

Penerapan Metodologi Rekayasa Perangkat Lunak untuk Efisiensi Pengembangan Sistem

Diterima:
26 November 2024
Revisi:
21 Desember 2024
Terbit:
31 Desember 2024

^aSafa Nadia Bakri, ^aMuhammad Irwan Padli Nasution
^aUniversitas Islam Negeri Sumatera Utara

Abstrak—Latar Belakang: Dalam pengembangan perangkat lunak, tantangan dalam hal efisiensi dan efektivitas sering kali muncul, terutama ketika menghadapi keterbatasan sumber daya untuk memenuhi tujuan kebutuhan pengguna. **Tujuan:** Studi ini bertujuan untuk menyelidiki berbagai pendekatan untuk pengembangan perangkat lunak, termasuk berbagai pendekatan, seperti Waterfall, Prototyping, dan Quick Application Improvement (RAD), dengan penekanan pada peningkatan efisiensi dalam proses pengembangan. **Metode:** Dengan menggunakan metode Waterfall, Prototyping, Rapid Application Development (RAD), dan melalui analisis kualitatif, penelitian ini dapat membedakan preferensi dan kekurangan dari masing-masing strategi yang dapat dianalisis berdasarkan studi literatur yang ada. **Hasil:** Metode Waterfall menawarkan kemudahan dalam manajemen proyek, meskipun memiliki fleksibilitas yang terbatas. Di sisi lain, prototyping mendukung identifikasi kebutuhan yang berulang, namun cenderung mahal. Sementara itu, pendekatan Rapid Application Development (RAD) dapat mempercepat proses pengembangan, tetapi kurang ideal untuk proyek-proyek berisiko tinggi. **Kesimpulan:** Pemilihan metodologi yang tepat harus diselaraskan dengan kebutuhan spesifik dari pengembangan yang dilakukan untuk mengoptimalkan efisiensi dan kualitas program yang dibuat.

Kata Kunci—Waterfall; Prototyping; RAD; Efisiensi

Abstract— Background: In software development, challenges in terms of efficiency and effectiveness often arise, especially when facing resource limitations to meet user needs. **objectives:** This study aims to investigate various approaches to software development, including various approaches, such as Waterfall, Prototyping, and Rapid Application Improvement (RAD), with an emphasis on improving efficiency in the development process. **Methods:** By using Waterfall, Prototyping, Rapid Application Development (RAD) methods, and through qualitative analysis, this study can distinguish the preferences and shortcomings of each strategy that can be analyzed based on the existing literature study. **Results:** The Waterfall method offers convenience in project management, although it has limited flexibility. On the other hand, prototyping supports iterative requirement identification, but tends to be expensive. Meanwhile, the Rapid Application Development (RAD) approach can speed up the development process, but is less ideal for high-risk projects. **Conclusion:** The selection of the right methodology should be aligned with the specific needs of the development to optimize the efficiency and quality of the program.

Keywords—Waterfall; Prototyping; RAD; Efficiency

This is an open access article under the CC BY-SA License.



Penulis Korespondensi:

Safa Nadia Bakri,
Sistem Informasi,
Universitas Islam Negeri Sumatera Utara,
Email: safanadia278@gmail.com

I. PENDAHULUAN

SDLC atau yang biasa disebut dengan (*Software Development Life Cycle*), merupakan model dan rangkaian proses yang diterapkan dalam pengembangan perangkat lunak. Proses ini menggambarkan perjalanan para pengembang, mulai dari identifikasi masalah hingga pencarian solusi yang tepat. Meskipun pengembangan sistem informasi dan rekayasa perangkat lunak yang menciptakan sistem atau perangkat lunak yang sepenuhnya baru, dan seringkali biasanya terjadi adalah penyempurnaan dari sistem yang telah ada [1]. Namun, terdapat masih adanya kesenjangan dalam penelitian terkait analisis komparatif yang mendalam antara ketiga metodologi tersebut. Hal ini terutama berkaitan dengan efisiensi, fleksibilitas, dan relevansi metode terhadap kebutuhan spesifik suatu proyek.

Perangkat lunak merupakan kumpulan instruksi yang dirancang untuk memproses informasi. Terdapat dua bentuk utama perangkat lunak yakni program dan prosedur. Program merupakan serangkaian perintah yang dapat dipahami oleh komputer, sementara itu prosedur merupakan perintah yang diperlukan oleh pengguna untuk mengelola dan memproses suatu informasi [2]. Rekayasa perangkat lunak berfokus pada pemanfaatan berbagai pendekatan, model, proses, dan metode untuk merancang dan mengembangkan solusi teknologi bersama sekelompok orang tertentu. Ini mencakup entitas yang sedang berkembang serta mereka yang berupaya membangun solusi teknologi untuk kelompok tertentu. Dengan demikian, tujuan utamanya adalah menghadirkan metodologi inovatif yang telah teruji cukup efektif dalam pengembangan dan penerapan perangkat lunak sepanjang perjalanan tersebut [3].

