XP). Pada tahap ini, tim pengembang menerapkan hasil dari tahap perencanaan dan desain ke dalam kode menggunakan prinsip XP seperti Pair Programming, Refactoring, dan Test-Driven Development (TDD). Berikut adalah implementasi pengkodean pada aplikasi manajemen kontrakan berbasis web:

```

    });

    // Handle file upload
    if ($request->hasFile('image')) {
        $imagePath = $request->file('image')->store('images',
'public');
    }

    Unit::create([
        'name' => $request->name,
        'description' => $request->description,
        'price' => $request->price,
        'status' => $request->status,
        'image' => $imagePath ?? null,
    ]);

    return redirect()->route('admin.units.index')-
>with('success', 'Unit berhasil ditambahkan.');
```

2. Read

Data unit ditampilkan kepada admin dan pengguna melalui halaman daftar unit. Fitur ini mencakup pencarian dan penyaringan data berdasarkan nama, status, atau rentang harga.

```

public function index()
{
    $units = Unit::all();
    return view('admin.units.index', compact('units'));
}
```

3. Update

Admin dapat mengedit informasi unit yang sudah ada. Form pengeditan menampilkan data yang sudah tersimpan sebelumnya untuk mempermudah proses pembaruan.

```

public function update(Request $request, Unit $unit)
{
    $request->validate([
        'name' => 'required|string|max:255',
        'description' => 'nullable|string',
        'price' => 'required|numeric',
        'status' => 'required|in:available,occupied',
        'image' =>
'nullable|image|mimes:jpeg,png,jpg|max:2048',
    ]);

    if ($request->hasFile('image')) {
        if ($unit->image) {
            Storage::delete($unit->image);
        }
        $unit->image = $request->file('image')->store('images',
'public');
    }

    $unit->update($request->all());

    return redirect()->route('admin.units.index')-
>with('success', 'Unit berhasil diperbarui.');
```

4. Delete

Admin dapat menghapus unit yang tidak aktif dari database.

Contoh kode:

```
public function destroy(Unit $unit)
{
    if ($unit->image) {
        Storage::delete($unit->image);
    }
    $unit->delete();

    return redirect()->route('admin.units.index')-
>with('success', 'Unit berhasil dihapus.');
```

1.4.2 Pair Programming

Pair Programming diterapkan dalam tahap pengkodean untuk meningkatkan kualitas kode dan mempercepat penyelesaian tugas. Dua pekerja dilakukan secara bersama menulis kode (*driver*), dan memeriksa kode (*observer*). Teknik ini membantu mengurangi kesalahan dan meningkatkan efisiensi.

1.4.3 Refactoring

Refactoring dilakukan untuk menyempurnakan struktur kode tanpa mengubah fungsionalitasnya. Langkah ini penting untuk menjaga agar kode tetap bersih, terorganisir, dan mudah dipelihara.

1.4.4 Standar Penulisan Kode

Kode ditulis mengikuti standar PSR (PHP Standards Recommendations) untuk memastikan keterbacaan dan konsistensi. Setiap fungsi diberi komentar untuk menjelaskan tujuannya, dan nama variabel dipilih agar deskriptif.

1.5 Pengujian (Testing)

Tahap pengujian adalah bagian penting dalam Extreme Programming (XP) untuk memastikan bahwa fitur-fitur aplikasi yang dikembangkan dapat berfungsi dengan baik sesuai kebutuhan pengguna. Dalam penelitian ini, pengujian dilakukan menggunakan metode User Acceptance Testing (UAT), yang melibatkan pengguna akhir dalam proses pengujian untuk menilai fungsionalitas, kegunaan, dan kesesuaian aplikasi dengan kebutuhan praktis.

1.5.1 Tujuan User Acceptance Testing (UAT)

UAT bertujuan untuk memvalidasi bahwa aplikasi manajemen kontrakan telah memenuhi kebutuhan pengguna, baik dari sisi fungsi maupun pengalaman pengguna. Pengujian ini dilakukan dengan melibatkan admin dan pengguna aplikasi dalam lingkungan yang menyerupai kondisi nyata.

1.5.2 Tahapan Pelaksanaan UAT

1. Persiapan Pengujian
 - a. Mengidentifikasi skenario pengujian berdasarkan user stories yang telah disusun pada tahap perencanaan.
 - b. Menyiapkan data uji, seperti daftar unit kontrakan, akun pengguna, dan transaksi penyewaan.
2. Pelaksanaan Uji Coba
 - a. Pengguna (admin) mencoba fitur-fitur utama aplikasi, seperti:
 - Menambahkan unit kontrakan.
 - Mengedit informasi unit.
 - Mencatat penyewaan unit.
 - b. Pengguna (umum) mencoba fitur seperti:
 - Melihat daftar unit yang tersedia.
 - Melihat detail unit kontrakan.
3. Pengumpulan Feedback

Setelah pengujian selesai, pengguna memberikan umpan balik mengenai:

 - a. Kemudahan navigasi dalam aplikasi.

