

# **Makalah**

## **Model Dan Paradigma Pengembangan PL Incremental Model**

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



**Disusun**

**Oleh:**

- |                                            |                      |
|--------------------------------------------|----------------------|
| <b>1. Muhammad Rifky Yusri</b>             | <b>(13020230098)</b> |
| <b>2. Muh Akbar Tompo</b>                  | <b>(13020230314)</b> |
| <b>3. Muhammad Raffi Rizkitullah Djiko</b> | <b>(13020230086)</b> |
| <b>4. M Afdhal Hamdan Al-Jaedy</b>         | <b>(13020230069)</b> |
| <b>5. Syahnabil</b>                        | <b>(13020230079)</b> |

**PROGRAM STUDI TEKNIK INFORMATIKA  
FAKULTAS ILMU KOMPUTER  
UNIVERSITAS MUSLIM INDONESIA  
MAKASSAR**

**2025**

# Daftar Isi

## KATA PENGANTAR

<b>1. Pendahuluan</b>	<b>4</b>
1.1 Latar Belakang.....	4
1.2 Tujuan .....	4
1.3 Manfaat.....	4
1.4 Rumus Masalah.....	4
<b>2. Kajian Teoritis</b>	<b>4</b>
2.1 Definisi dan Sejarah Model Proses Sesuai Topik.....	4
2.2 Penerapan Model .....	4
2.3 Karakteristik Model (Tim, Aliran Proses).....	4
2.4 Kelebihan dan Kelemahan model proses PL.....	4
2.5 Perbedaannya dengan model atau metodologi lainnya.....	4
2.6 Alat Bantu Model.....	4
<b>3. Penerapan Dalam Kehidupan Sehari-hari</b>	<b>8</b>
3.1 Pengembangan Aplikasi E-commerce.....	8
3.2 Pengembangan Sistem Operasi seperti Linux.....	8
3.3 Pengembangan Aplikasi Mobile.....	8
<b>4. Penutupan</b>	
<b>4.1 Kesimpulan.....</b>	<b>9</b>
<b>4.2 Saran.....</b>	<b>9</b>
<b>5. Daftar Pustaka.....</b>	<b>10</b>

## **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.

# 1. Pendahuluan

Incremental Model menjadi solusi yang relevan dalam menghadapi tantangan pengembangan perangkat lunak modern, terutama pada proyek-proyek dengan kebutuhan yang terus berubah, seperti aplikasi berbasis cloud, sistem IoT, atau platform berbasis AI. Pendekatan ini memungkinkan pengembang untuk menghasilkan produk fungsional lebih cepat melalui iterasi kecil, sehingga meminimalkan risiko kegagalan proyek akibat perubahan spesifikasi di tengah pengembangan. Selain itu, model ini mendukung pengembangan berbasis komponen, di mana setiap increment dapat dianggap sebagai modul independen yang memudahkan skalabilitas dan pemeliharaan sistem.

Dalam konteks rekayasa perangkat lunak objek, Incremental Model memfasilitasi penggunaan prinsip-prinsip seperti encapsulation, inheritance, dan polymorphism untuk membangun sistem yang modular dan mudah diadaptasi. Studi oleh Boehm et al. (2014) menunjukkan bahwa pendekatan ini dapat mengurangi risiko kegagalan hingga 50% dibandingkan model linier seperti Waterfall, terutama pada proyek dengan ketidakpastian tinggi.

## 1.1 Latar Belakang

Dalam rekayasa perangkat lunak objek, pengembangan sistem semakin kompleks karena tuntutan akan concurrency, modularitas, dan adaptasi terhadap perubahan kebutuhan pengguna yang dinamis.