Pengembangan sistem perangkat lunak yang efektif dan efisien menjadi tantangan utama pada perkembangan teknologi seperti sekarang adalah kebutuhan akan perangkat lunak berkualitas tinggi yang dapat dikembangkan dengan cepat dan efisien dari segi biaya terjangkau semakin menekankan pentingnya penerapan metodologi rekayasa perangkat lunak yang tepat. Dengan menentukan sebuah metodologi, kita dapat menciptakan dinamika cukup tinggi pada setiap tahap perancangan model mencerminkan aktivitas dan siklus hidup suatu sistem. Metodologi Pengembangan Perangkat Lunak dapat diartikan sebagai proses yang bertujuan untuk merancang perangkat lunak baru yang sepenuhnya menggantikan perangkat lunak lama, atau melakukan perbaikan pada perangkat lunak yang telah ada. Penerapan metodologi ini adalah aspek yang sangat penting. Untuk mempercepat dan memperjelas penggambaran solusi serta pengembangan perangkat lunak itu sendiri. Melalui metodologi ini, kita berpotensi untuk menghasilkan perangkat lunak berkualitas yang lebih baik [4].

Pada kajian penelitian terdahulu, penelitian yang mengkaji mengenai Darmawan Setiya Budi, Taghfirul Azhima Yoga Siswa, dan Heri Abijono: Analisis Pemilihan Penerapan Proyek

Metodologi Pengembangan Perangkat Lunak, pada tahun 2015, Menjelaskan dan menganalisis berbagai metodologi pengembangan perangkat lunak seperti Model *Waterfall*, Model *Paralel*, Model *Iteratif*, Model *Prototype*, *Rapid Application Development* (RAD), Model *Spiral*, Model *V*, dan Pengembangan *Agile*. Tujuan dari analisis ini adalah untuk membandingkan kelebihan dan kekurangan setiap metode, terutama yang berkaitan dengan efisiensi pengembangan sistem [5].

Pada penelitian yang di kaji oleh Diandara Tresya Haniva, Jadid Alif Ramadhan, dan Aries Suharso pada tahun 2023 mengenai *Systematic Literature Review* Penggunaan Metodologi Pengembangan Sistem Informasi *Waterfall*, *Agile*, dan *Hybrid* menjelaskan tentang perbandingan menggunakan metode pengembangan sistem informasi seperti *waterfall*, *agile* dan *hybrid* untuk pengembangan sistem informasi. Dalam studi ini, mengkaji lingkungan proyek yang cocok untuk setiap metodologi guna meningkatkan efisiensi pengembangan sistem [6].

Pada penelitian yang di kaji oleh Subhiyanto pada tahun 2023 mengenai Analisis Metode Pengembangan Sistem Informasi Manajemen : *A Systematic Literatur Review* juga menjelaskan mengenai Melalui tinjauan literatur yang sistematis, metode untuk mengembangkan sistem informasi manajemen dianalisis. Ada penekanan pada metodologi "*Waterfall*", "*Agile*" dan "*RAD*". Tujuannya adalah untuk mengidentifikasi kelebihan dan kekurangan serta dampak penggunaannya dalam rangka meningkatkan efisiensi pengembangan sistem informasi manajemen [7].

Perbedaan penelitian ini dari penelitian sebelumnya adalah bahwa karena mengintegrasikan evaluasi kelebihan dan kekurangan dari ketiga metodologi utama secara bersamaan, dengan pendekatan yang menekankan pentingnya adaptasi metode sesuai dengan karakteristik proyek yang sedang dijalankan. Pada penelitian sebelumnya perbedaan terdapat pada pendekatan yang Bersifat holistik, yaitu menganalisis secara rinci kelebihan dan kekurangan ketiga metode secara bersamaan, dengan penekanan pada relevansi penyesuaian metodologi dengan persyaratan spesifik suatu proyek.

Penelitian ini bertujuan untuk mengevaluasi metodologi *Waterfall*, *Prototyping*, dan *Rapid Application Development* (RAD) dalam konteks pengembangan perangkat lunak. Fokus utama dari penelitian ini adalah untuk mengungkap efisiensi, fleksibilitas, serta praktik terbaik yang dapat diterapkan dalam berbagai skenario proyek.

Keberhasilan dalam melakukan pengembangan perangkat lunak sangat dipegaruhi oleh pengelolaan proyek yang dilakukan secara menyeluruh. Pengelolaan ini mencakup mengenai perencanaan, pengorganisasian, pelaksanaan, dan evaluasi yang dilakukan dengan tujuan utama dari pengelolaan proyek perangkat lunak adalah agar memastikan produk yang dihasilkan telah memenuhi kebutuhan pengguna, serta selesai sesuai anggaran dan jadwal yang ditentukan. Salah

satu aspek kunci dalam mencapai hal tersebut adalah pemilihan metodologi yang tepat dalam proses pengembangan. Metodologi ini berfungsi sebagai kerangka kerja yang mengatur dan mengarahkan aktivitas selama proses pengembangan. Fleksibilitas dalam penerapan metodologi memungkinkan penyesuaian terhadap tantangan atau perubahan kebutuhan yang muncul di setiap tahap siklus hidup perangkat lunak, Dimulai dari menganalisis kebutuhan, serta dilanjutkan dengan desain sistem, implementasi, pengujian, dan diakhiri dengan pemeliharaan. Oleh karena itu, pemilihan dan penerapan metodologi yang sesuai tidak hanya berkontribusi pada kualitas perangkat lunak, tetapi juga meningkatkan efisiensi dan efektivitas tim pengembang [5].

II. METODE

Dalam studi ini, penulis menggunakan dan membandingkan data dari tinjauan jurnal. Dengan kata lain, penulis menggunakan metode penelitian deskriptif. Metode deskriptif merupakan metode penelitian yang digunakan untuk mempelajari keadaan sekelompok orang, suatu objek, suatu sistem pemikiran, serangkaian kondisi, atau serangkaian kejadian pada saat ini. Tujuan penelitian deskriptif adalah untuk menggambarkan, menjelaskan, atau mengilustrasikan fakta, karakteristik, dan hubungan fenomena yang sedang dipelajari secara sistematis, objektif, dan akurat[8].