- b. Relevansi fitur dengan kebutuhan mereka.
- c. Masalah atau bug yang ditemukan selama pengujian.

1.5.3 Hasil Pengujian

Hasil pengujian bertujuan untuk memastikan bahwa aplikasi yang dikembangkan berfungsi dengan baik, memenuhi kebutuhan pengguna, dan memberikan pengalaman yang optimal. Pengujian dilakukan berdasarkan skenario yang disusun sebelumnya melalui metode User Acceptance Testing (UAT). Berikut adalah rincian hasil pengujian berdasarkan tiga aspek utama: validasi fungsionalitas, pengalaman pengguna, dan identifikasi masalah.

Tabel 1 Tabel Pengujian

No	Aspek Pengujian	Deskripsi Pengujian	Hasil UAT
1	Validasi Fungsionalitas	Pengujian pada fitur CRUD unit kontrakan. Contoh: Admin menambahkan unit baru bernama "Cozy Hills" dengan data lengkap.	Fitur berjalan sesuai rencana. Unit "Cozy Hills" muncul di daftar unit setelah disimpan.
2	Pengalaman Pengguna	Evaluasi kemudahan navigasi aplikasi, seperti akses form pengelolaan unit dan tampilan daftar unit untuk pengguna umum.	Pengguna menyatakan antarmuka cukup intuitif. Contoh: "Form pengelolaan unit mudah digunakan."
3	Identifikasi Masalah	Pengujian pada validasi input dan konsistensi antarmuka. Masalah ditemukan pada validasi harga sewa yang memungkinkan angka negatif.	Validasi input perlu diperketat. Desain ikon status unit perlu lebih informatif (tersedia: hijau, tersewa: merah).

Rincian Hasil Pengujian:

1. Validasi Fungsionalitas
Semua fitur utama diuji, termasuk:
 - a. Penambahan, pengeditan, penghapusan, dan pengelolaan status unit kontrakan.
 - b. Pencarian dan penyaringan unit berdasarkan nama, status, atau harga.
 - c. Pencatatan transaksi penyewaan unit.
2. Pengalaman Pengguna
Pengujian menunjukkan bahwa aplikasi mudah digunakan oleh admin dan pengguna tanpa memerlukan pelatihan tambahan.
 - a. Antarmuka form dan navigasi menu dinilai intuitif.
 - b. Masukan pengguna: "Ikon status unit perlu diperjelas agar lebih mudah dikenali."
3. Identifikasi Masalah
Beberapa masalah ditemukan selama pengujian, seperti:
 - a. Validasi input harga sewa yang memungkinkan angka negatif.
 - b. Ketidakkonsistenan warna ikon pada status unit (tersedia atau tersewa).
 - c. Waktu pemuatan halaman unit yang lebih dari 3 detik untuk dataset besar.

1.5.4 Tindak Lanjut dari Hasil UAT

Berdasarkan hasil pengujian, beberapa langkah tindak lanjut dilakukan untuk perbaikan:

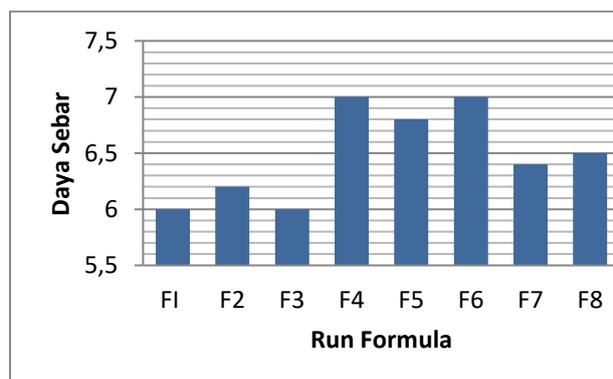
1. Memperbaiki Validasi Input
Menambahkan validasi pada form untuk memastikan harga sewa tidak bernilai negatif dan semua data wajib diisi.
2. Penyempurnaan Antarmuka
Ikon status unit diperbarui dengan warna dan simbol yang lebih informatif dan Konsistensi warna dan desain antarmuka diperbaiki pada semua halaman.
3. Optimasi Performa Sistem

Query database pada halaman daftar unit dioptimalkan untuk meningkatkan kecepatan pemuatan data.

