

# Analisis Komparatif dan Implikasi Strategis Model-Model Siklus Hidup Pengembangan Perangkat Lunak

Gufron<sup>1,\*</sup>

<sup>1</sup>Fakultas Keguruan dan Ilmu Pendidikan, Pendidikan Teknik Informatika & Komputer, Universitas Bung Hatta, Padang, Indonesia

Email: <sup>1,\*</sup>gufron@gufron.com

Email Penulis Korespondensi: <sup>1,\*</sup>gufron@gufron.com

**Abstrak**– Siklus Hidup Pengembangan Perangkat Lunak (SDLC) adalah kerangka kerja fundamental yang memandu proses rekayasa perangkat lunak dari inisiasi hingga pemeliharaan. Seiring dengan evolusi industri teknologi, berbagai model SDLC telah muncul, masing-masing dengan keunggulan dan kelemahan spesifik. Penelitian ini menyajikan tinjauan literatur sistematis untuk melakukan analisis komparatif mendalam terhadap empat model SDLC terkemuka: Waterfall, Agile, Spiral, dan V-Model. Metodologi yang digunakan melibatkan pengumpulan data dari literatur ilmiah dan industri untuk mengevaluasi model-model ini berdasarkan kriteria teknis dan manajerial, termasuk paradigma pengembangan, toleransi terhadap perubahan, manajemen risiko, dan estimasi biaya. Hasil penelitian ini menunjukkan bahwa tidak ada satu pun model SDLC yang secara universal “terbaik”. Model Waterfall dan V-Model sangat efektif untuk proyek-proyek dengan persyaratan yang stabil, di mana dokumentasi dan kontrol ketat menjadi prioritas. Sebaliknya, model Agile dan Spiral menawarkan adaptabilitas yang lebih besar dan sangat cocok untuk lingkungan proyek yang dinamis, kompleks, atau berisiko tinggi. Analisis komparatif yang disajikan dalam tabel-tabel terstruktur menyoroti hubungan kausal antara karakteristik model, seperti hubungan antara fleksibilitas yang tinggi dan kesulitan dalam estimasi biaya awal. Kesimpulan dari penelitian ini memberikan panduan berbasis bukti bagi para praktisi untuk memilih model SDLC yang paling sesuai, dan juga mengidentifikasi tren menuju pendekatan hibrida yang mengintegrasikan kekuatan dari berbagai model untuk mengoptimalkan hasil proyek.

Kata Kunci: SDLC; Waterfall; Agile; Spiral; V-Model

**Abstract**–The Software Development Life Cycle (SDLC) is a fundamental framework that guides the software engineering process from initiation to maintenance. Along with the evolution of the technology industry, various SDLC models have emerged, each with its specific strengths and weaknesses. This study presents a systematic literature review to conduct an in-depth comparative analysis of four prominent SDLC models: Waterfall, Agile, Spiral, and V-Model. The methodology involves collecting data from scientific and industry literature to evaluate these models based on technical and managerial criteria, including development paradigms, tolerance to change, risk management, and cost estimation. The findings indicate that no single SDLC model is universally the “best.” The Waterfall and V-Model are highly effective for projects with stable requirements, where documentation and strict control are prioritized. Conversely, Agile and Spiral models provide greater adaptability and are well-suited for dynamic, complex, or high-risk project environments. The comparative analysis, presented in structured tables, highlights causal relationships between model characteristics, such as the link between high flexibility and challenges in initial cost estimation. The conclusions of this study provide evidence-based guidance for practitioners in selecting the most appropriate SDLC model and identify emerging trends toward hybrid approaches that integrate the strengths of multiple models to optimize project outcomes.

Keywords: SDLC; Waterfall; Agile; Spiral; V-Model

## 1. PENDAHULUAN

Dalam ranah rekayasa perangkat lunak, *Software Development Life Cycle* (SDLC) memegang peranan krusial sebagai kerangka kerja yang mendefinisikan tahapan-tahapan yang terstruktur dan sistematis dalam pengembangan perangkat lunak [1]. Peran fundamental SDLC adalah untuk memastikan bahwa seluruh proses pengembangan, mulai dari konsepsi hingga pemeliharaan, berjalan secara terorganisir, efisien, dan menghasilkan produk dengan kualitas yang diinginkan [1]. Tanpa metodologi yang jelas, sebuah proyek perangkat lunak berisiko menghadapi inefisiensi, pembengkakan biaya, penundaan, atau bahkan kegagalan total [2].