Latar belakang munculnya berbagai model pengembangan perangkat lunak (Software Development Life Cycle/SDLC) adalah untuk mengelola risiko, biaya, dan waktu dalam proyek teknologi, di mana model tradisional seperti Waterfall sering kali kurang fleksibel. Waterfall, sebagai model sekuensial, mengharuskan penyelesaian setiap fase (requirements, design, implementation, testing, deployment, maintenance) secara linear sebelum melanjutkan, yang dapat menyebabkan deteksi kesalahan terlambat dan biaya rework tinggi jika persyaratan berubah. Sebagai alternatif,

Incremental Model menawarkan pendekatan iteratif di mana sistem dibangun melalui serangkaian increment kecil, masing-masing mencakup siklus mini dari desain hingga pengujian, sehingga menghasilkan versi kerja parsial yang dapat diuji dan disempurnakan secara bertahap. Model ini menggabungkan elemen Waterfall dalam setiap increment tetapi diterapkan secara berulang,

memungkinkan integrasi objek-oriented seperti class dan method dengan evolusi bertahap untuk mendukung prinsip encapsulation dan polymorphism. Pemilihan topik ini relevan karena model ini lebih adaptif terhadap proyek dengan persyaratan yang berevolusi, seperti dalam pengembangan aplikasi mobile atau sistem enterprise berbasis objek, di mana feedback pengguna awal dapat mengurangi risiko kegagalan hingga 50% dibandingkan model linier.

## **1.2 Tujuan**

Tujuan makalah ini adalah:

- (1) Menjelaskan konsep Incremental Model dalam rekayasa perangkat lunak objek, mencakup definisi, sejarah, proses, karakteristik, kelebihan-kelemahan, perbandingan dengan Waterfall, dan alat bantu;
- (2) Menganalisis bagaimana model ini mengintegrasikan prinsip object-oriented untuk mendukung pengembangan bertahap dan adaptif;
- (3) Memberikan wawasan tentang aplikasi model ini dalam proyek kompleks, sehingga pembaca memahami manfaatnya untuk efisiensi dan kualitas sistem berbasis objek.

## **1.3 Manfaat**

Manfaat dari pembahasan Incremental Model ini meliputi:

- (1) Memberikan panduan praktis bagi praktisi rekayasa perangkat lunak objek untuk menerapkan pendekatan incremental, mengurangi waktu pengembangan dan meningkatkan skalabilitas sistem;
- (2) Meningkatkan pemahaman tentang integrasi model dengan metodologi seperti Rational Unified Process (RUP) atau agile, yang mendukung concurrency dan modularitas objek;
- (3) Memberikan nilai edukatif bagi mahasiswa dan peneliti, dengan contoh aplikasi seperti incremental synchronization model organisasi, sehingga dapat diterapkan dalam proyek real-world untuk mengurangi risiko dan biaya.

## **1.4 Rumus Masalah**

Rumusan masalah dalam makalah ini mencakup tantangan utama dalam rekayasa perangkat lunak objek, seperti bagaimana mengelola kompleksitas sistem besar dengan skalabilitas tinggi, di mana model tradisional sering gagal karena kurangnya fleksibilitas terhadap perubahan emergent. Masalah spesifik termasuk:

- (1) Risiko integrasi akhir yang tinggi pada model sekuensial seperti Waterfall, menyebabkan deteksi kesalahan terlambat;
- (2) Kebutuhan sinkronisasi model organisasi dengan perubahan dinamis dalam pengembangan berbasis objek;
- (3) Kurangnya pendekatan incremental yang mendukung concurrency objek tanpa memerlukan modifikasi desain ekstensif selama eksekusi; serta
- (4) Tantangan dalam mengintegrasikan feedback pengguna untuk meningkatkan kualitas tanpa meningkatkan biaya secara signifikan. Masalah-masalah ini menjadi dasar untuk mengeksplorasi Incremental Model sebagai solusi yang lebih adaptif.

