Journal of Innovative and Creativity
Journal Homepage: https://joecy.org/index.php/joecy
e-mail: journaljoecy@gmail.com

Faculty of Education
University of Pahlawan Tuanku Tambusai
ISSN 2775-771X (Online)
**ISSN** 2962-570X **(Printed)**
5 (2) 2025, Page: 13665-13671

# Perbandingan Model Pengembangan Perangkat Lunak: Waterfall vs Agile

**Siti Rawiyah[1], Maria Rosmelinda Rona Pani[2], Nabila Syabani[3], Nadilah Ulul Azmi[4]**
**[1,2,3,4]Program Studi Teknik Informatika, Universitas Ibnu Sina, Lubuk Baja Kota, Kec.Lubuk Baja, Kota Batam, Kepulauan Riau 29444 Indonesia**
Corresponding Author: 231055201012@uis.ac.id[1], 231055201094@uis.ac.id[2], 231055201076@uis.ac.id[3], 231055201048@uis.ac.id[4]

Published: July , 2025

## ABSTRACT
Pengembangan perangkat lunak merupakan proses yang kompleks dan memerlukan pendekatan metodologis yang tepat agar produk akhir dapat memenuhi kebutuhan pengguna secara optimal. Di antara berbagai model pengembangan yang ada, Waterfall dan Agile adalah dua pendekatan yang paling sering digunakan dan mewakili dua filosofi yang sangat berbeda. Waterfall, sebagai model klasik, mengusung pendekatan linier dan terstruktur, di mana setiap fase harus diselesaikan sebelum fase berikutnya dimulai. Sementara itu, Agile menawarkan pendekatan iteratif dan fleksibel, dengan menekankan kolaborasi tim, keterlibatan aktif pengguna, serta kemampuan untuk merespons perubahan dengan cepat selama proses pengembangan berlangsung. Artikel ini bertujuan untuk membandingkan kedua model tersebut dari berbagai aspek penting, seperti fleksibilitas, efisiensi waktu, pengelolaan risiko, skala proyek, serta tingkat keterlibatan pengguna. Dengan menggunakan metode studi literatur dari berbagai sumber akademik dan praktisi industri, analisis dilakukan untuk memberikan gambaran mendalam mengenai kelebihan, kekurangan, dan konteks penerapan masing-masing model. Hasil kajian menunjukkan bahwa tidak ada satu model yang secara mutlak lebih baik dari yang lain, melainkan pemilihan model yang tepat sangat bergantung pada karakteristik proyek yang dijalankan. Waterfall cenderung lebih cocok untuk proyek dengan kebutuhan yang jelas dan stabil, sementara Agile lebih sesuai untuk proyek yang dinamis dan berkembang. Temuan dalam artikel ini diharapkan dapat membantu pengembang, manajer proyek, maupun akademisi dalam menentukan model pengembangan yang paling relevan dan efektif untuk diterapkan.(Thesing, Feldmann, and Burchardt 2021)

**Kata kunci**: Pengembangan Perangkat Lunak, Waterfall, Agile, Metodologi, Perbandingan, Manajemen Proyek

## ABSTRACT
Software development is a complex process and requires an appropriate methodological approach to ensure the final product optimally meets user needs. Among the various development models available, Waterfall and Agile are the two most frequently used approaches, representing two very different philosophies. Waterfall, as the classic model, promotes a linear and structured approach, where each phase must be completed before the next phase can begin. Meanwhile, Agile offers an iterative and flexible approach, emphasizing team collaboration, active user involvement, and the ability to respond quickly to changes throughout the development process. This article aims to compare the two models in various key aspects, such as flexibility, time efficiency, risk management, project scale, and level of user involvement. Using literature review methods from various academic sources and industry practitioners, the analysis was conducted to provide an in-depth overview of the advantages, disadvantages, and application contexts of each model. The results of the study indicate that no one model is absolutely better than the other; rather, the selection of the right model depends heavily on the characteristics of the project being undertaken. Waterfall tends to be more suitable for projects with clear and stable needs, while Agile is more appropriate for dynamic and evolving projects. The findings in this article are expected to help developers, project managers, and academics in determining the most relevant and effective development model to implement. (Thesing, Feldmann, and Burchardt 2021)