Seiring dengan pesatnya kemajuan teknologi dan dinamika kebutuhan pasar, lanskap pengembangan perangkat lunak telah mengalami transformasi signifikan. Model-model SDLC tradisional yang bersifat linear kini ditantang oleh pendekatan yang lebih adaptif dan iteratif. Keragaman



metodologi yang tersedia, mulai dari model sekuensial yang ketat hingga model iteratif yang fleksibel, menimbulkan tantangan besar bagi manajer proyek dan pengembang dalam menentukan pilihan yang paling sesuai [2]. Kesalahan dalam pemilihan model dapat secara langsung memengaruhi keberhasilan proyek [3]. Oleh karena itu, pemahaman yang mendalam tentang karakteristik, keunggulan, dan kelemahan setiap model menjadi sangat penting untuk pengambilan keputusan strategis.

Berangkat dari latar belakang tersebut, penelitian ini berupaya menjawab pertanyaan-pertanyaan kunci yakni bagaimana perbandingan karakteristik, keunggulan, dan kelemahan dari model SDLC Waterfall, Agile, Spiral, dan V-Model dan faktor-faktor kunci apa yang paling signifikan dalam menentukan pemilihan model yang paling sesuai untuk suatu proyek?

Penelitian ini bertujuan untuk menganalisis secara komparatif model SDLC Waterfall, Agile, Spiral, dan V-Model berdasarkan kriteria teknis dan manajerial yang relevan; menyediakan panduan berbasis bukti untuk membantu para praktisi dan akademisi dalam pengambilan keputusan pemilihan model SDLC yang optimal; dan menyajikan temuan dalam format terstruktur dan mudah dipahami, yang dapat dijadikan referensi terpercaya.

Meskipun banyak studi telah melakukan tinjauan komparatif terhadap model-model SDLC, sebagian besar cenderung memberikan perbandingan yang bersifat umum atau hanya berfokus pada dua model saja. Keterbatasan ini menciptakan kebutuhan akan analisis yang lebih komprehensif yang akan melakukan perbandingan multi-kriteria secara terstruktur dan terpadu; menyajikan data perbandingan dalam bentuk tabel yang mudah dicerna, mengintegrasikan informasi dari berbagai sumber [4]; dan menghubungkan secara eksplisit karakteristik model dengan implikasi praktis dalam konteks proyek yang beragam.

Studi ini dirancang untuk mengisi kesenjangan tersebut dengan menyajikan analisis komparatif yang terperinci dan bernuansa. Dengan menyusun data penelitian yang ada ke dalam kerangka kerja yang sistematis, penelitian ini bertujuan untuk menawarkan wawasan yang lebih holistik dan dapat diterapkan secara praktis, yang melampaui deskripsi umum yang sering ditemukan dalam literatur yang ada.

## **2. METODOLOGI PENELITIAN**

### **2.1 Tahapan Penelitian**

Penelitian ini menggunakan pendekatan tinjauan literatur sistematis untuk mengumpulkan, mengevaluasi, dan mensintesis data dari berbagai sumber terpercaya. Sumber data mencakup jurnal ilmiah, laporan teknis, dan artikel industri yang relevan.

Prosedur analisis melibatkan beberapa tahapan: pertama, data dan informasi spesifik terkait karakteristik, keunggulan, dan kelemahan setiap model diekstrak dari sumber-sumber yang telah dikumpulkan. Kedua, informasi tersebut dikategorikan berdasarkan kriteria perbandingan yang telah ditetapkan. Ketiga, data kualitatif dan kuantitatif (jika tersedia) dianalisis untuk mengidentifikasi pola, tren, dan hubungan kausal di antara model-model yang berbeda. Proses ini memastikan bahwa hasil yang diperoleh tidak hanya deskriptif, tetapi juga memberikan pemahaman yang lebih dalam tentang implikasi praktis dari setiap model.