## **2. Kajian Teoritas**

Incremental Model memiliki keunggulan dalam mengelola kompleksitas proyek besar melalui pendekatan iteratif yang terstruktur. Berbeda dengan model tradisional seperti Waterfall, model ini memungkinkan pengujian dan validasi awal pada setiap increment, sehingga masalah dapat diidentifikasi dan diperbaiki lebih cepat. Dalam konteks historis, model ini terinspirasi dari Cleanroom Software Engineering pada 1980-an, yang menekankan verifikasi formal untuk meningkatkan keandalan perangkat lunak.

Perkembangan lebih lanjut pada 2000-an mengintegrasikan model ini dengan metodologi seperti Rational Unified Process (RUP) dan Agile, yang memungkinkan pengembangan berbasis objek dengan fase-fase seperti Inception, Elaboration, Construction, dan Transition. Pendekatan ini sangat cocok untuk proyek dengan arsitektur modular, di mana setiap increment dapat dikembangkan sebagai komponen independen yang mendukung prinsip object-oriented seperti reusability dan maintainability. Menurut Larman & Basili (2003), model ini meningkatkan efisiensi pengembangan hingga 20% pada proyek dengan persyaratan yang dinamis dibandingkan model sekuensial

### **2.1 Definisi dan sejarah Model/Proses Sesuai Topik**

Incremental Model didefinisikan sebagai pendekatan pengembangan perangkat lunak yang membagi persyaratan menjadi bagian-bagian kecil (increment) yang dikembangkan secara independen dari spesifikasi hingga kode executable, memungkinkan pengujian paralel dan pertumbuhan sistem secara bertahap.

Dalam konteks rekayasa perangkat lunak objek, model ini mendukung evolusi sistem melalui iterasi yang memanfaatkan prinsip seperti encapsulation dan inheritance. Secara historis, model ini berakar pada Cleanroom Software Engineering yang dikembangkan oleh Mills pada 1980-an, sebagai respons terhadap kebutuhan kontrol kualitas awal, dan telah berevolusi menjadi Iterative and Incremental Development (IID) yang terintegrasi dengan metodologi agile sejak awal 2000-an. Evolusi ini didorong oleh studi seperti Larman & Basili (2003), yang menunjukkan keunggulannya dalam menangani persyaratan berubah.

### **2.2 Penerapan Model (Aktivitas /Fase)**

Peran utama Incremental Model adalah membangun perangkat lunak melalui siklus iteratif, di mana setiap fase melibatkan identifikasi persyaratan, analisis, desain spesifikasi, coding, dan pengumpulan feedback stakeholder sebelum melanjutkan ke increment berikutnya.

Aktivitas mencakup pengembangan increment menggunakan model waterfall mini atau spiral, diikuti sertifikasi reliabilitas melalui usage testing, hingga sistem lengkap tercapai. Dalam rekayasa objek, fase ini mendukung refactoring objek secara bertahap, dengan maintenance berkelanjutan setelah setiap siklus.

## 2.3 Karakteristik Model(Tim, Aliran Proses)

Karakteristik Incremental Model mencakup pembagian increment berdasarkan risiko dan penggunaan, dengan tim yang terpisah untuk pengembangan dan pengujian untuk memastikan independensi. Tim memerlukan keterampilan komunikasi tinggi dan interaksi konstan dengan klien untuk menangani perubahan persyaratan. Aliran proses bersifat paralel, di mana pengembangan dan pengujian berjalan bersamaan,

memungkinkan iterasi kecil untuk identifikasi isu dini dan evolusi desain, cocok untuk proyek objek-oriented dengan arsitektur modular. Pendekatan ini berkembang melalui siklus berulang perencanaan, desain, implementasi, dan evaluasi, memungkinkan penyempurnaan berkelanjutan berdasarkan umpan balik, dan tidak semua iterasi menghasilkan increment nyata—beberapa fokus pada pembelajaran atau eksplorasi kelayakan teknis