**Keywords:** Software Development, Waterfall, Agile, Methodology, Comparison, Project Management

© The Author(s). 2021 Open Access This article is licensed under a Creative Commons Atibution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. To view a copy of this licence, visit http://creativecommons.org/licenses/by/4.0

## PENDAHULUAN

Perangkat lunak telah menjadi bagian integral dari berbagai sektor kehidupan, mulai dari pendidikan, kesehatan, pemerintahan, industri, hingga sektor hiburan. Seiring meningkatnya kompleksitas kebutuhan pengguna dan perkembangan teknologi yang sangat pesat, pengembangan perangkat lunak menuntut pendekatan yang sistematis, efisien, dan adaptif. Dalam konteks tersebut, metodologi atau model pengembangan perangkat lunak (software development lifecycle models) berfungsi sebagai panduan utama dalam mengatur tahapan-tahapan kerja, pembagian peran tim, pengelolaan waktu, serta pencapaian target fungsionalitas perangkat lunak secara keseluruhan.(Hidayah Nova et al. 2022)

Dari sekian banyak model pengembangan yang telah diperkenalkan dan diterapkan, dua di antaranya yang paling dikenal luas adalah model Waterfall dan model Agile. Keduanya menawarkan kerangka kerja yang berbeda dalam merancang, membangun, dan menguji perangkat lunak. Model Waterfall merupakan pendekatan konvensional yang mengikuti alur linier dan berurutan, di mana tahapan pengembangan dilakukan satu per satu secara runtut, mulai dari analisis kebutuhan hingga pemeliharaan. Model ini menekankan dokumentasi lengkap dan perencanaan yang matang di awal proyek. Sebaliknya, model Agile merupakan pendekatan modern yang menekankan pada fleksibilitas, kolaborasi tim, dan iterasi cepat, dengan memungkinkan perubahan kebutuhan selama siklus pengembangan berlangsung.(Leong et al. 2023)

Perbedaan mendasar antara kedua model ini mencerminkan dua paradigma yang berbeda dalam manajemen proyek perangkat lunak: perencanaan terstruktur versus adaptasi berkelanjutan. Masing-masing model memiliki kelebihan dan kekurangan yang perlu dipertimbangkan secara cermat, tergantung pada karakteristik proyek, ukuran tim, ketersediaan waktu, anggaran, serta tingkat kepastian terhadap kebutuhan pengguna. Dalam proyek berskala besar dan memiliki spesifikasi kebutuhan yang stabil, pendekatan Waterfall mungkin lebih cocok karena dapat meminimalisasi ambiguitas. Sebaliknya, untuk proyek yang dinamis, seperti pengembangan aplikasi startup atau sistem berbasis cloud yang terus berkembang, pendekatan Agile memberikan ruang lebih besar untuk eksplorasi dan perubahan.

Artikel ini bertujuan untuk memberikan analisis komprehensif terhadap perbandingan antara model Waterfall dan Agile dalam konteks pengembangan perangkat lunak. Melalui kajian literatur dari berbagai sumber akademik dan praktis, pembahasan akan difokuskan pada aspek proses kerja, fleksibilitas, efisiensi, manajemen risiko, serta tingkat kepuasan pengguna. Dengan memahami keunggulan dan keterbatasan masing-masing model, diharapkan pembaca—baik dari kalangan akademisi, mahasiswa, maupun praktisi industri—dapat memperoleh wawasan yang lebih baik dalam memilih model pengembangan yang paling sesuai dengan kebutuhan proyek mereka.

## METODE PENELITIAN

Penelitian ini menggunakan pendekatan kualitatif deskriptif dengan metode studi literatur (literature review). Tujuan utama dari metode ini adalah untuk menghimpun, menganalisis, dan menyintesis berbagai sumber referensi ilmiah yang relevan mengenai model pengembangan perangkat lunak Waterfall dan Agile, guna memperoleh pemahaman yang mendalam tentang karakteristik, keunggulan, serta keterbatasan masing-masing model.(Nurmalasari et al. 2023)

Data dikumpulkan melalui penelusuran sumber-sumber sekunder yang bersifat ilmiah dan kredibel, antara lain:

1. Jurnal-jurnal internasional dan nasional yang terindeks (seperti IEEE, Elsevier, Springer, dan Google Scholar).
2. Buku-buku ajar mengenai rekayasa perangkat lunak (misalnya karya Pressman, Sommerville, Highsmith, dsb).
3. Artikel whitepaper dan dokumentasi resmi metodologi Agile dan Waterfall.
4. Laporan proyek dan studi kasus implementasi model Waterfall dan Agile di dunia industri.