### **2.2 Model yang Diperbandingkan**

Penelitian ini memfokuskan analisis pada empat model SDLC yang paling representatif dan banyak digunakan dalam industri, yakni:

1. Model Waterfall: Dipilih sebagai fondasi dan model tertua dalam SDLC [5]. Model ini merepresentasikan paradigma sekuensial-linear yang menjadi dasar bagi banyak model selanjutnya.
2. Model Agile: Mewakili revolusi paradigma dari pendekatan linear menuju pendekatan iteratif dan responsif [6]. Model ini sangat relevan dengan kebutuhan pasar modern yang dinamis.

3. Model Spiral: Dipilih karena mewakili pendekatan berbasis risiko yang menggabungkan elemen sekuensial dan iteratif [7]. Hal ini menjadikannya model yang ideal untuk proyek-proyek yang kompleks dan memiliki tingkat ketidakpastian yang tinggi.
4. V-Model: Dianggap sebagai "ekstensi" atau varian dari Waterfall. Pemilihan model ini penting karena secara eksplisit menyoroti pentingnya verifikasi dan validasi, yang merupakan respons langsung terhadap salah satu kelemahan terbesar model Waterfall [8].

### 2.3 Kerangka Perbandingan dan Kriteria Analisis

Untuk memastikan analisis yang komprehensif, perbandingan dilakukan berdasarkan kriteria yang relevan, yang dikompilasi dari berbagai literatur. Kriteria-kriteria utama yang digunakan meliputi:

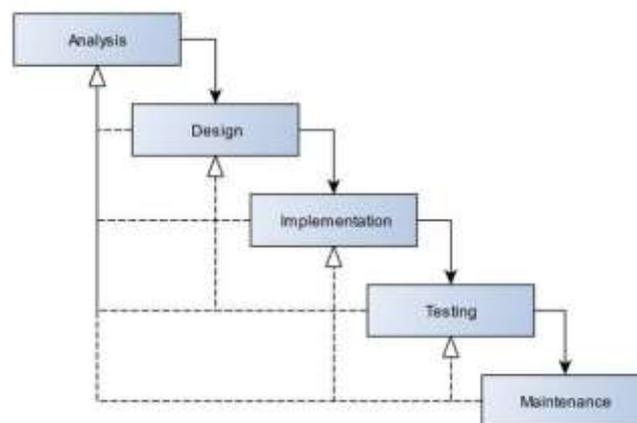
- Paradigma Pengembangan: Linear vs. Iteratif vs. Berbasis Risiko.
- Stabilitas Kebutuhan (*Requirements*): Apakah model ini cocok untuk kebutuhan yang tetap atau sering berubah?
- Fleksibilitas terhadap Perubahan: Seberapa mudah model mengakomodasi perubahan?
- Tingkat Risiko Proyek: Apakah model ini dirancang untuk mengelola risiko tinggi?
- Keterlibatan Pelanggan: Kapan dan seberapa sering pelanggan berinteraksi?
- Deteksi Cacat (*Error Detection*): Kapan cacat biasanya ditemukan dalam siklus hidup?
- Estimasi Biaya dan Jadwal: Seberapa mudah untuk memperkirakan biaya dan waktu?
- Kompleksitas Proyek: Apakah cocok untuk proyek kecil, sedang, atau besar/kompleks?
- Kualitas dan Dokumentasi: Sejauh mana model menekankan kualitas dan dokumentasi?

## 3. HASIL DAN PEMBAHASAN

### 3.1 Karakteristik Dasar Model-Model SDLC

#### 3.1.1 Model Waterfall

Model Waterfall adalah model SDLC tertua dan paling dikenal, yang mengilustrasikan proses pengembangan perangkat lunak dalam aliran sekuensial-linear yang mengalir ke bawah seperti air terjun. Setiap fase, yaitu analisis, desain, implementasi, pengujian, dan pemeliharaan, harus diselesaikan sepenuhnya sebelum fase berikutnya dimulai [5]. Hasil dari setiap fase menjadi input untuk fase berikutnya, memastikan pendekatan yang sangat terstruktur.



