

BAB II TINJAUAN PUSTAKA

2.1 Studi Sebelumnya

Identifikasi penelitian terdahulu yang relevan dengan topik penelitian merupakan salah satu tujuan utama dari tinjauan pustaka. Penelitian-penelitian tersebut berperan sebagai landasan dalam mengembangkan penelitian baru. Berikut adalah beberapa contoh penelitian sebelumnya yang dijadikan sebagai acuan dan informasi latar belakang untuk penelitian ini.

Niayu, dkk. menjelaskan bahwa sistem informasi repositori merupakan solusi untuk mengelola arsip, seperti koleksi skripsi mahasiswa, agar lebih terorganisir, aman, mudah dikelola, dan dapat diakses secara efisien. Penelitian mereka menunjukkan bahwa pengelolaan skripsi secara manual, seperti yang dilakukan di Perpustakaan Universitas Prabumulih, memiliki keterbatasan signifikan, terutama dalam hal aksesibilitas dan efisiensi waktu. Untuk mengatasi permasalahan tersebut, mereka menggunakan metode deskriptif kualitatif melalui observasi langsung, wawancara, dan studi pustaka dalam pengumpulan data. Hasil penelitian mereka berupa rancangan sistem informasi repositori berbasis web yang dirancang dengan metode *Extreme Programming (XP)*, menggunakan alat bantu pemodelan sistem *Unified Modeling Language (UML)*, serta memanfaatkan *PHP* dan *MySQL* sebagai teknologi utama. Solusi ini diharapkan dapat meningkatkan efisiensi pengelolaan dan akses koleksi skripsi bagi staf perpustakaan maupun mahasiswa. Penelitian ini menjadi rujukan penting dalam pengembangan sistem informasi berbasis web untuk mendukung pengelolaan arsip akademik [6].

Nurkholis, dkk. menjelaskan bahwa Kabupaten Sukarame saat ini masih melaksanakan proses administrasi publik secara konvensional, seperti pengelolaan data kependudukan dan layanan surat administratif, menggunakan Microsoft Office untuk penyimpanan data. Pendekatan ini memiliki keterbatasan karena tidak memungkinkan akses *real-time* oleh masyarakat. Untuk mengatasi permasalahan tersebut, mereka mengembangkan sistem informasi yang dirancang untuk mengelola data kependudukan, pengaduan masyarakat, dan layanan surat seperti izin lingkungan, subsidi listrik, serta izin usaha. Sistem ini dikembangkan menggunakan metode *Extreme Programming*

dengan pemodelan sistem yang mencakup *use case diagram*, *class diagram*, dan *activity diagram* berbasis *Unified Modeling Language (UML)*. Uji kelayakan dilakukan menggunakan metode *black-box testing*, yang menunjukkan tingkat akurasi 100%, sehingga sistem dinyatakan layak secara fungsional. Penelitian ini menjadi kontribusi penting dalam pengembangan sistem informasi berbasis teknologi untuk meningkatkan efisiensi dan kualitas layanan publik [7].

Widiastuti mengembangkan sistem informasi kepegawaian untuk Kantor Kelurahan Komering Agung Kecamatan Gunung Sugih guna mengatasi permasalahan sistem manual yang masih digunakan dalam pengelolaan data kepegawaian. Sistem manual ini dinilai memiliki berbagai kelemahan, seperti tingginya penggunaan kertas, rendahnya tingkat keamanan data, tingginya potensi kesalahan pencatatan, serta lamanya waktu yang dibutuhkan dalam pengolahan dan pencarian informasi. Sistem informasi kepegawaian yang dikembangkan dirancang menggunakan metode *Extreme Programming* yang meliputi tahapan *planning* (pengumpulan data, analisis sistem, dan kebutuhan sistem), *design* (pemodelan *use case* dan *activity diagram* serta rancangan antarmuka pengguna), *code* (dengan PHP, Laravel, HTML, CSS, dan MySQL), serta diuji menggunakan standar ISO 25010. Sistem ini dilengkapi dengan fitur *login* dan hak akses pengguna, sehingga mampu meningkatkan keamanan data, mengurangi penggunaan kertas, mempercepat pembuatan laporan, dan menghasilkan informasi yang lebih akurat. Berdasarkan hasil pengujian menggunakan standar ISO 25010, sistem ini dinilai memiliki tingkat keberhasilan "Sangat Baik" dan layak untuk diimplementasikan, sehingga memberikan solusi yang lebih efisien dibandingkan dengan sistem manual [8].