## 2.4 Kelebihan dan Kelemahan model proses PL

### Kelebihan

- Memungkinkan pengujian awal dan deteksi kesalahan melalui siklus kecil.
- Mendukung persyaratan yang berubah dan memberikan akses lebih awal kepada pengguna dibandingkan model sekuensial, sehingga mengurangi risiko integrasi besar-besaran.
- Memungkinkan pengiriman awal dan umpan balik lebih cepat, mengurangi risiko melalui siklus pembelajaran, dan memfasilitasi penyelarasan pemangku kepentingan melalui interaksi sering.
- Menghasilkan perangkat lunak kerja dengan cepat, lebih fleksibel untuk perubahan ruang lingkup, lebih mudah diuji dalam iterasi kecil, dan lebih baik dalam mengelola risiko (berdasarkan studi komparatif).
- Kemampuan pengiriman hasil bertahap dalam waktu singkat, pemantauan klien yang ketat, dan penyesuaian selama pengembangan, cocok untuk perusahaan yang mengutamakan kualitas tinggi dengan manajemen perubahan efisien (berdasarkan artikel lain).

### Kelemahan

- Risiko scope creep dari umpan balik sering yang menyebabkan loop pengembangan tak berujung.
- Arsitektur sistem yang mungkin terpengaruh di iterasi akhir karena persyaratan tidak dikumpulkan sepenuhnya di awal, dan risiko desain kode buruk dari pendekatan build-and-patch.
- Memerlukan kontrol versi yang kuat dan dokumentasi jelas karena evolusi persyaratan, perencanaan increment yang menantang, dan kolaborasi intensif yang memakan sumber daya.
- Fase iterasi yang kaku tanpa tumpang tindih dan masalah arsitektur potensial karena persyaratan tidak lengkap di awal.
- Kemungkinan rework dan kebutuhan manajemen perubahan efektif.

## 2.5 Perbedaannya dengan model atau metodologi lainnya

Incremental Model berbeda dari model lain seperti Spiral (fokus risiko) atau Prototyping (eksplorasi persyaratan). Dibandingkan Waterfall: Incremental memungkinkan pengembangan paralel dan sertifikasi bertahap, sementara Waterfall sekuensial dan hanya menguji di akhir, menunda kontrol kualitas. Incremental lebih unggul dalam biaya, durasi, dan kualitas untuk proyek berubah, tidak seperti Waterfall yang linear dan kurang fleksibel. Contoh: Incremental mencapai reliabilitas sama dalam waktu lebih singkat (2737 vs. 3014 unit waktu). Dibandingkan Waterfall, model incremental lebih fleksibel dan iteratif, memungkinkan integrasi dan pengujian awal daripada menunggu akhir, mengurangi risiko penemuan isu terlambat. Studi komparatif menunjukkan Waterfall cocok untuk proyek dengan persyaratan yang dipahami dengan baik, sementara Incremental ideal untuk proyek berisiko tinggi di mana pengembangan keseluruhan sekaligus berbahaya, memungkinkan pengiriman fungsionalitas dasar awal

Dibandingkan Agile: Incremental lebih terstruktur dengan increment tetap, sementara Agile lebih adaptif dengan sprint. Agile menekankan kolaborasi harian, sedangkan Incremental fokus pada rilis bertahap. Menurut penelitian, Incremental cocok untuk proyek dengan timeline panjang, sementara Agile untuk perubahan cepat. Perbandingan ini menunjukkan bahwa Incremental adalah hybrid antara Waterfall dan Agile.

## 2.6 Alat Bantu Model

Alat bantu mencakup UML untuk pemodelan relasi user-sistem, mendukung rekayasa objek. Arsitektur 3-tier (GUI, logic, data layer) dengan bahasa seperti PHP, serta teknik usage testing seperti Markov chain untuk sertifikasi reliabilitas. Tools lain termasuk Box Structures untuk desain dan Stepwise Refinement untuk implementasi dalam konteks Cleanroom.

Alat modern meliputi:

- **Version Control Systems:** Git untuk mengelola perubahan increment.
- **Project Management Tools:** Jira atau Trello untuk tracking increment.
- **CI/CD Pipelines:** Jenkins untuk otomatisasi testing dan deployment.
- **Modeling Tools:** Enterprise Architect untuk UML diagram.