Gambar 1. Waterfall Model

Model ini menawarkan beberapa keunggulan, seperti kesederhanaan dan kemudahan untuk dipahami serta dikelola. Dengan alur yang kaku, setiap fase memiliki deliverables dan proses tinjauan yang jelas, memungkinkan dokumentasi yang sangat terperinci dan menyeluruh. Estimasi biaya total dan jadwal dapat dilakukan dengan tingkat akurasi yang relatif tinggi di awal proyek. Model ini sangat ideal untuk proyek skala kecil dengan persyaratan yang sangat jelas, stabil, dan tidak ambigu.

Meskipun demikian, Waterfall memiliki kelemahan signifikan. Sifatnya yang kaku membuatnya sangat tidak fleksibel terhadap perubahan. Jika modifikasi diperlukan setelah suatu fase selesai, proyek

harus kembali ke tahap awal, yang dapat menimbulkan biaya substansial [9]. Selain itu, deteksi cacat cenderung terjadi sangat terlambat dalam siklus hidup proyek, biasanya pada fase pengujian atau bahkan setelah produk dikirim. Keterlambatan ini dapat menyebabkan biaya perbaikan yang sangat tinggi.

### 3.1.2 Model Agile

Model Agile adalah pendekatan yang sangat fleksibel dan iteratif, berfokus pada pengiriman perangkat lunak dalam siklus pengembangan pendek yang dikenal sebagai sprint, biasanya berlangsung selama 2-4 minggu. Berbeda dengan model tradisional, Agile menekankan pada kolaborasi, adaptabilitas, dan umpan balik pelanggan yang berkelanjutan [6]. Inti dari filosofi Agile terletak pada empat nilai utama yang tertuang dalam Manifesto Agile: mengutamakan individu dan interaksi, perangkat lunak yang berfungsi, kolaborasi dengan pelanggan, dan respons terhadap perubahan.



**Gambar 2.** *Agile Model*

Model ini sangat responsif terhadap perubahan kebutuhan, memungkinkan tim untuk menyesuaikan diri dengan cepat terhadap dinamika pasar dan umpan balik pengguna. Keterlibatan pelanggan yang tinggi dan berkelanjutan pada setiap iterasi memastikan bahwa produk yang dikembangkan benar-benar sesuai dengan ekspektasi. Pendekatan ini juga memungkinkan waktu pemasaran (*time-to-market*) yang lebih cepat karena fungsionalitas yang berfungsi dapat dikirimkan secara bertahap.

Di sisi lain, Agile memiliki tantangan. Model ini kurang cocok untuk proyek yang sangat besar dan kompleks kecuali tim yang terlibat sangat terampil dan berpengalaman. Dokumentasi yang minimal dapat menjadi masalah dalam pemeliharaan jangka panjang atau saat anggota tim baru bergabung. Selain itu, estimasi biaya dan jadwal di awal proyek cenderung sulit dan tidak akurat karena sifatnya yang beradaptasi dengan perubahan yang konstan [9].

### 3.1.3 Model Spiral

Model Spiral adalah pendekatan berbasis risiko yang menggabungkan elemen sekuensial dari Waterfall dengan pengembangan iteratif. Setiap "spiral" atau iterasi dari model ini berfokus pada empat aktivitas utama: perencanaan, analisis risiko, rekayasa (pengembangan), dan evaluasi. Proses ini berulang, dengan setiap iterasi membangun dan memperbaiki prototipe perangkat lunak.



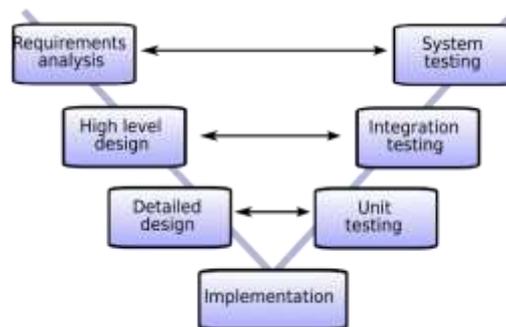
**Gambar 3.** *Spiral Model*

Keunggulan utama Model Spiral terletak pada kemampuannya yang sangat kuat dalam manajemen risiko. Model ini mengidentifikasi, menganalisis, dan memitigasi risiko sejak dini, sebelum risiko tersebut meningkat dan menyebabkan kegagalan proyek [7]. Sifatnya yang fleksibel memungkinkan adaptasi terhadap perubahan kebutuhan yang signifikan, menjadikannya pilihan yang ideal untuk proyek yang sangat kompleks, skala besar, dan berisiko tinggi. Keterlibatan pelanggan secara teratur pada setiap siklus evaluasi juga memastikan bahwa produk selaras dengan harapan.