Agung dan Sanusi. mengidentifikasi masalah signifikan dalam pengelolaan aset yang dilakukan oleh Kantor Kecamatan Natar. Saat ini, pengelolaan aset di kantor tersebut masih dilakukan secara manual menggunakan aplikasi Microsoft Excel, yang menyebabkan sejumlah kendala, di antaranya pencatatan aset yang tersebar di beberapa file terpisah. Hal ini membuat proses pencarian data menjadi lambat dan tidak efisien, terutama ketika dibutuhkan secara mendadak. Selain itu, sistem pencatatan yang tidak terintegrasi ini juga menyebabkan dokumentasi aset kurang terorganisir dengan baik, sehingga meningkatkan risiko kesalahan dalam pengelolaan dan pelaporan. Untuk mengatasi permasalahan tersebut, penelitian ini bertujuan mengembangkan sistem

pengelolaan inventaris aset kantor dengan pendekatan *Extreme Programming (XP)*, yang diharapkan dapat menghasilkan perangkat lunak dengan waktu pengembangan yang lebih singkat dan efisien. Pengujian sistem dengan metode *usability* menghasilkan penilaian rata-rata sebesar 89%, yang termasuk dalam kategori baik, menandakan bahwa sistem yang dikembangkan layak untuk diterapkan guna meningkatkan efisiensi pengelolaan aset kantor secara keseluruhan [9].

Putra mengidentifikasi permasalahan signifikan dalam pengelolaan pelayanan perpustakaan di SMK 1 Muhammadiyah Malang. Saat ini, perpustakaan masih mengandalkan sistem manual untuk mencatat transaksi peminjaman dan pengembalian buku menggunakan pembukuan pada buku folio bergaris. Sistem ini menyebabkan beberapa masalah, di antaranya adalah waktu yang terbuang banyak untuk mencatat aktivitas peminjaman dan pengembalian buku secara manual, serta tidak adanya otomatisasi dalam pengolahan denda yang membuat proses menjadi lebih lambat dan rentan kesalahan. Selain itu, petugas perpustakaan juga harus melakukan pencatatan ulang untuk pembuatan laporan transaksi, yang tentunya mempengaruhi efisiensi dan akurasi pengelolaan data. Penelitian ini bertujuan untuk mengatasi permasalahan tersebut dengan mengembangkan sistem informasi perpustakaan yang dapat mempercepat proses peminjaman, pengembalian, pengolahan denda secara otomatis, serta pembuatan laporan transaksi yang lebih efisien. Sistem yang dikembangkan menggunakan metode *Extreme Programming (XP)*, dengan pemodelan sistem menggunakan *Unified Modeling Language (UML)*, dan implementasi sistem menggunakan bahasa pemrograman C-Sharp (C#). Hasil pengujian *white-box testing* dan *black-box testing* menunjukkan bahwa sistem berjalan sesuai dengan kebutuhan, dan hasil *User Acceptance Testing (UAT)* menyatakan bahwa sistem diterima 100% oleh petugas perpustakaan dan layak diterapkan [10].

Tabel 2.1. Literatur Studi Terdahulu

No	Penulis	Tahun	Metode	Tujuan	Hasil
1.	Niayu, dkk.[6]	2023	<i>Extreme Programming</i>	Penelitian ini bertujuan untuk merancang dan membangun sistem informasi repositori skripsi berbasis web guna meningkatkan efisiensi dalam pengelolaan dan kemudahan akses terhadap koleksi skripsi di lingkungan Perpustakaan Universitas Prabumulih.	Penelitian ini mengembangkan sistem informasi repositori skripsi berbasis web untuk memudahkan pengelolaan dan akses skripsi di Perpustakaan Universitas Prabumulih.
2.	Andi ,dkk.[7]	2021	<i>Extreme Programming</i>	Penelitian ini bertujuan untuk mengembangkan sistem informasi berbasis web yang dapat digunakan untuk pengelolaan data kependudukan, pengaduan masyarakat, dan pelayanan surat administratif secara efektif di lingkungan Pemerintah Kabupaten Sukrame.	Penelitian ini menghasilkan sistem informasi yang dapat mengelola data kependudukan, pengaduan masyarakat, dan layanan surat administratif di Kabupaten Sukrame.

