

Makalah

Iterative Development Model

Dosen Pengampuh: Mardiyah Hasnawi, S.Kom., M.T., MTA



Disusun

Oleh:

- | | |
|-----------------------------------|----------------------|
| 1. Fahresa Arif | (13020230354) |
| 2. Rifqy Pratama H Janggih | (13020230102) |
| 3. Muhammad Aulia Rahman | (13020230097) |
| 4. Bagas Nopriyanto | (13020230349) |
| 5. Zakiul Fahmi Gunawan | (13020230345) |

PROGRAM STUDI TEKNIK INFORMATIKA
FAKULTAS ILMU KOMPUTER
UNIVERSITAS MUSLIM INDONESIA
MAKASSAR

2025

Daftar Isi

KATA PENGANTAR	3
1. Pendahuluan	4
1.1 Latar Belakang	4
1.2 Tujuan	4
1.3 Manfaat.....	4
1.4 Rumus Masalah.....	5
2. Kajian Teoritis	5
2.1 Definisi dan Sejarah Model Proses Sesuai Topik	5
2.2 Penerapan Model	6
2.3 Karakteristik Model (Tim, Aliran Proses).....	6
2.4 Kelebihan dan Kelemahan model proses PL.....	6
2.5 Perbedaannya dengan model atau metodologi lainnya.....	7
2.6 Alat Bantu Model.....	7
3. Penutupan	7
3.1 Kesimpulan.....	7
3.2 Saran.....	8
4. Daftar Pustaka	9

Kata Pengantar

Puji syukur kami panjatkan ke hadirat Allah SWT atas rahmat dan hidayah-Nya, sehingga kami dapat menyelesaikan makalah ini dengan judul "Rekayasa Perangkat Lunak Objek: Incremental Model". Makalah ini disusun sebagai bagian dari tugas mata kuliah Rekayasa Perangkat Lunak, dengan tujuan untuk memperdalam pemahaman kami mengenai model pengembangan perangkat lunak yang adaptif dan efisien.

Kami menyadari bahwa dalam penyusunan makalah ini masih terdapat kekurangan, baik dari segi isi maupun penyajian. Oleh karena itu, kami mengharapkan kritik dan saran yang membangun dari pembaca untuk penyempurnaan di masa mendatang. Semoga makalah ini bermanfaat bagi kami sebagai penulis, serta bagi para pembaca yang membutuhkan referensi mengenai Incremental Model dalam rekayasa perangkat lunak objek.

Terima kasih kepada dosen pembimbing dan teman-teman yang telah memberikan dukungan dan masukan selama proses penyusunan. Semoga ilmu yang kami peroleh dapat diterapkan dalam kehidupan sehari-hari dan berkontribusi pada kemajuan teknologi informasi.

Makassar, 27 September 2025

1. Pendahuluan

1.1 Latar Belakang

Dalam era digital yang berkembang pesat, pengembangan perangkat lunak (PL) menjadi tulang punggung berbagai sektor, seperti pendidikan, kesehatan, dan bisnis. Rekayasa Perangkat Lunak (RPL) bertujuan menghasilkan PL yang berkualitas tinggi, efisien, dan dapat diandalkan. Namun, kompleksitas proyek PL sering kali menyebabkan kegagalan, terutama ketika kebutuhan pengguna berubah di tengah proses pengembangan. Menurut laporan Chaos Report oleh Standish Group, lebih dari 30% proyek PL gagal karena kurangnya fleksibilitas dalam metodologi pengembangan [1]. Iterative Development Model muncul sebagai solusi untuk mengatasi tantangan ini dengan pendekatan berulang yang memungkinkan adaptasi terhadap perubahan kebutuhan dan pengelolaan risiko secara dini. Model ini sangat relevan dalam RPL berbasis objek, di mana desain modular mendukung penyempurnaan bertahap.

1.2 Tujuan

Makalah ini bertujuan untuk:

1. Menjelaskan definisi dan sejarah Iterative Development Model.
2. Menguraikan fase-fase penerapan model dalam pengembangan PL.
3. Mengidentifikasi karakteristik model, termasuk peran tim dan aliran proses.
4. Menganalisis kelebihan dan kelemahan model ini.
5. Membandingkan Iterative Development Model dengan model lain, termasuk Waterfall, Spiral, dan Agile.
6. Menyusun daftar alat bantu yang mendukung penerapan model dalam RPL objek..