Namun, Spiral juga memiliki kekurangan. Model ini sangat mahal dan kompleks untuk dikelola, bahkan lebih dari model lainnya. Model ini membutuhkan keahlian khusus yang tinggi dalam analisis risiko untuk memastikan keberhasilannya. Selain itu, untuk proyek skala kecil atau berisiko rendah, waktu dan biaya yang dihabiskan untuk analisis risiko mungkin tidak efisien.

### 3.1.4 Model V-Model

V-Model, atau Verifikasi dan Validasi Model, dapat dianggap sebagai ekstensi dari model Waterfall yang menempatkan penekanan signifikan pada pengujian. Model ini memvisualisasikan proses pengembangan dan pengujian yang berjalan secara paralel dan berpasangan, membentuk bentuk "V". Setiap fase pengembangan di sisi kiri (analisis, desain, implementasi) memiliki fase pengujian yang sesuai di sisi kanan (pengujian unit, integrasi, sistem, dan penerimaan pengguna).



Gambar 4. V-Model Model

Keunggulan utama V-Model adalah kemampuannya untuk mendeteksi cacat sejak dini karena perencanaan dan pelaksanaan pengujian dimulai sejak fase awal pengembangan. Hal ini secara langsung mengatasi kelemahan terbesar Waterfall (deteksi cacat yang terlambat) [10]. Penekanan yang kuat pada verifikasi dan validasi pada setiap tahap memastikan kualitas dan keandalan produk yang tinggi. Selain itu, model ini menyediakan ketertelusuran (*traceability*) yang jelas antara persyaratan dan hasil pengujian, yang sangat berharga untuk kepatuhan (*compliance*) dan audit.

Namun, V-Model memiliki kelemahan yang sama dengan Waterfall, yaitu sangat kaku dan tidak fleksibel terhadap perubahan setelah suatu fase selesai. Hal ini membuatnya tidak cocok untuk proyek-proyek dengan persyaratan yang berpotensi berubah dengan cepat. Model ini juga dapat menjadi lebih mahal dan membutuhkan lebih banyak sumber daya dibandingkan Waterfall karena sifatnya yang paralel. Meskipun kaku, V-Model adalah respons langsung terhadap masalah fundamental Waterfall, menjadikannya solusi pragmatis untuk industri yang diatur secara ketat, di mana kualitas dan kepatuhan lebih diutamakan daripada kecepatan.

## 3.2 Analisis Komparatif Berbasis Kriteria

Perbedaan antara model-model SDLC ini tidak hanya terletak pada tahapan-tahapannya, tetapi juga pada filosofi dan paradigma dasarnya. Model Waterfall berakar pada filosofi prediktabilitas dan kontrol melalui perencanaan yang mendalam di awal. Ini secara alami menyebabkan model ini menuntut persyaratan yang stabil dan dokumentasi yang ketat, yang pada gilirannya menghasilkan ketidakefleksibelan. Sebaliknya, Model Agile mengedepankan adaptabilitas dan nilai pelanggan melalui iterasi berkelanjutan, yang memungkinkannya merespons perubahan dengan cepat namun dengan konsekuensi dokumentasi yang tidak sekomprehensif model linear. Model Spiral berpusat pada mitigasi risiko, dan V-Model berfokus pada verifikasi dan kualitas.

Hubungan sebab-akibat yang mendasari filosofi ini terungkap dengan jelas ketika kriteria teknis dianalisis secara mendalam. Model-model yang dirancang untuk mengatasi proyek berisiko tinggi dan kompleks (Spiral dan Agile) memiliki biaya yang lebih tinggi dan estimasi biaya yang lebih sulit dibandingkan model-model linear (Waterfall). Hal ini bukanlah kelemahan acak, melainkan harga yang harus dibayar untuk fleksibilitas dan kemampuan mitigasi risiko. Proses berulang dan adaptif membutuhkan sumber daya ahli yang lebih banyak dan manajemen yang lebih kompleks, yang secara inheren meningkatkan biaya.