Kriteria inklusi dalam pemilihan sumber adalah publikasi yang membahas secara langsung model pengembangan perangkat lunak dan diterbitkan dalam 10 tahun terakhir, sedangkan kriteria eksklusi adalah sumber yang bersifat opini pribadi atau tidak memiliki dasar metodologis yang kuat.(Surahman, Deni Wahyudi, and Sintaro 2020)

Analisis data dilakukan dengan cara membandingkan kedua model berdasarkan lima indikator utama, yaitu:

1. Fleksibilitas terhadap perubahan kebutuhan proyek
2. Kecepatan dan efisiensi waktu pengembangan
3. Keterlibatan pengguna dalam proses pengembangan
4. Manajemen risiko dan penanganan kesalahan
5. Kesesuaian terhadap skala dan jenis proyek

Hasil dari kajian literatur kemudian dikategorikan dan disusun dalam bentuk tabel perbandingan, narasi analisis, serta ringkasan temuan. Validitas data dijaga dengan melakukan triangulasi sumber, yaitu membandingkan hasil dari berbagai referensi yang memiliki pandangan berbeda untuk memastikan objektivitas hasil kajian.(Samsiyah, Musadad, and Pelu 2020)

## HASIL DAN PEMBAHASAN

Berdasarkan hasil studi literatur dan analisis terhadap berbagai sumber, diperoleh perbandingan antara model Waterfall dan Agile berdasarkan lima indikator utama, yaitu fleksibilitas, efisiensi waktu, keterlibatan pengguna, manajemen risiko, dan kesesuaian

terhadap skala proyek. Setiap indikator dianalisis secara mendalam untuk memahami kelebihan dan kekurangan masing-masing model.(Diansyah et al. 2023)

### Fleksibilitas

Model Waterfall bersifat kaku dan kurang fleksibel dalam menangani perubahan kebutuhan. Dalam model ini, seluruh tahapan proyek harus diselesaikan secara berurutan, sehingga perubahan yang terjadi setelah tahap analisis atau desain sulit untuk diakomodasi tanpa mengulang proses dari awal. Hal ini dapat menimbulkan keterlambatan dan biaya tambahan apabila terdapat perubahan signifikan dari klien.

Sebaliknya, model Agile didesain untuk menghadapi perubahan dengan lebih baik. Agile bersifat iteratif dan adaptif, memungkinkan tim pengembang untuk merespons perubahan kebutuhan pengguna secara cepat dalam setiap sprint atau iterasi. Dengan pendekatan ini, produk yang dikembangkan dapat terus disesuaikan seiring dengan berkembangnya kebutuhan dan kondisi pasar.

### Efesiensi Waktu

Waterfall membutuhkan waktu yang lebih lama karena setiap tahap harus diselesaikan sepenuhnya sebelum melanjutkan ke tahap berikutnya. Hal ini menjadikan siklus pengembangan panjang, terutama ketika terjadi revisi besar. Meskipun cocok untuk proyek yang perencanaannya sangat rinci sejak awal, model ini cenderung kurang efisien dalam proyek dengan tenggat waktu yang ketat.

Agile justru menekankan pada pengembangan cepat melalui iterasi singkat yang biasanya berlangsung 1–4 minggu. Fitur-fitur kecil yang selesai lebih cepat dapat langsung diuji dan digunakan, sehingga pengguna mendapatkan nilai fungsional sejak dini. Hal ini membuat Agile lebih efisien dalam pengiriman produk awal dan memungkinkan perbaikan berkelanjutan secara real-time.

### Keterlibatan Pengguna

Dalam model Waterfall, keterlibatan pengguna umumnya hanya dilakukan pada tahap awal (analisis kebutuhan) dan tahap akhir (pengujian akhir atau implementasi). Hal ini mengurangi kemungkinan adanya umpan balik berkelanjutan selama proses pengembangan, yang dapat mengakibatkan ketidaksesuaian antara produk akhir dengan harapan pengguna.