Ini memungkinkan tim untuk mengintegrasikan alat dengan proses incremental secara efisien



### **3. Penerapan Dalam Kehidupan Sehari-hari**

Penerapan Incremental Model dalam kehidupan sehari-hari sangat luas, terutama dalam pengembangan teknologi yang membutuhkan adaptasi cepat terhadap perubahan pasar atau kebutuhan pengguna. Dalam pengembangan aplikasi e-commerce, seperti Amazon, model ini memungkinkan peluncuran fitur dasar seperti pencarian produk dan katalog dalam waktu singkat, diikuti oleh penambahan fitur seperti rekomendasi berbasis AI atau integrasi pembayaran digital berdasarkan feedback pengguna.

Dalam pengembangan sistem operasi seperti Linux, Incremental Model mendukung kolaborasi global melalui komunitas open-source, di mana setiap kontribusi (seperti driver baru atau fitur keamanan) diintegrasikan secara bertahap tanpa mengganggu stabilitas sistem. Untuk aplikasi mobile seperti WhatsApp, pendekatan ini memungkinkan pengembang untuk merilis fitur inti seperti pesan teks terlebih dahulu, kemudian menambahkan fitur kompleks seperti panggilan video atau enkripsi end-to-end secara bertahap. Pendekatan ini tidak hanya mengurangi risiko kegagalan tetapi juga memungkinkan aplikasi untuk tetap kompetitif dengan merespons tren teknologi, seperti integrasi AI atau kebutuhan privasi data, dengan lebih cepat dan efisien.

#### **3.1 Pengembangan Aplikasi E-commerce**

Salah satu contoh penerapan Model Incremental adalah dalam pengembangan aplikasi e-commerce, seperti platform belanja online Amazon atau Shopify. Pengembangan dimulai dengan increment awal berupa fitur pencarian dan tampilan produk yang fungsional, di mana tim menganalisis kebutuhan dasar, mendesain antarmuka sederhana, mengimplementasikan kode, dan menguji modul tersebut sebelum dirilis sebagai MVP (Minimum Viable Product).

Kemudian, secara bertahap ditambahkan fitur seperti keranjang belanja untuk menyimpan item, integrasi pembayaran dengan gateway seperti Stripe atau PayPal, dan pelacakan pesanan real-time menggunakan API logistik. Pendekatan ini memungkinkan peluncuran cepat versi awal ke pasar, pengumpulan feedback pengguna untuk perbaikan, serta penyesuaian fitur berdasarkan data penggunaan, sehingga mengurangi risiko kegagalan keseluruhan proyek dan memastikan aplikasi tetap kompetitif di pasar yang dinamis.

#### **3.2 Pengembangan Sistem Operasi seperti Linux**

Kernel Linux dikembangkan menggunakan Model Incremental, di mana setiap versi baru dibangun berdasarkan versi sebelumnya dengan penambahan fitur, perbaikan bug, dan optimalisasi secara bertahap. Misalnya, increment awal mungkin fokus pada kernel dasar untuk manajemen hardware, kemudian ditambahkan modul seperti dukungan driver baru, fitur keamanan seperti SELinux, atau integrasi dengan teknologi cloud.

Komunitas open-source berkontribusi melalui pull requests dan releases minor, memungkinkan evolusi sistem operasi yang stabil dan adaptif terhadap

kebutuhan pengguna tanpa harus merombak keseluruhan kode dari awal. Model ini sangat cocok untuk proyek besar seperti Linux karena memungkinkan pengujian dini pada setiap increment, mengurangi kompleksitas, dan memfasilitasi kolaborasi global, sehingga Linux telah berkembang dari kernel sederhana pada 1991 menjadi sistem operasi yang mendukung jutaan perangkat saat ini

### **3.3 Pengembangan Aplikasi Mobile**

Dalam pengembangan aplikasi mobile, Model Incremental diterapkan pada aplikasi pesan seperti WhatsApp atau Telegram, dimulai dengan fitur dasar pengiriman teks yang melibatkan analisis kebutuhan komunikasi sederhana, desain UI minimalis, implementasi backend untuk sinkronisasi pesan, dan pengujian keamanan enkripsi end-to-end. Kemudian, incrementally menambahkan media sharing untuk gambar dan video, panggilan suara, video call, serta fitur grup dan channel.