Tabel-tabel berikut menyajikan ringkasan analisis komparatif berdasarkan kriteria yang telah ditetapkan.

**Tabel 1.** Perbandingan Karakteristik Utama dan Kategori Proyek Model SDLC

Fitur	Waterfall	Agile	Spiral	V-Model
Paradigma	Sekuensial-linear	Iteratif dan Inkremental	Berbasis risiko	Sekuensial dengan penekanan pengujian
Fokus Utama	Perencanaan dan Dokumentasi	Kolaborasi, Responsif, dan Adaptasi	Manajemen risiko	Verifikasi dan Kualitas
Toleransi Perubahan	Rendah (Kaku)	Sangat tinggi	Tinggi	Rendah (Kaku)
Keterlibatan Klien	Rendah (terutama di awal)	Tinggi (berkelanjutan)	Rendah (setelah setiap iterasi)	Rendah (terutama di awal)
Kategori Proyek Ideal	Kebutuhan stabil, kecil-menengah	Kebutuhan dinamis, waktu-ke-pasar cepat	Kompleks, berisiko tinggi	Kebutuhan stabil, kualitas & kepatuhan krusial

Tabel di atas menunjukkan bahwa setiap model SDLC memiliki orientasi dan kategori proyek yang berbeda sesuai karakteristik utamanya. Waterfall dan V-Model sama-sama menekankan pendekatan sekuensial dengan toleransi perubahan yang rendah serta keterlibatan klien yang minim, sehingga lebih tepat untuk proyek dengan kebutuhan stabil, ukuran relatif kecil hingga menengah, atau yang menuntut kualitas serta kepatuhan regulasi tinggi. Sebaliknya, Agile hadir dengan paradigma iteratif dan inkremental, mengedepankan kolaborasi serta adaptasi cepat terhadap perubahan, sehingga sangat sesuai untuk proyek dengan kebutuhan dinamis dan target waktu-ke-pasar yang ketat. Sementara itu, Spiral menonjol karena berbasis pada manajemen risiko, dengan toleransi perubahan yang tinggi dan penerapan iteratif, menjadikannya pilihan ideal untuk proyek besar, kompleks, serta berisiko tinggi. Dengan demikian, pemilihan model yang tepat bergantung pada stabilitas kebutuhan, tingkat kompleksitas, serta tingkat keterlibatan klien yang diharapkan.

**Tabel 2.** Analisis Kriteria Teknis Model SDLC

Kriteria	Waterfall	Spiral	Agile (Iteratif)	V-Model
Stabilitas Kebutuhan	Mutlak	Tidak semua dan sering berubah	Tidak semua dan sering berubah	Mutlak
Proyek Jangka Panjang	Tidak tepat	Tepat	Tepat	Tepat
Proyek Kompleks	Tidak tepat	Tepat	Tepat	Tepat
Estimasi Biaya	Mudah	Sulit	Sulit	Mudah
Fleksibilitas	Tidak fleksibel	Fleksibel	Sangat fleksibel	Kaku
Simplicity	Sederhana	Menengah	Menengah	Sederhana
Dukungan Proyek Berisiko Tinggi	Tidak tepat	Tepat	Tepat	Tidak tepat
Jaminan Keberhasilan	Rendah	Tinggi	Tinggi	Tinggi

Keterlibatan Pelanggan	Rendah	Rendah (setelah setiap iterasi)	Tinggi (setelah setiap iterasi)	Rendah (di awal)
Waktu Pengujian	Terlambat	Pada akhir setiap fase	Setelah setiap iterasi	Dimulai sejak fase awal
Pemeliharaan Kemudahan Implementasi	Paling sulit Mudah	Dapat dikelola Kompleks	Dapat dikelola Mudah	Dapat dikelola Mudah