3.	Widiastuti[8]	2023	<i>Extreme Programming</i>	Penelitian ini bertujuan untuk merancang dan membangun sistem informasi kepegawaian yang dapat menggantikan proses manual, mengurangi penggunaan kertas, meningkatkan keamanan data melalui fitur <i>login</i> dan hak akses, serta menyajikan data kepegawaian yang lebih akurat dan terstruktur.	Hasil penelitian menunjukkan bahwa sistem informasi kepegawaian yang dibangun dapat mengatasi masalah pengelolaan data kepegawaian yang sebelumnya dilakukan secara manual. Sistem ini berhasil mengurangi penggunaan kertas, meningkatkan keamanan data dengan fitur <i>login</i> dan hak akses, serta menghasilkan informasi yang lebih akurat.
4.	Agung dan Sanusi. [11]	2023	<i>Extreme Programming</i>	Penelitian ini bertujuan untuk merancang dan membangun sistem informasi pengelolaan inventaris aset kantor yang mampu meningkatkan efisiensi dan efektivitas dibandingkan metode	Penelitian ini menghasilkan sistem pengelolaan inventaris aset kantor yang lebih efisien dibandingkan dengan metode pencatatan manual

				pencatatan manual berbasis Microsoft Excel.	menggunakan Microsoft Excel.
5.	Putra [10]	2019	Extreme Programming	Penelitian ini bertujuan untuk merancang dan membangun sistem informasi perpustakaan yang dapat menggantikan proses pengelolaan transaksi manual, sehingga meningkatkan efisiensi dan akurasi layanan perpustakaan di SMK 1 Muhammadiyah Malang.	Hasil dari penelitian menunjukkan bahwa sistem informasi perpustakaan yang dikembangkan berhasil mengatasi permasalahan pengelolaan transaksi secara manual di SMK 1 Muhammadiyah Malang.

2.2 Dasar Teori

2.2.1. Pengembangan *Website*

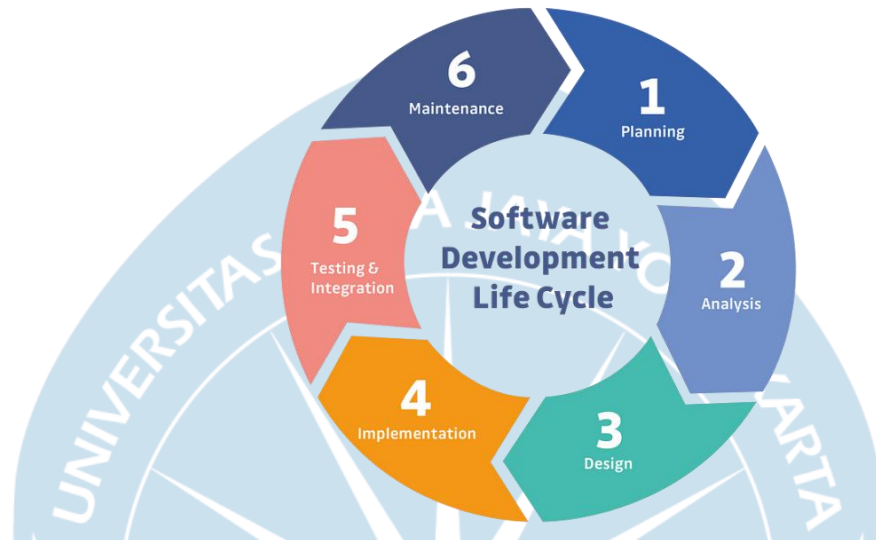
Pengembangan *website* melibatkan berbagai elemen seperti desain antarmuka pengguna (*UI*), pengalaman pengguna (*UX*), dan teknologi *backend* untuk menyediakan layanan yang efisien. Dalam konteks ini, penting untuk mempertimbangkan metodologi pengembangan yang efisien seperti *Agile (Extreme Programming)*, serta pemanfaatan *framework* modern untuk memastikan fleksibilitas dan skalabilitas aplikasi. *Framework* seperti Java Script memberikan kemudahan melalui struktur yang modular dan alat bantu bawaan, mempercepat proses pengembangan dan mengurangi kesalahan sistem.