Berdasarkan tabel (1), tabel harus ringkas dan dinomori dengan sistem numerik Arab dalam teks (contoh: Tabel 1, Tabel 2, dan seterusnya). Judul tabel harus jelas dan dapat mengindikasikan isi dari tabel, jika ada informasi tambahan dapat dituliskan dalam catatan kaki. Angka-angka di dalam tabel tidak boleh diulang-ulang dalam narasi verbal baik sebelum maupun sesudahnya.

Penulisan ekuasi harap menggunakan Microsoft Equation seperti contoh dibawah, dan juga dinomori dengan sistem numerik Arab.

$$Y = 0,71912 (A) + 0,43764 (B)..... (1)$$



Gambar 9 Nama Gambar

Semua ilustrasi harus jelas, terbaca dan disitasi secara urut menggunakan sistem numerik Arab pada teks (contoh: Gambar 1, Gambar 2, dan selanjutnya). Mohon untuk menyertakan gambar di dalam teks dan memiliki resolusi tinggi.

Pembahasan

Berdasarkan hasil penelitian ini, beberapa saran diberikan untuk pengembangan dan peningkatan aplikasi di masa mendatang:

1. Pengembangan Fitur Pembayaran Online
Menambahkan fitur pembayaran online akan memberikan kenyamanan lebih bagi penyewa dan pemilik, memungkinkan pembayaran dilakukan langsung melalui aplikasi dan

mempercepat proses pelacakan pembayaran.

2. Penyempurnaan Antarmuka Pengguna (UI)
Meskipun aplikasi saat ini memiliki UI yang intuitif, peningkatan desain UI agar lebih responsif dan menarik dapat meningkatkan pengalaman pengguna, terutama bagi penyewa yang menggunakan aplikasi pada perangkat mobile.
3. Penambahan Fitur Notifikasi Otomatis
Menyediakan notifikasi otomatis melalui email atau SMS untuk pengingat pembayaran dan pengingat akhir masa sewa dapat membantu pemilik dan penyewa untuk tetap terinformasi dan mengurangi risiko keterlambatan pembayaran.

SIMPULAN (PENUTUP)

Berdasarkan penelitian dan pengembangan aplikasi manajemen kontrakan berbasis web menggunakan metode *Extreme Programming (XP)*, berikut kesimpulan yang dapat diambil:

1. Efisiensi Pengelolaan Kontrakan
Aplikasi manajemen kontrakan ini berhasil menyediakan solusi yang efisien untuk pemilik properti dalam mengelola penyewaan, pelacakan pembayaran, dan pengaturan fasilitas. Sistem berbasis web memungkinkan pemilik kontrakan untuk memantau dan mengelola kontrakan dengan lebih terstruktur dan akurat dibandingkan metode manual.
2. Kemampuan Adaptasi dan Pengembangan Berkelanjutan
Metode Extreme Programming (XP) terbukti efektif dalam pengembangan aplikasi ini, karena mendukung adaptasi cepat terhadap umpan balik pengguna. Dengan iterasi yang pendek, aplikasi dapat berkembang sesuai dengan kebutuhan pengguna secara dinamis, menjaga relevansi fitur yang disediakan.
3. Keamanan Data Pengguna

Implementasi *Laravel Sanctum* untuk otentikasi berbasis token memberikan lapisan keamanan tambahan, sehingga data penyewa dan pemilik terlindungi dari akses yang tidak sah. Sistem otentikasi ini menjaga privasi pengguna, yang penting dalam pengelolaan properti berbasis digital.

UCAPAN TERIMA KASIH

Pertama tama saya ucapkan terimakasih kepada yang terlibat pada penelitian ini yang telah membantu saya dalam pengerjaan penelitian ini.