### 3.3 Hasil Analisis

Analisis komparatif terhadap berbagai model SDLC memperlihatkan bahwa setiap pendekatan memiliki keunggulan dan keterbatasan yang melekat pada paradigma dan orientasi utamanya. Model Waterfall dan V-Model menekankan alur kerja yang sekuensial dengan kebutuhan yang harus stabil sejak awal, sehingga keduanya lebih sesuai digunakan pada proyek yang bersifat terstruktur, dengan ruang lingkup jelas, serta menuntut kepatuhan regulasi dan kualitas tinggi. Keduanya memberikan kemudahan dalam estimasi biaya dan kesederhanaan implementasi, namun kelemahannya terletak pada fleksibilitas yang rendah dan keterlibatan klien yang terbatas. Akibatnya, jika terjadi perubahan kebutuhan di tengah proyek, maka adaptasi menjadi sulit dilakukan.

Berbeda halnya dengan Agile, yang menawarkan pendekatan iteratif dan inkremental dengan tingkat fleksibilitas tertinggi. Agile sangat efektif dalam menangani kebutuhan yang dinamis dan memberikan ruang adaptasi yang cepat terhadap perubahan. Kolaborasi intensif dengan klien menjadi faktor kunci yang memungkinkan produk berkembang sesuai ekspektasi pasar dalam waktu yang lebih singkat. Dari sisi teknis, Agile memang menyulitkan dalam hal estimasi biaya dan memerlukan keterlibatan pelanggan secara berkelanjutan, namun kompensasinya adalah jaminan keberhasilan proyek yang relatif tinggi serta proses pemeliharaan yang lebih mudah karena pengujian dan evaluasi dilakukan terus-menerus.

Sementara itu, Spiral menempati posisi unik dengan orientasi pada manajemen risiko. Model ini memadukan sifat iteratif dengan pendekatan berbasis evaluasi risiko di setiap siklusnya. Hasil analisis menunjukkan bahwa Spiral lebih tepat digunakan pada proyek jangka panjang dan kompleks dengan tingkat ketidakpastian serta risiko kegagalan yang tinggi. Walaupun estimasi biaya pada Spiral sulit dilakukan dan implementasinya cenderung lebih kompleks, namun fleksibilitas dan kemampuannya dalam mengelola risiko menjadikan Spiral salah satu model dengan tingkat keberhasilan paling tinggi pada kategori proyek yang kritis.

Secara keseluruhan, tabel komparatif mengindikasikan bahwa tidak ada satu model pun yang bersifat universal untuk semua jenis proyek. Waterfall dan V-Model unggul dalam stabilitas, kejelasan dokumentasi, dan proyek yang berorientasi pada kualitas serta kepatuhan. Agile lebih adaptif pada konteks kebutuhan yang cepat berubah dan menuntut keterlibatan aktif dari pemangku kepentingan. Spiral menjadi pilihan strategis untuk proyek besar dan berisiko tinggi yang membutuhkan pendekatan pengelolaan risiko secara sistematis. Dengan demikian, pemilihan model SDLC sebaiknya disesuaikan dengan karakteristik proyek, tingkat kompleksitas, kebutuhan klien, serta toleransi terhadap risiko dan perubahan.

## 4. KESIMPULAN

Berdasarkan analisis komparatif yang mendalam, dapat disimpulkan bahwa tidak ada satu model SDLC pun yang secara universal superior. Setiap model memiliki karakteristik unik yang membuatnya optimal untuk konteks proyek tertentu. Tinjauan ini mengonfirmasi bahwa model Waterfall dan V-Model sangat cocok untuk proyek-proyek dengan persyaratan yang sudah jelas, di mana struktur, dokumentasi, dan kualitas adalah prioritas utama. Sebaliknya, model Agile dan Spiral menawarkan keunggulan tak tertandingi dalam proyek-proyek yang dinamis, kompleks, dan berisiko tinggi, di mana adaptabilitas dan kemampuan mitigasi risiko adalah kunci keberhasilan. Pemilihan model SDLC yang tepat merupakan keputusan strategis yang secara langsung memengaruhi hasil proyek.