Agile, sebaliknya, menjadikan keterlibatan pengguna sebagai bagian penting dari setiap iterasi. Pengguna atau perwakilannya diundang untuk memberikan umpan balik secara rutin, sehingga pengembang dapat menyesuaikan dan memperbaiki produk secara terus menerus. Pendekatan ini terbukti meningkatkan tingkat kepuasan pengguna dan relevansi fitur yang dikembangkan.

### Manajemen Resiko

Model Waterfall cenderung mengidentifikasi risiko di akhir proyek, khususnya pada tahap pengujian. Akibatnya, kesalahan atau bug yang ditemukan di tahap akhir akan sulit dan mahal untuk diperbaiki karena berdampak pada keseluruhan sistem yang sudah dikembangkan.

Model Agile lebih unggul dalam manajemen risiko karena pengujian dilakukan secara berulang di setiap sprint. Masalah dapat ditemukan dan diatasi lebih awal, sehingga meminimalisasi dampak terhadap keseluruhan sistem. Selain itu, pendekatan iteratif membuat proyek lebih adaptif terhadap risiko perubahan, teknologi baru, atau kebutuhan pasar.

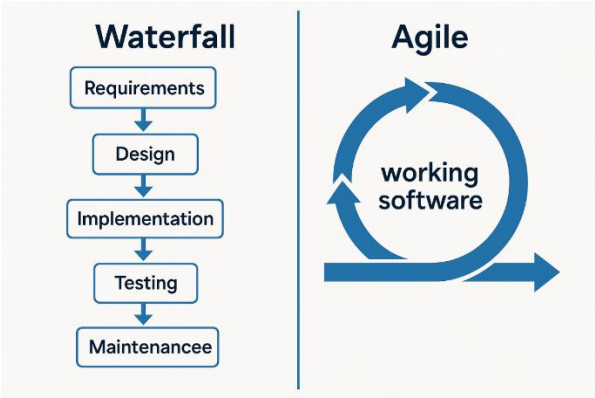
### Kesesuaian Terhadap Skala Proyek

Waterfall lebih cocok untuk proyek skala besar dengan kebutuhan tetap dan spesifikasi teknis yang jelas di awal. Contohnya adalah pengembangan sistem pemerintah, aplikasi perbankan, atau perangkat lunak industri yang regulasinya ketat. Model ini memungkinkan dokumentasi yang lengkap dan audit yang lebih mudah dilakukan.

Agile lebih sesuai untuk proyek kecil hingga menengah yang bersifat dinamis dan terus berkembang, seperti aplikasi startup, platform e-commerce, atau sistem berbasis cloud. Fleksibilitas dan iterasi cepat pada Agile mendukung perubahan berkelanjutan yang sering terjadi dalam proyek jenis ini.

### Tabel Perbandingan Waterfall vs Agile

| Aspek | Waterfall | Agile |
|---|---|---|
| **Fleksibilitas** | Rendah, sulit menyesuaikan perubahan | Tinggi, adaptif terhadap perubahan |
| **Efisiensi Waktu** | Lama, karena proses berurutan | Cepat, karena iterasi singkat dan berulang |
| **Keterlibatan Pengguna** | Minim, hanya di awal dan akhir | Tinggi, terlibat dalam setiap sprint |
| **Manajemen Risiko** | Risiko tinggi, terdeteksi di akhir | Risiko rendah, terdeteksi di awal iterasi |
| **Skala Proyek** | Cocok untuk proyek besar dengan kebutuhan tetap | Cocok untuk proyek kecil-menengah yang dinamis |




### Interpretasi Temuan

Dari hasil analisis di atas, dapat disimpulkan bahwa tidak ada satu model yang paling sempurna untuk semua jenis proyek. Pemilihan model pengembangan perangkat lunak sebaiknya mempertimbangkan konteks proyek, tingkat kompleksitas, kebutuhan fleksibilitas, serta keterlibatan pengguna. Kombinasi kedua pendekatan atau penggunaan metode hybrid (misalnya: Water-Scrum-Fall) juga semakin banyak diterapkan dalam praktik industri untuk menggabungkan kelebihan masing-masing model.