Sebagai perbandingan, penulis memilih beberapa metode yang diperlukan untuk pengembangan perangkat lunak antara lain:

1. Waterfall
2. Prototype
3. RAD (*Rapid Application Development*)

Metodologi penelitian ini bisa dipahami sebagai pendekatan ilmiah dalam pengumpulan data dengan tujuan dan manfaat tertentu. Dalam hal ini, terdapat empat kata kunci penting yang perlu diperhatikan: cara ilmiah, data, tujuan, dan kegunaan, yang semuanya dijelaskan dalam jurnal [9].

Data ini kemudian dianalisis untuk memahami berbagai model metodologi seperti *Waterfall*, *Prototyping*, RAD, dan kombinasi metode lainnya serta dampaknya terhadap efisiensi pengembangan perangkat lunak. Proses analisis data dilakukan secara kualitatif dengan mengidentifikasi, mengkategorikan dan menganalisis informasi yang relevan dari literatur. Tujuan dari langkah ini ialah untuk mengevaluasi kekuatan dan kelemahan masing-masing metodologi dan menilai efektivitasnya dalam mengatasi tantangan pengembangan perangkat lunak [10].

III. HASIL DAN PEMBAHASAN

Dalam penelitian ini ditemukan bahwa pemilihan metodologi pengembangan perangkat lunak yang tepat, seperti *Waterfall*, *Prototyping*, dan *Rapid Application Development (RAD)*, memiliki dampak yang signifikan terhadap efisiensi dan kualitas pengembangan perangkat lunak. Setiap metodologi memiliki kelebihan dan kekurangan yang harus dipertimbangkan sesuai dengan kebutuhan proyek tertentu.

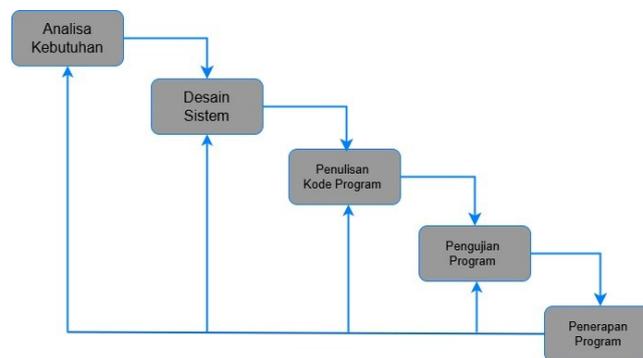
Konsep (SDLC) atau yang biasa disebut dengan siklus hidup pengembangan sistem, berkaitan erat dengan bidang komputer dan sistem informasi. SDLC berfungsi sebagai kerangka kerja dalam pengembangan sistem perangkat lunak, Untuk mengembangkan sistem perangkat lunak, terdapat beberapa fase penting yang harus dilalui, yaitu ppenyusunan, penelitian, rancangan, penerapan, uji coba, dan pengelolaan. Ada tiga metode utama yang sering diterapkan dalam siklus hidup sistem, yaitu siklus hidup sistem tradisional, siklus hidup berbasis prototipe, dan siklus hidup yang berorientasi objek. Dalam konteks rekayasa sistem dan perangkat lunak, SDLC merupakan proses yang mencakup pembuatan serta modifikasi sistem dilengkapi dengan bentuk dan metode yang mendukung pengembangan proses tersebut. Secara keseluruhan, konsep ini mengacu pada sistem komputer atau informasi [11]. Menurut [12], Bagian dari metodologi pengembangan software dapat dikelompokkan menjadi tiga unit utama. Yang pertama adalah Metode, yang merujuk pada pendekatan atau teknik sistematis yang digunakan dalam proses pengembangan software.

Metode ini mencakup berbagai aspek, mulai dari perencanaan dan estimasi proyek hingga menganalisis persyaratan dalam pengembangan software, terdapat beberapa aspek penting yang harus dipertimbangkan, yakni seperti sistem dan perangkat lunak, perancangan struktur data, arsitektur program, prosedur algoritma, penulisan kode, pengujian, dan pemeliharaan. Selain itu, ada juga alat bantu yang mendukung proses tersebut, baik yang bersifat manual maupun otomatis. Alat bantu ini dapat dibedakan menjadi dua kategori: alat bantu manual dan alat bantu otomatis. Ketiga, ada Prosedur yang mendefinisikan urutan pekerjaan atau alur dari metode dan alat yang telah disebutkan sebelumnya sangatlah penting. Perlu dicatat juga bahwa proyek pengembangan perangkat lunak sering kali dihadapkan pada spesifikasi yang terus berubah, dinamika teknologi dan standar yang tinggi, serta kebutuhan akan tenaga kerja yang terampil dan kolaborasi dalam tim yang solid.