2.2.2. Pengelolaan Aset

Aset yaitu suatu barang fisik maupun tidak berwujud yang dimiliki dan digunakan oleh orang atau organisasi untuk mencapai tujuannya[12]. Barang-barang tersebut yang

mempunyai nilai ekonomi, nilai komersial, dan nilai perdagangan yang dimiliki oleh orang perseorangan, badan swasta, atau pemerintah. Sedangkan pengelolaan aset merupakan suatu proses yang terstruktur meliputi seluruh aset yang berperan dalam memberikan dukungan dan menyediakan layanan untuk berlangsungnya perusahaan .

2.2.3 *Software Development Life Cycle*



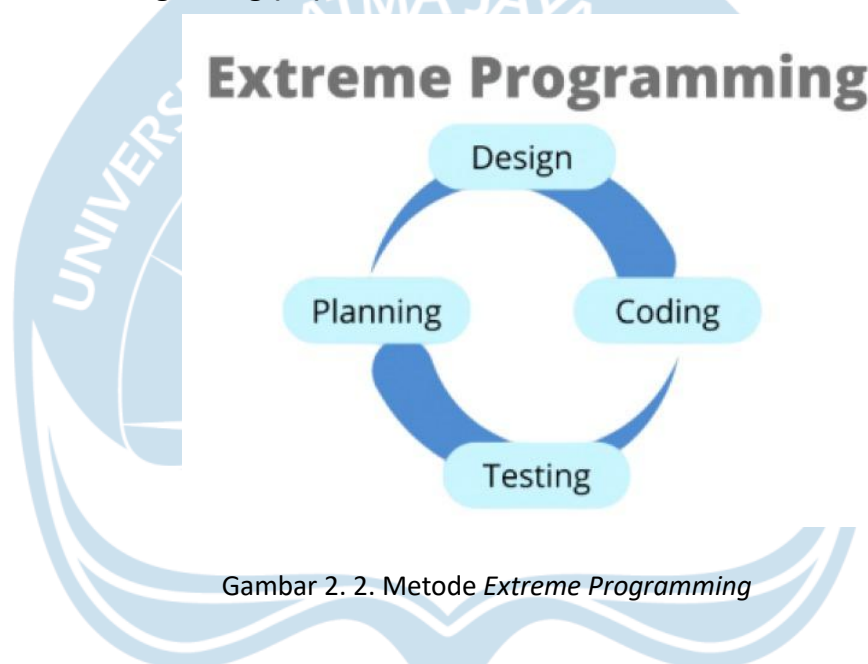
Gambar 2.1. *Software Development Life Cycle* [13]

Gambar 2.1 merupakan Siklus Pengembangan Perangkat Lunak (*Software Development Life Cycle* atau SDLC) rangkaian tahapan yang sistematis dalam pengembangan sebuah sistem, meliputi pra-produksi, fase produksi, hingga pasca-produksi. Setiap tahapan SDLC dirancang dengan tujuan spesifik untuk memastikan pengelolaan proyek yang efisien dan menghasilkan sistem yang memenuhi kebutuhan pengguna[14] .

Tahapan pertama adalah perencanaan, di mana ruang lingkup proyek ditentukan melalui pengumpulan data dari *stakeholder*. Pada tahap ini juga disusun struktur tim, jadwal, serta anggaran proyek. Selanjutnya, dalam fase analisis, tim mengidentifikasi kebutuhan fungsional sistem dan mengatasi potensi masalah yang dapat menghambat pengembangan . Pada tahap desain, dilakukan perancangan arsitektur sistem, termasuk antarmuka pengguna, serta pemilihan alat dan bahasa pemrograman yang akan digunakan. Fase berikutnya adalah pengembangan, yang mencakup penulisan dan implementasi kode berdasarkan rancangan yang telah disepakati. Setelah

pengembangan selesai, sistem masuk ke fase pengujian, di mana dilakukan evaluasi terhadap fungsionalitas sistem, sekaligus perbaikan terhadap kesalahan yang ditemukan. Setelah lulus pengujian, sistem diterapkan dalam lingkungan produksi pada tahap implementasi. Terakhir, fase pemeliharaan mencakup pemantauan dan pembaruan sistem untuk memastikan keberlangsungan fungsi yang optimal di masa mendatang. Dengan menjalankan siklus SDLC secara terstruktur, tim pengembang dapat menjaga kualitas hasil akhir sekaligus memastikan sistem mampu memenuhi kebutuhan pengguna dan beradaptasi terhadap perubahan yang mungkin terjadi di masa depan.