DAFTAR PUSTAKA

- Bima Krisnawan, A., Kurniasih, T., Studi Teknik Informatika, P., Teknologi Informasi, F., Kristen Satya Wacana, U., Salatiga, K., & Jawa Tengah, P. (2024). Optimalisasi Sistem Manajemen Barang dalam Bisnis Sewa Properti melalui Implementasi Appsheets (Studi Kasus: Sewa Rumah BSB). In *Jurnal Indonesia : Manajemen Informatika dan Komunikasi (JIMIK)* (Vol. 5, Issue 2). <https://journal.stmiki.ac.id>
- Hartomi, Z. H., Saputra, H. T., & Arischa, D. (n.d.). INFORMATION SYSTEM DESIGN PAYMENT OF EDUCATION DEVELOPMENT DONATIONS (SPP) WEB-BASED USING LARAVEL. In *Jurnal Testing dan Implementasi Sistem Informasi* (Vol. 1, Issue 2).
- Mukhlis, I. R., & Santoso, R. (2023). Perancangan Basis Data Perpustakaan Universitas Menggunakan MySQL dengan Physical Data Model dan Entity Relationship Diagram. *Journal of Technology and Informatics (JoTI)*, 4(2), 81–87. <https://doi.org/10.37802/joti.v4i2.330>
- Muthia Kansha, W. (n.d.). Analisis Perbandingan Struktur dan Performa Framework Codeigniter dan Laravel dalam Pengembangan Web Application.
- Prabandanizwaransa, I. P., Ahmad, I., & Susanto, E. R. (2023). Implementasi Metode Extreme Programming untuk Sistem Pengajuan Tempat PKL Berbasis Web. *Jurnal Informatika Dan Rekayasa Perangkat Lunak*, 4(2), 221–227. <https://doi.org/10.33365/jatika.v4i2.2601>
- Riadi, I., Umar, R., & Busthomi, I. (n.d.). Optimasi Keamanan Autentikasi dari Man in the Middle Attack (MiTM) Menggunakan Teknologi Blockchain.
- Rizky, M., & Sugiarti, Y. (2022). Penggunaan Metode Scrum Dalam Pengembangan Perangkat Lunak: Literature Review. *Journal of Computer Science and Engineering (JCSE)*, 3(1), 41–48. <https://doi.org/10.36596/jcse.v3i1.353>
- Romzi, M., & Kurniawan, B. (2020). Implementasi Pemrograman Python Menggunakan Visual Studio Code. In *JIK: Vol. XI* (Issue 2). www.python.org
- Siswidiyanto, S., Munif, A., Wijayanti, D., & Haryadi, E. (2020). Sistem Informasi Penyewaan Rumah Kontrakan Berbasis Web Dengan Menggunakan Metode Prototype. *Jurnal Interkom: Jurnal Publikasi Ilmiah Bidang Teknologi Informasi Dan Komunikasi*, 15(1), 18–25. <https://doi.org/10.35969/interkom.v15i1.64>
- Wijanarko, S. (n.d.). Penerapan Test Driven Development (TDD) pada Laravel Menggunakan PHPUnit.
- Yustiana, I. (n.d.). Perancangan Sistem Informasi Kontrakan Bu Jamilah Menggunakan Website. <https://doi.org/10.37034/jidt.v5i1.345>
- Bransford, J. D., Brown, A. L., & Cocking, R. R. (2005). *How people learn: Brain, mind, experience and school*. from <https://www.nap.edu/catalog/9853/how-people-learn-brain-mind-experience-and-school-expanded-edition>.

- Tobias, S., & Duffy, T. M. (Eds.). (2009). *Constructivist instruction: Success or failure?* New York, NY: Routledge.
- Sahlberg, P. (2012). The most wanted: Teachers and teacher education in Finland. In L. Darling-Hammond & A. Lieberman (Eds.), *Teacher education around the world: changing policies and practices*. London: Routledge.
- Schunk, D. H. (2012a). *Learning theories an educational perspective*. Boston, MA: Pearson Education, Inc.
- Schunk, D. H. (2012b). *Learning theories an educational perspective* (E. Hamdiah & R. Fajar, Trans.). Yogyakarta: Pustaka Pelajar. (Original work published 2012).
- Tabachnick, B. G., & Fidell, L. S. (2007). *Using multivariate statistics* (Fifth ed.). Needham Heights, MA: Allyn & Bacon.
- Nurgiyantoro, B. & Efendi, A. (2017). Re-Actualization of Puppet Characters in Modern Indonesian Fictions of The 21st Century. *3L: The Southeast Asian Journal of English Language Studies*. 23 (2), 141-153, from <http://doi.org/10.17576/3L-2017-2302-11>.
- Retnowati, E., Fathoni, Y., & Chen, O. (2018). Mathematics Problem Solving Skill Acquisition: Learning by Problem Posing or by Problem Solving? *Cakrawala Pendidikan*, 37(1), 1-10, from doi: <http://dx.doi.org/10.21831/cp.v37i1.18787>.
- Janssen, J., Kirschner, F., Erkens, G., Kirschner, P. A., & Paas, F. (2010). Making the black box of collaborative learning transparent: Combining process-oriented and cognitive load approaches. *Educational Psychology Review*, 22(2), 139-154. doi: 10.1007/s10648-010-9131-x.
- Retnowati, E. (2012, 24-27 November). *Learning mathematics collaboratively or individually*. Paper presented at the The 2nd International Conference of STEM in Education, Beijing Normal University, China. Retrieved from http://stem2012.bnu.edu.cn/data/short%20paper/stem2012_88.pdf.
- NCTM. (2000). *Principles and standards for school mathematics*. Reston, VA: Author.
- Permendiknas 2009 No. 22, Kompetensi Dasar Pendidikan Pancasila dan Kewarganegaraan Sekolah Dasar Kelas I-VI.