Dalam penelitian ini, penulis memilih beberapa metode pengembangan sebagai objek studi untuk dilakukan perbandingan, di antaranya:

A. Waterfall

Metode *Waterfall* merupakan metode pengembangan software yang paling banyak digunakan. Metode ini bersifat linier, memulai proses dari tahap perencanaan hingga pemeliharaan, yang merupakan langkah terakhir dalam pengembangan sistem. Sebelum melanjutkan ke tahap berikutnya, setiap tahap harus diselesaikan, tanpa adanya kemungkinan untuk kembali atau mengulang tahap yang telah dilalui[8]. Metode waterfall SDLC, juga dikenal sebagai pendekatan klasik atau model sekuensial linier, memberikan struktur yang jelas dalam proses pengembangan perangkat lunak. Proses ini dimulai dengan analisis, dilanjutkan dengan rancangan, code, uji coba, dan akhirnya tahap dukungan. Berikut Adalah tahapan atau langkah-langkah dalam model *Waterfall*:



Gambar 1. Langkah Model *Waterfall*

[8]

Pada gambar 1. menunjukkan model air terjun pengembangan software atau perangkat lunak, salah satu pendekatan klasik dalam rekayasa perangkat lunak. Model ini menunjukkan fase-fase terstruktur dari proses pengembangan perangkat lunak, dengan setiap langkah dilakukan secara berurutan. Langkah-langkah individual dijelaskan di bawah ini[13]:

1. Analisa Kebutuhan

Pada fase ini, peningkatan sistem harus dapat berkomunikasi. dengan pengguna agar dapat memahami harapan mereka dan keterbatasan perangkat lunak. Informasi dapat diperoleh melalui wawancara, diskusi, atau investigasi langsung. Informasi diurai untuk mendapatkan data yang dibutuhkan pengguna.

2. Desain Sistem

Pada fase ini, pengembang membuat desain sistem. Ini akan membantu menentukan persyaratan perangkat keras dan sistem dan juga membantu mendefinisikan arsitektur sistem secara keseluruhan.

3. Penulisan Kode Program

Pada fase ini, pengembang mulai menerjemahkan desain sistem ke dalam kode program yang ditulis menggunakan bahasa pemrograman tertentu. Proses ini dilakukan menggunakan bahasa

pemrograman berdasarkan kebutuhan proyek. Pengembang bertanggung jawab memastikan semua bagian desain sistem diimplementasikan dengan benar dalam kode dan perangkat lunak yang dihasilkan berfungsi sebagaimana mestinya.

4. Pengujian Program

Pada fase ini, sistem diperiksa dan diuji untuk melihat apakah sistem memenuhi persyaratan sistem secara penuh atau sebagian. Pengujian dapat berupa pengujian unit (dijalankan pada modul kode tertentu), pengujian sistem (memeriksa bagaimana sistem merespons ketika semua modul terintegrasi), pengujian penerimaan (menggunakan atau nama pelanggan) dan menjalankan pengujian untuk memastikan semua pelanggan persyaratannya terpenuhi. puas.

5. Penerapan Program

Tahap terakhir adalah implementasi atau penerapan program di lingkungan pengguna. Sistem yang diuji dipasang dan dioperasikan oleh pengguna akhir. Lalu masing-masing bagian akan dilakukan pengembangan serta diuji untuk fungsionalitasnya atau biasa dikenal dengan pengujian unit.

Berikut adalah kelebihan dan kekurangan dari metode *Waterfall* [14]

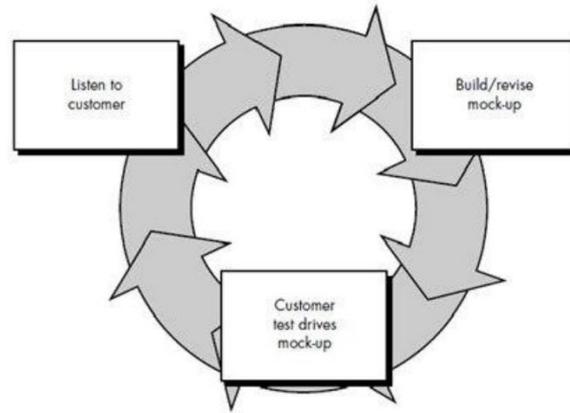
Tabel 1. Metode *Waterfall* Kelemahan dan Kelebihan

Kelebihan	Kelemahan
Pengelolaan dapat menjadi lebih mudah dikarenakan hampir semua kebutuhan telah ditemukan dan dicatat dengan baik.	Langkah yang dilakukan berurutan secara linier tidak memungkinkan adanya untuk kembali kepada langkah sebelumnya.
Tahapan yang dilakukan secara berurutan dan linier serta diidentifikasi dan dokumentasi yang dilakukan secara menyeluruh.	Masih kurangnya kemampuan untuk dapat beradaptasi dengan setiap perubahan yang terjadi selama tahap pengembangan sistem.
Lebih baik untuk produk perangkat lunak/program yang persyaratannya jelas sejak awal dan kesalahan diminimalkan.	Hampir tidak memiliki toleransi terhadap setiap kesalahan yang telah dilakukan, terutama dalam tahap perencanaan dan desain.
Metode ini biasanya menghasilkan perangkat lunak berkualitas tinggi.	Customer atau pengguna cukup sulit untuk dapat mengalami perubahan kebutuhan sesuai dengan yang diinginkan.

B. Model Prototype

Model *Prototyping* adalah suatu teknik yang memungkinkan pengumpulan informasi secara cepat mengenai kebutuhan pengguna. Pendekatan ini memfokuskan pada penyajian berbagai aspek dari software yang akan terlihat oleh pengguna. Prototipe yang dihasilkan kemudian dinilai oleh pelanggan untuk mengidentifikasi dan menyeleksi kebutuhan dalam peningkatan software. Prototipe dapat dipahami sebagai alat yang mengajarkan calon pembuat dan pengguna tentang cara sistem bekerja lebih utuh. Proses ini digunakan untuk menghasilkan prototipe ini dikenal dengan sebutan *prototyping*. Seperti terlihat pada gambar 2, bentuk prototipe tersebut

mencerminkan versi awal dari tahap-tahap pengembangan sistem perangkat lunak, yang berfungsi untuk mempresentasikan ide, melakukan eksperimen pada desain, serta mencari dan menyelesaikan berbagai masalah yang mungkin muncul. Dengan menggunakan model *prototyping*, pengguna dapat memahami bagaimana tahapan sistem dibangun, sehingga sistem dapat beroperasi dengan baik [15].