Contoh lain adalah Gmail, yang diperbarui secara incremental dengan penambahan fitur seperti peningkatan storage dari 1GB awal menjadi unlimited, integrasi AI untuk sorting email pintar, dan antarmuka baru berdasarkan feedback pengguna. Pendekatan ini memastikan aplikasi tetap relevan, dapat dirilis secara bertahap ke pasar melalui update app store, dan memungkinkan iterasi cepat untuk menanggapi tren seperti integrasi AI atau privasi data, sehingga aplikasi mobile seperti ini dapat berkembang dari prototipe sederhana menjadi ekosistem lengkap yang digunakan miliaran orang.

## **4. Penutupan**

### **4.1 Kesimpulan**

Incremental Model merupakan pendekatan yang efektif dalam rekayasa perangkat lunak objek untuk mengatasi kompleksitas dan perubahan dinamis, dengan membangun sistem secara bertahap melalui increment kecil. Model ini mengintegrasikan prinsip object-oriented seperti encapsulation dan inheritance, memungkinkan pengujian awal, umpan balik cepat, dan pengurangan risiko.

Dari kajian teoritis, terlihat bahwa model ini unggul dalam fleksibilitas dibandingkan Waterfall, meskipun memiliki tantangan seperti manajemen scope dan rework. Aplikasi nyata di e-commerce, sistem operasi seperti Linux, dan aplikasi mobile menunjukkan keberhasilannya dalam proyek real-world. Secara keseluruhan, Incremental Model meningkatkan efisiensi, kualitas, dan adaptabilitas dalam pengembangan perangkat lunak, terutama untuk proyek dengan ketidakpastian tinggi.

### **4.2 Saran**

Untuk penerapan lebih lanjut, disarankan agar tim pengembang fokus pada manajemen perubahan yang kuat untuk menghindari scope creep, serta integrasi alat

bantu seperti UML dan teknik testing awal. Penelitian selanjutnya dapat mengeksplorasi hybridisasi dengan model agile untuk proyek skala besar. Mahasiswa dan praktisi dianjurkan untuk menerapkan model ini dalam proyek kecil untuk memahami manfaatnya secara praktis.

## 5. Daftar Pustaka

- Larman, C., & Basili, V. R. (2003). Iterative and Incremental Development: A Brief History. *IEEE Computer*, 36(6), 47-56.
- Boehm, B. (1987). A Spiral Model of Software Development and Enhancement. *ACM SIGSOFT Software Engineering Notes*, 11(4), 14–24.
- Boehm, B., Lane, J., Koolmanojwong, S., & Turner, R. (2014). *The Incremental Commitment Spiral Model: Principles and Practices for Successful Systems and Software*. Addison-Wesley Professional.
- Incremental Model - an overview. ScienceDirect Topics. Diakses dari <https://www.sciencedirect.com/topics/computer-science/incremental-model>.
- Incremental Development Approach. SEBoK. Diakses dari [https://sebokwiki.org/wiki/Incremental\\_Development\\_Approach](https://sebokwiki.org/wiki/Incremental_Development_Approach).
- A Comparative Study between Waterfall and Incremental Software Development Life Cycle Model. Diakses dari <https://journals.indexcopernicus.com/api/file/viewById/174825.pdf>.
- The Incremental Model in Software Development: A Structured and Interactive Way to Deliver Quality Products. *Research, Society and Development*. Diakses dari <https://rsdjournal.org/rsd/article/view/40934>.