2.2.4 Extreme Programming (EP)



Gambar 2. 2. Metode *Extreme Programming*

Extreme Programming (XP) adalah salah satu metodologi dalam pendekatan *Agile* yang bertujuan untuk meningkatkan kualitas perangkat lunak sekaligus memungkinkan fleksibilitas tinggi dalam menghadapi perubahan kebutuhan pengguna [15].

XP mengadopsi siklus pengembangan perangkat lunak yang iteratif dan pendek, sehingga memungkinkan tim untuk menghasilkan perangkat lunak dalam waktu yang lebih singkat dan dapat diuji secara berkala. Metodologi ini menekankan pada kolaborasi intensif antara anggota tim dan pemangku kepentingan untuk memastikan bahwa perangkat lunak yang dikembangkan memenuhi kebutuhan pengguna dengan tepat.

Pendekatan XP sangat menekankan pada prinsip kesederhanaan (*simplicity*), di mana hanya fitur yang benar-benar diperlukan yang dikembangkan untuk mengurangi kompleksitas sistem. XP juga mengutamakan praktik pengujian intensif seperti *BlackBox*, yaitu pendekatan di mana pengujian dilakukan terlebih dahulu sebelum penulisan kode, sehingga setiap fungsi perangkat lunak dapat langsung diuji keberhasilannya. Selain itu, XP juga menerapkan *continuous integration*, yaitu penggabungan perubahan kode secara terus-menerus ke dalam sistem utama untuk mendeteksi potensi masalah sejak awal pengembangan [7] .

Dalam *Extreme Programming* (XP), pengembangan perangkat lunak dilakukan melalui serangkaian tahapan yang saling terkait dan bersifat iteratif. Empat tahapan utama dalam XP yaitu *Planning* (Perencanaan), *Design* (Desain), *Coding* (Pengkodean), dan *Testing* (Pengujian), masing-masing memiliki tujuan yang jelas dan saling melengkapi satu sama lain untuk menghasilkan perangkat lunak berkualitas tinggi yang responsif terhadap kebutuhan pengguna.

Pada fase *Planning*, tim pengembang dan pemangku kepentingan bersama-sama mendefinisikan ruang lingkup dan prioritas proyek. Perencanaan dalam XP tidak bersifat kaku, melainkan fleksibel, dengan fokus pada pemahaman kebutuhan pengguna dan perencanaan iterasi pengembangan yang dapat menyesuaikan dengan perubahan yang muncul. Proses ini berorientasi pada kolaborasi aktif antara tim pengembang dan pengguna untuk memastikan bahwa setiap iterasi pengembangan memenuhi ekspektasi dan kebutuhan yang ada .

Pada fase *Design*, dalam XP, desain tidak dibuat secara mendalam di awal, melainkan lebih kepada pembuatan solusi yang bisa langsung digunakan untuk memenuhi kebutuhan yang ada saat itu. Desain tersebut akan terus dievaluasi dan diperbaiki berdasarkan umpan balik yang diperoleh selama siklus pengembangan. Proses desain yang fleksibel ini memungkinkan pengembang untuk melakukan perubahan sesuai dengan hasil pengujian dan perkembangan kebutuhan pengguna .

Pada fase *Coding*, XP menerapkan praktik *pair programming*, yaitu dua pengembang bekerja bersama di satu stasiun kerja. Satu pengembang menulis kode sementara yang lain melakukan review secara langsung. Hal ini bertujuan untuk

memastikan kualitas kode dan mengurangi kemungkinan kesalahan sejak awal. Namun, jika dalam konteks tertentu hanya satu pengembang yang terlibat dalam pengkodean, proses tetap akan dilakukan dengan prinsip yang sama, yaitu menjaga kesederhanaan dan fokus pada kualitas kode. Pada tahap ini, pengembang akan menulis kode dengan perhatian penuh terhadap kebutuhan fungsional dan desain yang telah disepakati sebelumnya. Proses ini berlangsung dalam siklus pendek dengan prioritas pada pengembangan fitur yang paling dibutuhkan .