Gambar 2. Model *Prototype* [8]

Kelebihan dan Kelemahan Metode *Prototype* adalah sebagai berikut:[16]

Tabel 2. Kelebihan dan Kelemahan Metode *Prototype*

Kelebihan	Kelemahan
Karena kebutuhan telah diidentifikasi dan didokumentasikan dengan baik, evaluasi berkala dan masukan dari pemilik proyek terkait membantu pengelolaan.	Kesalahan dan redundansi dapat dikurangi dengan cara melakukan identifikasi yang tepat terhadap purwa rupa.
Peningkatan pengalaman pengguna terjadi secara bertahap, berkat adanya upaya yang dilakukan secara terus menerus agar dapat menguji dan mengevaluasi bentuk. Tahapan yang dilakukan secara beruntun dan linier serta identifikasi dan dokumentasi dengan menyeluruh.	Akan ada kebutuhan biaya tambahan terkait dengan pembuatan <i>prototype</i> . Dan juga penyesuaian terhadap versi <i>prototype</i> , dapat dilakukan sesuai kebutuhan hingga akhirnya disetujui oleh pemilik proyek.
Kesalahan dan redundansi yang telah dilakukan dapat diminimalkan melalui identifikasi yang baik terhadap <i>prototype</i> .	Dapat memberikan beban tambahan kepada <i>programmer</i> .
Dengan metode ini pengguna dapat lebih mudah untuk menentukan keinginannya.	Biasanya metode ini kurang fleksibel dalam menghadapi segala jenis perubahan.

C. Rapid Application Development (RAD)

Metodologi ini adalah pendekatan bertahap dalam pengembangan sistem melalui perangkat lunak, dengan penekanan pada kemajuan di setiap tahap. Ia memecah setiap fase menjadi bagian-bagian yang lebih kecil, dan setiap prosesnya mengikuti prinsip yang sama seperti metode air

terjun (*waterfall*). Tujuan dari metodologi ini adalah mengembangkan sistem secara bertahap, meskipun perencanaan awal dilakukan dengan keseluruhan. (RAD) *Rapid Application Development* menekankan tentang kecepatan dan akurasi dalam setiap tahapan pembangunan, karena kedua aspek ini sangat penting dalam metodologi ini. Metode RAD mengintegrasikan prinsip-prinsip dari metode *waterfall* dalam proses pengembangan sistem, dengan fokus pada efisiensi dan responsivitas terhadap kebutuhan pengguna[17]. Pengembangan Aplikasi Cepat (RAD) *Rapid Application Development* menekankan pada tahap pembuatan yang cepat dan akurat, karena aspek ini sangat krusial dalam metodologi ini.

Metode RAD mengadopsi sistem dari metode *Waterfall* dalam proses pembangunan sebuah sistem dengan cara:

1. Komponen berbasis konstruksi (pemrograman berbasis komponen).
2. Melibatkan penggunaan kembali bagian perangkat lunak lama.
3. Metode ini mendorong peningkatan otomatisasi kode, sehingga mempercepat dan mempermudah proses pembuatan perangkat lunak.

Untuk menyelesaikan tugas yang sebanding namun berbeda, diperlukan kolaborasi dari berbagai tim. Berikut adalah langkah-langkah dalam proses *Rapid Application Development* (RAD):

1. Business Modelling

Tujuan dari proses ini adalah untuk memahami kebutuhan pengguna dan mengatur proses bisnis. Pada titik ini, tim akan memutuskan sistem apa yang akan dibuat, sumber data apa yang diperlukan, dan siapa yang memiliki wewenang untuk mengelola informasi tersebut.

2. Documentation Modelling

Pada tahap ini, fokus dilakukan untuk mengidentifikasi data yang diperlukan dan yang tersedia untuk dikembangkan menjadi sebuah prototipe yang sesuai dengan harapan pengguna.

3. Process Modelling

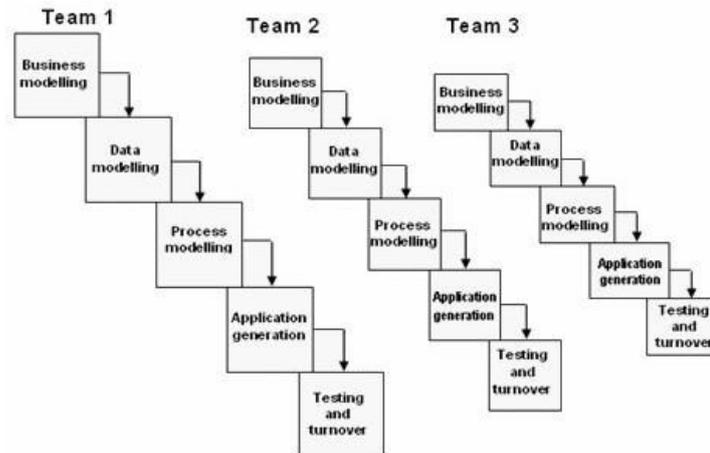
Tujuan dari prosedur ini adalah untuk mendefinisikan dokumentasi yang telah dibuat sebelumnya dan melakukan modifikasi agar dapat memenuhi kebutuhan pengguna dalam konteks proses bisnis.