## KESIMPULAN

Berdasarkan hasil kajian dan analisis literatur terhadap model pengembangan perangkat lunak Waterfall dan Agile, dapat disimpulkan bahwa kedua model memiliki keunggulan dan kelemahan masing-masing yang menjadikannya cocok untuk konteks proyek yang berbeda. Model Waterfall menawarkan pendekatan yang terstruktur, sistematis, dan didukung dokumentasi yang lengkap, sehingga sangat sesuai untuk proyek-proyek besar yang memiliki spesifikasi kebutuhan tetap dan jelas sejak awal, seperti pengembangan sistem pemerintahan atau perangkat lunak dengan standar tinggi dan regulasi ketat.

Di sisi lain, model Agile menawarkan fleksibilitas, kecepatan, dan kolaborasi intensif antar tim dan pengguna. Agile lebih unggul dalam hal menangani perubahan kebutuhan secara dinamis, memberikan hasil lebih cepat melalui iterasi, dan meningkatkan kepuasan pengguna dengan umpan balik yang berkelanjutan. Oleh karena itu, Agile sangat ideal untuk proyek dengan ruang lingkup yang berkembang, seperti pengembangan aplikasi startup, layanan digital, dan proyek berbasis cloud.

Dengan demikian, pemilihan model pengembangan perangkat lunak sebaiknya tidak dilakukan secara sembarangan, tetapi berdasarkan pada analisis kebutuhan proyek, karakteristik tim pengembang, tingkat keterlibatan pengguna, dan skala pengembangan. Dalam beberapa kasus, pendekatan hybrid yang menggabungkan elemen-elemen dari kedua model juga dapat menjadi solusi efektif untuk memenuhi kebutuhan proyek secara optimal.

## DAFTAR PUSTAKA


Diansyah, Ahmad Febri, Muhammad Rusdi Rahman, Rizky Handayani, D. Diffran Nur Cahyo, and Ema Utami. 2023. “Comparative Analysis of Software Development Lifecycle Methods in Software Development: A Systematic Literature Review.” *International Journal of Advances in Data and Information Systems* 4 (2): 97–106. https://doi.org/10.25008/ijadis.v4i2.1295.

Hidayah Nova, Sausan, Aris Puji Widodo, Budi Warsito, and Sekolah Pasca Sarjana. 2022. “Analisis Metode Agile Pada Pengembangan Sistem Informasi Berbasis Website: Systematic Literature Review Analysis of Agile Method on Website-Based Information System Development: Systematic Literature Review.” *Februari* 21 (1): 139–48. https://scholar.google.com.

Leong, Jason, Kiu May Yee, Onalethata Baitsegi, Lingesvaran Palanisamy, and R. Kanesaraj Ramasamy. 2023. “Hybrid Project Management between Traditional Software Development Lifecycle and Agile Based Product Development for Future Sustainability.” *Sustainability (Switzerland)* 15 (2). https://doi.org/10.3390/su15021121.

Nurmalasari, Nurmalasari, Yusuf Arif Setiawan, Widi Astuti, M. Rangga Ramadhan Saelan, Siti Masturoh, and Tuti Haryanti. 2023. “Classification for Papaya Fruit Maturity Level With Convolutional Neural Network.” *Jurnal Riset Informatika* 5 (3): 331–38. https://doi.org/10.34288/jri.v5i3.225.

Samsiyah, Siti, Akhmad Arif Musadad, and Musa Pelu. 2020. “Urgency Of Project Based Learning Model In Social Studies Learning To Improve Students Learning Achievement.” *Social, Humanities, and Educational Studies (SHEs): Conference Series* 3 (2): 243. https://doi.org/10.20961/shes.v3i2.46245.

Surahman, Ade, Agung Deni Wahyudi, and Sanriomi Sintaro. 2020. “Implementasi Teknologi Visual 3D Objek Sebagai Media Peningkatan Promosi Produk E-Marketplace.” *Jurnal Buana Informatika* 11 (2): 123–31. https://doi.org/10.24002/jbi.v11i2.3701.

Thesing, Theo, Carsten Feldmann, and Martin Burchardt. 2021. “Agile versus Waterfall Project Management: Decision Model for Selecting the Appropriate Approach to a Project.” *Procedia Computer Science* 181:746–56. https://doi.org/10.1016/j.procs.2021.01.227.