Pada fase *Testing*, pengujian difokuskan pada aspek fungsionalitas sistem secara keseluruhan, tanpa memperhatikan struktur internal atau kode sumber perangkat lunak. Pengujian ini dilakukan dengan memberikan *input* yang berbeda ke dalam sistem dan memeriksa apakah *output* yang dihasilkan sesuai dengan yang diharapkan, berdasarkan spesifikasi yang telah ditentukan. Metode ini lebih mengutamakan bagaimana perangkat lunak berinteraksi dengan pengguna atau komponen lain dalam sistem, bukan bagaimana perangkat lunak tersebut dibangun atau diimplementasikan di dalamnya. Oleh karena itu, black-box testing sangat berguna untuk menguji aspek fungsionalitas aplikasi secara *end-to-end* dan memastikan bahwa sistem berjalan sesuai dengan kebutuhan pengguna, tanpa harus mengetahui bagaimana bagian-bagian sistem bekerja di dalam .

2.2.5 Black Box Testing

Black-box Testing adalah metode pengujian perangkat lunak yang berfokus pada pengujian fungsionalitas sistem dari perspektif pengguna tanpa mengetahui atau memperhatikan implementasi internal perangkat lunak[16]. Pengujian ini dilakukan dengan memberikan *input* yang sesuai dan memverifikasi bahwa *output* yang dihasilkan oleh sistem sesuai dengan yang diharapkan berdasarkan spesifikasi yang telah ditentukan sebelumnya. Dalam pengujian ini, penguji tidak perlu mengetahui struktur kode atau algoritma yang digunakan dalam sistem, melainkan hanya berfokus pada hasil yang diberikan oleh sistem sebagai respons terhadap *input* yang diberikan.

Pada pengujian black-box, penguji akan memastikan bahwa perangkat lunak berfungsi dengan baik sesuai dengan kebutuhan fungsional pengguna dan dapat melakukan tugas-tugas yang diinginkan dengan benar. Pengujian ini sangat berguna untuk mengevaluasi apakah seluruh sistem beroperasi secara keseluruhan dengan baik,

terutama dalam skenario pengujian integrasi atau pengujian sistem yang melibatkan beberapa komponen yang saling berinteraksi.

Dalam konteks XP, black-box testing diterapkan pada tahap akhir pengembangan, di mana perangkat lunak diuji dalam keadaan yang lebih mendekati kondisi nyata. Hal ini bertujuan untuk memastikan bahwa sistem yang telah dikembangkan bekerja dengan baik dan memenuhi semua persyaratan pengguna yang telah didefinisikan di awal. Meskipun pengujian ini tidak memberikan *feedback* langsung pada level kode atau komponen individual, pengujian ini berperan penting untuk memvalidasi apakah perangkat lunak siap digunakan oleh pengguna akhir[17].

Tabel 2.2. Perbandingan Metode

Aspek	<i>Extreme Programming</i>	<i>V-Model</i>	<i>Waterfall</i>	<i>Scrum</i>
Pendekatan Pengembangan	Iteratif dan inkremental	Iteratif dan terstruktur,	Linear dan terstruktur	Iteratif dengan sprint
Pengujian	<i>Black-box Testing</i>	Pengujian terkait dengan tahap pengembangan, lebih terstruktur	Pengujian dilakukan setelah pengembangan selesai	Pengujian dilakukan selama setiap sprint
Fleksibilitas	Perubahan mengikuti <i>Client</i>	Kurang fleksibel, perubahan sulit dilakukan setelah tahap tertentu	Sulit melakukan perubahan setelah tahap tertentu	Fleksibel selama sprint
Kesesuaian dengan Proyek	Cocok untuk proyek dengan kebutuhan yang sering berubah	Cocok untuk proyek dengan kebutuhan stabil dan jelas	Cocok untuk proyek dengan kebutuhan stabil	Cocok untuk proyek yang berkembang dengan cepat

Pemilihan metode *Extreme Programming* (XP) dalam penelitian ini didasarkan pada keunggulannya yang signifikan dalam mendukung pengembangan perangkat lunak secara iteratif dan berkelanjutan. XP menekankan kolaborasi yang intens antara pengembang dan klien, serta penerapan pengujian yang terus-menerus, yang memungkinkan deteksi masalah lebih cepat dan perbaikan yang lebih efisien. Metode

ini juga fleksibel terhadap perubahan, di mana setiap iterasi memungkinkan penyesuaian fungsionalitas sesuai dengan kebutuhan pengguna yang terus berkembang.