4. Application Generation Rapid Application Development (RAD)

Menggunakan pendekatan generasikeempat, yang menekankan pemanfaatan komponen alat yang sudah ada. Jika diperlukan, tim juga dapat menciptakan komponen baru. Sebagian besar kasus menggunakan alat bantu otomatis untuk membantu dalam membangun perangkat lunak.

5. Testing dan Turnover

Untuk mempercepat waktu pengujian, proses ini memanfaatkan kembali komponen dari program yang telah diuji sebelumnya. Namun, komponen yang masih baru harus diuji secara menyeluruh.



Gambar 3. Model *Rapid Application Development (RAD)* [17]

Kelebihan dan Kelemahan Metode *Rapid Application Development (RAD)*[18].

Tabel 3. Kelebihan dan Kelemahan Metode *Rapid Application Development (RAD)*

Kelebihan	Kelemahan
Jika dibandingkan dengan metode pengembangan lainnya, model ini lebih cocok untuk siklus pengembangan cepat dan kualitasnya sangat baik.	Lebih berfokus terhadap proses pengembangan lanjutan dari pada menekankan kepada perencanaan tugas.
Dengan memanfaatkan sistem yang baru diperoleh akan menghemat lebih banyak uang daripada membuatnya sendiri.	Model RAD tidak sesuai untuk sistem dengan resiko yang sangat tinggi.
Model RAD ini memiliki kemampuan untuk beradaptasi dengan sistem yang sudah ada sebelumnya, sehingga dapat memudahkan apabila ingin melakukan penyesuaian dalam komponennya.	Dalam proses upaya pengembangan perangkat lunak atau aplikasi, tidak semua proyek cocok untuk menggunakan <i>Rapid Application Development (RAD)</i> . Karena disebabkan oleh fakta bahwa jika modul-modul dalam sistem atau aplikasi tidak dikelola dengan baik, penerapan model RAD dapat menimbulkan masalah.

Jurnal ini membahas analisis berbagai metodologi pengembangan perangkat lunak, termasuk Waterfall, Prototype, Iterative, Spiral, dan RAD. Tujuannya adalah untuk memberikan panduan untuk memilih metodologi sesuai dengan kebutuhan proyek, dengan mempertimbangkan kekurangan dan kelebihan dari masing-masing metodologi [19]. Temuan penelitian ini konsisten dengan penelitian[5], yang menunjukkan bahwa metodologi waterfall cocok digunakan untuk proyek dengan persyaratan awal yang stabil, namun memiliki kelemahan pada fleksibilitasnya. Hasil ini mendukung kesimpulan bahwa metodologi waterfall efektif untuk manajemen proyek tetapi tidak ideal untuk proyek yang dinamis.

Studi ini juga didukung oleh[6]. melalui tinjauan literatur sistematis, menyoroti pentingnya memilih metodologi yang tepat untuk lingkungan proyek untuk meningkatkan efisiensi pengembangan sistem. Ini mendukung temuan kami bahwa pilihan metodologi pengembangan perlu disesuaikan dengan persyaratan spesifik proyek. Bagian fase pengembangan untuk mengevaluasi metodologi pengembangan perangkat lunak yang dipilih. Terdiri dari kriteria yang secara umum merujuk pada tahap-tahap perkembangan. Meliputi lima fase utama metodologi pengembangan yaitu perencanaan, analisis, desain, implementasi dan pemeliharaan, disajikan dalam format tabel.

Tabel 4. Perbandingan Pengembangan Metode Perangkat Lunak [20]

Tahapan Pengembangan Perangkat Lunak	Waterfall	Prototype	RAD
Perencanaan Sistem (<i>System Planning</i>)	Tahap perencanaan di mulai dari awal.	Perencanaan berawal dari kebutuhan awal.	Perencanaan di mulai dari kebutuhan awal user.
Analisis Sistem (<i>System Analysis</i>)	Untuk dapat terhindar dari kesalahan pada saat membuat sistem, pertama-tama anda harus menganalisis kebutuhan data Anda secara akurat.	Selama melakukan persyaratan data harus bertambah atau berkurang kebutuhan pengguna.	Data yang didapat kemudian disesuaikan dengan kebutuhan pengguna ketika ingin melakukan percobaan.
	Jika ingin mengubah data atau fungsinya, maka seluruh Langkah atau proses berikutnya akan berubah.	Perubahan pada tahap ini dapat diterapkan pada perangkat lunak atau sistem saat masih dalam tahap prototipe.	Kebutuhan yang ada pada fungsional utama dapat distandarisasi dan juga dilaksanakan secara independen dalam jangka waktu yang ditentukan oleh tim.
Implementasi Sistem (<i>Systems Implementation</i>)	Tahapan perancangan dalam model waterfall ini dilakukan setelah tahap analisis, dan biasanya setelah itu akan	Pada tahapan ini tidak mengimplementasikan proses perancangan yang efektif.	Pada tahap ini proses perencanaan tidak dapat dilaksanakan dengan baik jika dilakukan secara terburu-buru.

	menghasilkan rencana yang sangat mendetail.		
	Pada saat perangkat lunak telah selesai maka dilakukan evaluasi.	Evaluasi dilaksanakan pada saat prototipe selesai dibangun.	Pada saat perangkat lunak telah selesai di bangun maka dapat di lakukam evaluasi.
Pemeliharaan Sistem (Sistem Maintenance)	Pemeliharaan dalam metode Waterfall ini cenderung kearah yang lebih mudah karena awal pengembangan, semua persyaratan telah ditentukan dengan jelas.	Pemeliharaan pada metode Prototype lebih mudah dilakukan, karena perubahan diidentifikasi sebelum tahap pengembangan telah dimulai.	Proses pemeliharaan pada metode ini mengikuti perjanjian kerja yang telah disepakati antara user atau pengguna dan pengembang.