1.3 Manfaat

Manfaat dari pembahasan Iterative Development Model bermanfaat bagi:

- Bagi Mahasiswa: Menambah pemahaman tentang Iterative Development Model dan aplikasinya dalam RPL berbasis objek.
- Bagi Akademisi: Sebagai referensi untuk penelitian lebih lanjut tentang metodologi pengembangan PL.
- Bagi Praktisi: Memberikan wawasan tentang alat bantu dan strategi penerapan

model untuk proyek PL yang adaptif.

1.4 Rumus Masalah

Rumusan masalah dalam makalah ini mencakup tantangan utama dalam rekayasa perangkat lunak objek, khususnya dalam mengelola proyek-proyek kompleks yang membutuhkan adaptasi terhadap perubahan kebutuhan yang dinamis. Model tradisional sering kali tidak mampu menangani fleksibilitas yang diperlukan dalam pengembangan berbasis objek, sehingga memunculkan beberapa masalah spesifik:

1. Risiko kegagalan proyek akibat pendekatan yang kaku. Model sekuensial seperti Waterfall memiliki risiko tinggi karena deteksi kesalahan yang terlambat akibat penyelesaian fase secara linier, yang tidak mendukung perubahan kebutuhan di tengah pengembangan.
2. Kompleksitas manajemen iterasi dalam desain berbasis objek: Bagaimana memastikan sinkronisasi antara desain kelas dan objek dengan perubahan kebutuhan yang muncul tanpa mengorbankan modularitas dan reusability dalam pengembangan perangkat lunak.
3. Kurangnya pendekatan berulang yang efisien: Tantangan dalam menerapkan pendekatan berulang yang mendukung pengembangan incremental tanpa memerlukan perancangan ulang ekstensif, terutama dalam menjaga integritas hubungan antar-objek.
4. Integrasi umpan balik pengguna secara efektif: Bagaimana mengintegrasikan umpan balik pengguna dalam setiap iterasi untuk meningkatkan kualitas perangkat lunak tanpa menambah biaya atau waktu pengembangan secara signifikan.

2. Kajian Teoritis

2.1 Definisi dan sejarah Model Proses

Iterative Development Model adalah pendekatan pengembangan perangkat lunak yang melibatkan siklus berulang untuk membangun sistem secara bertahap, memungkinkan penyempurnaan berdasarkan umpan balik. Setiap iterasi menghasilkan produk parsial yang dapat diuji dan disempurnakan. Dalam RPL objek, model ini mendukung desain berbasis kelas dan objek yang dapat direfinasi.

Secara historis, konsep iterasi berasal dari siklus PDCA (Plan-Do-Study-Act) Walter Shewhart pada

1930-an, dipopulerkan oleh W. Edwards Deming. Dalam PL, pendekatan ini mulai digunakan pada 1950-an dalam proyek X-15 jet hipersonik dan Proyek Mercury NASA (1960-an). Pada 1968, Brian Randell dan F.W. Zurcher di IBM memperkenalkan pendekatan formal iterative, diikuti oleh proyek besar seperti sistem kapal selam Trident (1972) dan perangkat lunak pesawat ulang-alik NASA (1977-1980). Pada 1990-an, metode seperti Scrum, Extreme Programming (XP), dan Rational Unified Process (RUP) mempopulerkan model ini.

2.2 Penerapan Model (Aktivitas /Fase)

Peran Penerapan model ini melibatkan siklus berulang dengan fase berikut:

1. Requirement Gathering and Analysis: Mengumpulkan dan menganalisis subset kebutuhan untuk iterasi tertentu.
2. Design: Merancang arsitektur dan komponen berbasis objek untuk mendukung modularitas.
3. Implementation/Coding: Mengembangkan kode untuk menghasilkan prototipe atau versi parsial.
versi parsial.
4. Testing: Melakukan pengujian unit, integrasi, dan pengguna untuk memvalidasi iterasi.
5. Deployment and Feedback: Menyebarkan iterasi kepada pengguna untuk mendapatkan umpan balik.

Siklus diulang hingga sistem lengkap, dengan fokus pada refactoring objek dalam RPL.

2.3 Karakteristik Model(Tim, Aliran Proses)

Model ini membutuhkan tim kecil (5-10 orang) yang kolaboratif, terdiri dari developer, tester, dan stakeholder. Aliran proses bersifat siklik, risk-driven, dan feedback-driven, dengan iterasi berlangsung 1-6 minggu. Dalam RPL objek, aliran ini mendukung penyempurnaan kelas dan hubungan antar-objek secara berulang

2.4 Kelebihan dan Kelemahan model proses PL

Kelebihan:

- Pengembangan fungsi cepat dan pengukuran kemajuan berkala.
- Fleksibel terhadap perubahan kebutuhan.