Implikasi praktis dari penelitian ini mengarah pada rekomendasi untuk menggunakan Waterfall atau V-Model jika persyaratan proyek sangat jelas, terperinci, dan tidak akan berubah. Proyek berskala kecil dengan durasi singkat juga sangat cocok. Model ini ideal untuk industri yang sangat diatur di mana kepatuhan dan keandalan menjadi prioritas, seperti industri medis, aviasi, atau pertahanan. Gunakan Agile jika persyaratan proyek tidak pasti dan cenderung berubah. Proyek yang menuntut waktu pemasaran yang cepat dan kolaborasi pelanggan yang aktif akan mendapatkan manfaat maksimal dari model ini. Dan gunakan Spiral jika proyek memiliki tingkat risiko yang sangat tinggi, kompleksitas besar, dan persyaratan yang tidak jelas di awal. Model ini adalah pilihan terbaik ketika investasi awal yang besar diperlukan untuk memitigasi risiko potensial.

Pengamatan lebih lanjut menunjukkan bahwa dalam praktik, semakin banyak organisasi yang mengadopsi model hibrida yang menggabungkan elemen terbaik dari berbagai model SDLC. Sebagai contoh, menggabungkan dokumentasi terstruktur dari Waterfall dengan pengembangan iteratif dari Agile dapat menawarkan keseimbangan optimal antara kontrol dan fleksibilitas untuk proyek-proyek tertentu. Evolusi ini mencerminkan pengakuan bahwa pendekatan yang kaku seringkali tidak memadai untuk mengatasi kompleksitas proyek modern.

## REFERENCES

- [1] Khisty. S, *Waterfall, Agile, Spiral: Which SDLC Model is Right?*, devron, 5 Mei 2025. [Online]. Tersedia: <https://devtron.ai/blog/waterfall-vs-agile-vs-spiral-sdlc-methodologies-compared/> [Diakses 26 Juli 2025].
- [2] Gupta. A, Rawal. A, Barge. Y, *Comparative Study of Different SDLC Models*, IJRASET, Volume 9 Issue XI, pp.73-80, Nov 2021. DOI: 10.22214/ijraset.2021.38736.
- [3] Farah Fadhillah. F, et. al, *Analisis Faktor Kesuksesan Proyek Perangkat Lunak*, JIITE, Vol. 1 No. 3, pp.148.156, Desember 2024. DOI: <https://doi.org/10.63547/jiite.v1i3.40>.
- [4] M.A. Adeagbo, *Project Implementation Decision Using Software Development Life Cycle Models: A Comparative Approach*, AJOL, Vol. 28 No. 1, September 2021. DOI: <http://dx.doi.org/10.4314/jcsia.v28i1.10>.
- [5] Gufron, *Pengembangan Aplikasi Pembukuan Pada UMKM MN Djamboe Berbasis Bootstrap Dengan Metode Waterfall*, Jurnal Media Informatika, Vol. 5 No. 3, Agustus 2024, pp.240-247. [Online] Tersedia: <https://ejournal.sisfokomtek.org/index.php/jumin/article/view/3930>
- [6] Six Sigma, *The Guide to Agile SDLC: A Modern Approach to Software Development*, Six Sigma, 7 Januari 2025. [Online]. Tersedia: <https://www.6sigma.us/six-sigma-in-focus/agile-sdlc-software-development-life-cycle/> [Diakses: 26 Juli 2025].
- [7] Talreja. Abhay, *Spiral Model in Software Development: Guide to Risk-Driven Development*, Teaching Agile, 14 Agustus 2025. [Online]. Tersedia: <https://teachingagile.com/sdlc/models/spiral> [Diakses 27 Juli 2025].
- [8] Kazankova. N, *Guide: V-model & testing embedded software*, Code Intelligence Blog. [Online] Tersedia: <https://www.code-intelligence.com/blog/everything-about-v-model-and-testing-embedded-software> [Diakses 27 Juli 2025].
- [9] Alshamrani. A, Bahattab. A, *A Comparison Between Three SDLC Models Waterfall Model, Spiral Model, and Incremental/Iterative Model*, IJCSI, Vol. 12 Issue 1, Januari 2015, pp.106-111.
- [10] Balaji. S, Murugaiyan. M.S, *Waterfall Vs V-Model Vs Agile: A Comparative Study on SDLC*, IJITBM, Vol. 2 No. 1, Juni 2012, pp.26-30.