IV. KESIMPULAN

Setiap metodologi memiliki kelebihan dan kekurangan yang dipengaruhi oleh kompleksitas dan kebutuhan proyek. Memilih metodologi yang tepat dapat berkontribusi pada peningkatan kualitas perangkat lunak dan efisiensi tim pengembang. Oleh karena itu, studi tambahan diperlukan untuk mempelajari gabungan metode yang dapat memenuhi kebutuhan yang terus berubah. Memilih metodologi pengembangan perangkat lunak yang tepat memiliki dampak yang signifikan terhadap keberhasilan proyek Anda. Setiap metodologi memiliki karakteristik unik yang disesuaikan dengan kebutuhan spesifik. Dengan pendekatan liniernya, metode Waterfall sangat ideal untuk proyek yang memiliki persyaratan yang jelas dan stabil di awal, namun kurang fleksibel untuk berubah seiring berjalannya proses pengembangan. Pembuatan prototipe memungkinkan spesifikasi persyaratan yang lebih tepat melalui iterasi dan masukan pengguna, namun memerlukan biaya pembuatan prototipe tambahan. Rapid Application Development (RAD) dicirikan oleh kecepatan dan fleksibilitas pengembangan, namun tidak ideal untuk proyek berisiko tinggi atau sistem yang kompleks.

Menurut penelitian ini, tidak ada strategi yang dapat digunakan secara universal untuk semua situasi. Saat memilih metodologi, Anda harus mempertimbangkan faktor-faktor seperti kebutuhan pengguna, anggaran, sumber daya yang tersedia, tingkat risiko, dan kompleksitas proyek. Dengan memahami kekuatan dan kelemahan masing-masing metodologi, tim pengembangan dapat menyesuaikan pendekatan mereka untuk meningkatkan efisiensi, mengoptimalkan kualitas perangkat lunak, serta menyelesaikan proyek sesuai anggaran dan tepat waktu. Penelitian lebih lanjut direkomendasikan dengan mempertimbangkan penggabungan beberapa metode untuk menciptakan pendekatan yang lebih adaptif dan efektif untuk mengatasi tantangan pengembangan

perangkat lunak di masa. Beberapa keterbatasan studi ini perlu dipertimbangkan. Pertama, analisis yang dilakukan hanya mencakup tiga metodologi utama: waterfall, prototyping, dan rapid application development (RAD), dan tidak mencakup metodologi yang lebih modern seperti agile dan hybrid, yang semakin populer. Tidak demikian. Kedua, ini adalah studi kualitatif yang menggunakan data dari tinjauan pustaka, jadi tidak ada studi kasus atau validasi eksperimen langsung yang dilakukan. Ketiga, penelitian ini berfokus pada aspek efisiensi tanpa menggali lebih dalam dampak metodologi terhadap kualitas perangkat lunak dari perspektif pengguna akhir. Selain itu, keterbatasan waktu dan sumber daya juga menjadi kendala dalam mengintegrasikan evaluasi ke dalam proyek yang lebih besar atau lebih kompleks. Disarankan untuk melakukan penelitian lebih lanjut tentang kombinasi metodologi yang lebih adaptif dengan mempertimbangkan aspek seperti skala proyek, dinamika kebutuhan pengguna, dan integrasi teknologi terbaru seperti kecerdasan buatan atau pengembangan berbasis cloud. Selain itu, evaluasi kuantitatif terhadap efisiensi dan kualitas perangkat lunak yang dibuat dengan metode gabungan dapat menjadi topik yang relevan untuk penelitian lebih lanjut.

DAFTAR PUSTAKA

- [1] A. S. Sofyan, P. Puspitorini, and M. A. Yulianto, "Aplikasi Media Informasi Sekolah Berbasis SMS Gateway Dengan Metode SDLC," *J. Sisfotek Glob.*, vol. 6, no. 2, pp. 5–12, 2016.
- [2] M. B. Rosyadi, "Pengembangan Aplikasi Mobile Learning Untuk Meningkatkan Kemandirian Dan Prestasi Belajar Pada Mata Kuliah Rekayasa Perangkat Lunak," *J. Ilm. Mandala Educ.*, vol. 8, no. 1, pp. 468–477, 2022, doi: 10.58258/jime.v8i1.2675.
- [3] M. Rizky and Y. Sugiarti, "Penggunaan Metode Scrum Dalam Pengembangan Perangkat Lunak: Literature Review," *J. Comput. Sci. Eng.*, vol. 3, no. 1, pp. 41–48, 2022, doi: 10.36596/jcse.v3i1.353.
- [4] I. P. Prabandaniwaransa, I. Ahmad, and E. R. Susanto, "Implementasi Metode Extreme Programming untuk Sistem Pengajuan Tempat PKL Berbasis Web," *J. Inform. dan Rekayasa Perangkat Lunak*, vol. 4, no. 2, pp. 221–227, 2023, doi: 10.33365/jatika.v4i2.2601.
- [5] D. . Budi, D. S., "Analisis Pemilihan Penerapan Proyek Metodologi Pengembangan Rekayasa Perangkat Lunak," *25th Int. Conf. Comput. Theory Appl. ICCTA 2015 - Proc.*, vol. 5, no. November, pp. 106–111, 2015, doi: 10.1109/ICCTA37466.2015.9513455.
- [6] D. T. Haniva, J. A. Ramadhan, and A. Suharso, "Systematic Literature Review Penggunaan Metodologi Pengembangan Sistem Informasi Waterfall, Agile, dan Hybrid," *J. Inf. Eng. Educ. Technol.*, vol. 7, no. 1, pp. 36–42, 2023, doi: 10.26740/jieet.v7n1.p36-42.
- [7] Subhiyanto, "Analisis Metode Pengembangan Sistem Informasi Manajemen : ASystematic Literatur Review," *Remik*, vol. 7, no. 1, pp. 821–834, 2023, doi: 10.33395/remik.v7i1.12177.
- [8] T. Pricillia and Zulfachmi, "Perbandingan Metode Pengembangan Perangkat Lunak (Waterfall, Prototype, RAD)," *J. Bangkit Indones.*, vol. 10, no. 1, pp. 6–12, 2021, doi: 10.52771/bangkitindonesia.v10i1.153.
- [9] D. Murdiani and M. Sobirin, "Perbandingan Metodologi Waterfall Dan RAD Dalam Pengembangan Sistem Informasi," *JINTEKS (Jurnal Inform. Teknol. dan Sains)*, vol. 4, no. 4, pp. 302–306, 2022, [Online]. Available:

- <http://www.jurnal.uts.ac.id/index.php/JINTEKS/article/view/2008>
- [10] Y. Amrozi, S. D. K. Wardani, and R. Ramadhan, “Quo Vadis Pengembangan Rekayasa Perangkat Lunak,” *J. Teknol. Terap. G-Tech*, vol. 3, no. 2, pp. 237–244, 2020, doi: 10.33379/gtech.v3i2.427.
- [11] D. M. Vira Adi Kurniyanti, “Konsep Siklus Hidup Pengembangan Sistem (SDLC) umumnya berkaitan dengan komputer atau sistem informasi. SDLC berfungsi sebagai kerangka kerja untuk mengembangkan sistem perangkat lunak, yang terdiri dari beberapa fase penting: perencanaan, analisis, desai,” vol. 9, no. 08, pp. 356–363, 2022.
- [12] Maikel Bolung and H. R. K. Tampangela, “ANALISA PENGGUNAAN METODOLOGI PENGEMBANGAN PERANGKAT LUNAK,” *Commun. Comput. Inf. Sci.*, vol. 1210 CCIS, no. 1, pp. 116–134, 2020, doi: 10.1007/978-981-15-7530-3_9.
- [13] A. A. Wahid, ““Analisis Metode Waterfall Untuk Pengembangan Sistem Informasi,” ,” *J. Ilmu-ilmu Inform. dan Manaj. STMIK*, vol. 1, no. October, 2020.
- [14] Y. Wahyudin and D. N. Rahayu, “Analisis Metode Pengembangan Sistem Informasi Berbasis Website: A Literatur Review,” *J. Interkom J. Publ. Ilm. Bid. Teknol. Inf. dan Komun.*, vol. 15, no. 3, pp. 26–40, 2020, doi: 10.35969/interkom.v15i3.74.
- [15] W. W. Widiyanto, “Analisa Metodologi Pengembangan Sistem Dengan Perbandingan Model Perangkat Lunak Sistem Informasi Kepegawaian Menggunakan Waterfall Development Model, Model Prototype, Dan Model Rapid Application Development (Rad),” *J. Inf. Politek. Indonusa Surakarta ISSN*, vol. 4, no. 1, pp. 34–40, 2018, [Online]. Available: <http://www.informa.poltekindonusa.ac.id/index.php/informa/article/view/34>
- [16] Y. N. Saputri, L. E. Sudiati, and N. Haryani, “SISTEM INFORMASI BUMDes DENGAN METODE RAD ATAU PROTOTYPE?(STUDI KOMPARASI ANTARA RAD & PROTOTYPE),” *Sosced*, vol. 6, no. 2, pp. 442–449, 2023.
- [17] F. R. Cahyadi and M. A. Sutisna, “Perbandingan Model Waterfall Dengan Rad Berbasis Website,” *J. Ilm. METADATA*, vol. 5, no. 1, pp. 19–33, 2023, doi: 10.47652/metadata.v5i1.308.
- [18] L. E. Zen and D. U. Iswavigra, “Critical Review: Analogi RAD, OOP dan EUD Method dalam Proses Development Sistem Informasi,” *J. Inf. dan Teknol.*, vol. 5, no. 1, pp. 184–190, 2023, doi: 10.37034/jidt.v5i1.286.
- [19] A. Z. D. Nur Adiya, D. L. Anggraeni, and Ilham Albana, “Analisa Perbandingan Penggunaan Metodologi Pengembangan Perangkat Lunak (Waterfall, Prototype, Iterative, Spiral, Rapid Application Development (RAD)),” *Merkurius J. Ris. Sist. Inf. dan Tek. Inform.*, vol. 2, no. 4, pp. 122–134, 2024, doi: 10.61132/mercurius.v2i4.148.
- [20] M. Z. Ramadhan and F. Angelia, “Mengoptimalkan pengembangan aplikasi mobile melalui perbandingan metode pengembangan perangkat lunak (Waterfall, Prototype, Mobile-D, Agile, RAD),” *J. Ilm. Teknol. Inf. dan Sains*, vol. 3, no. 2, pp. 13–19, 2023.