- Manajemen risiko efektif melalui identifikasi dini.

Kelemahan:

- Membutuhkan sumber daya lebih banyak.
- Manajemen kompleks dan membutuhkan skill tinggi.
- Tidak cocok untuk proyek kecil

2.5 Perbandingan Dengan Model Lain

- **Vs. Waterfall:** Waterfall linier dan kaku, sedangkan iterative fleksibel dengan umpan balik berulang. Waterfall cocok untuk proyek kecil, iterative untuk proyek kompleks.
- **Vs. Spiral:** Spiral lebih kompleks dengan fokus pada prototype dan analisis risiko, sedangkan iterative lebih sederhana dan cepat.
- **Vs. Agile:** Agile mencakup iterative namun lebih ringan dan customer-centric, sedangkan iterative lebih terstruktur.

2.6 Alat Bantu Model

Alat bantu yang mendukung model ini meliputi:

- Version Control: Git untuk pengelolaan kode.
- Project Management: Jira, Trello untuk pelacakan iterasi.
- IDE: Eclipse, Visual Studio untuk coding berbasis objek.
- Testing Tools: Selenium, JUnit untuk pengujian otomatis.

3. Penutupan

3.1 Kesimpulan

Secara menyeluruh, Iterative Development Model menawarkan pendekatan yang fleksibel dan adaptif dalam rekayasa perangkat lunak berbasis objek, memungkinkan pengembangan bertahap melalui siklus berulang yang mendukung penyempurnaan berdasarkan umpan balik pengguna. Model ini efektif dalam mengelola risiko melalui identifikasi dini, menjadikannya pilihan ideal untuk proyek kompleks dengan kebutuhan

yang dinamis, sebagaimana diuraikan dalam definisi, sejarah, dan fase penerapannya. Karakteristiknya, seperti tim kolaboratif dan aliran proses yang risk-driven, mendukung efisiensi pengembangan, terutama dalam desain berbasis objek yang menekankan modularitas dan reusability. Meskipun memiliki kelemahan seperti kebutuhan sumber daya yang lebih besar dan kompleksitas manajemen, kelebihanannya dalam fleksibilitas dan kemampuan menangani perubahan kebutuhan menjadikannya unggul dibandingkan model seperti Waterfall, yang kaku, serta Spiral dan Agile, yang memiliki fokus berbeda. Alat bantu modern seperti Git, Jira, dan Selenium memperkuat implementasi model ini, memastikan kolaborasi dan pengujian yang efisien.

3.2 Saran

- Bagi pengembang, disarankan untuk mengintegrasikan alat bantu seperti Jira dan Git sejak awal untuk mempermudah pelacakan iterasi dan pengelolaan kode.
- Tim pengembang sebaiknya dilatih untuk memiliki keterampilan manajemen risiko dan analisis kebutuhan yang kuat agar iterasi berjalan efektif.
- Untuk proyek skala kecil, pertimbangkan model lain seperti Waterfall untuk mengurangi kompleksitas manajemen, tetapi gunakan iterative untuk proyek dengan ketidakpastian kebutuhan tinggi.
- Penelitian lebih lanjut dapat dilakukan untuk mengoptimalkan durasi iterasi dan integrasi alat bantu berbasis AI dalam mendukung Iterative Development Model.

4. Daftar Pustaka

- C. Larman and V. R. Basili, "Iterative and incremental development: A brief history," *Computer*, vol. 36, no. 6, pp. 47–56, 2003.
- I. Alam, N. Sarwar, and I. Noreen, "Statistical analysis of software development models by six-pointed star framework," *PLoS ONE*, vol. 17, no. 4, p. e0264420, 2022.
- P. Banerjee, B. Kumar, A. Singh, A. Singh, and R. Kumari, "Efficiency analysis of software development life cycle models," *Int. J. Comput. Sci. Trends Technol.*, vol. 8, no. 2, pp. 152–162, 2020.
- Q. M. Yas, A. ALazzawi, and B. Rahmatullah, "A comprehensive review of software development life cycle methodologies: Pros, cons, and future directions," *Iraqi J. Comput. Sci. Math.*, vol. 4, no. 4, pp. 90–173, 2023.

