

MAKALAH
RATIONAL UNIFIED PROCESS (RUP)



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

Disusun oleh:

KELOMPOK V

Syarah Yanti (13020230038)

Nova Febryna. A (13020230040)

Sahra Zhafirah (13020230043)

Khayla Alifia Indrawati (13020230329)

PROGRAM STUDI TEKNIK INFORMATIKA
FAKULTAS ILMU KOMPUTER
UNIVERSITAS MUSLIM INDONESIA
MAKASSAR
2025

DAFTAR ISI

DAFTAR ISI	ii
BAB I	1
PENDAHULUAN	1
A. Latar Belakang	1
B. Tujuan Penulisan	1
BAB II	3
KAJIAN TEORITIS	3
A. Definisi dan Sejarah	3
B. Penerapan Model	3
C. Karakteristik Model	4
D. Kelebihan dan Kelemahan	5
E. Perbedaan dengan Model atau Metodologi Lainnya	6
F. Alat Bantu Model atau Metode	8
BAB III	10
PENUTUP	10
A. Kesimpulan	10
DAFTAR PUSTAKA	11

BAB I

PENDAHULUAN

A. Latar Belakang

Philippe Kruchten dalam bukunya *The Rational Unified Process: An Introduction* memperkenalkan Rational Unified Process (RUP) sebagai kerangka proses rekayasa perangkat lunak yang bersifat iteratif dan incremental, menekankan aspek arsitektur sistem (*architecture-centric*), berbasis *use-case*, serta didukung dokumentasi yang sistematis [1].

Pemilihan topik RUP didasarkan pada kemampuannya mendeteksi risiko lebih awal, menjaga kualitas sistem melalui pendekatan arsitektur, dan memberikan fleksibilitas menghadapi perubahan kebutuhan pengguna. Hal ini menjadikan RUP relevan untuk dipelajari karena masih digunakan sebagai acuan dalam proyek perangkat lunak skala menengah hingga besar.

Secara teoritis, RUP memiliki sejarah dari *Objectory Process*, diterapkan melalui empat fase utama (*inception, elaboration, construction, transition*), dengan karakteristik berorientasi objek, iteratif, terdokumentasi, serta berbasis risiko. Metode ini memiliki kelebihan seperti dukungan UML dan kontrol perubahan sistematis, namun juga kelemahan berupa waktu dan biaya yang lebih besar. Dibandingkan dengan Waterfall, Prototyping, Agile Unified Process (AUP), dan Rapid Application Development (RAD), RUP unggul dalam dokumentasi dan manajemen risiko, meski kurang cepat dan fleksibel. Penerapannya turut didukung oleh alat bantu seperti **Enterprise Architect** untuk pemodelan UML dan **Simphony.NET** untuk simulasi proses.

Latar belakang penulisan ini berangkat dari kebutuhan metodologi yang mampu mengatasi kompleksitas sistem, keterbatasan waktu, serta pentingnya dokumentasi dan pengendalian risiko. RUP dipandang sebagai solusi metodologis yang sistematis dan relevan untuk menjawab tantangan tersebut.

B. Tujuan Penulisan

Makalah ini memiliki tujuan sebagai berikut:

- Menguraikan definisi dan sejarah dari Rational Unified Process.
- Menjelaskan fase-fase, aktivitas, dan alur kerja dalam Rational Unified Process beserta contoh penerapannya.
- Mengidentifikasi karakteristik utama Rational Unified Process, kelebihan dan kelemahannya.

- Melakukan perbandingan antara Rational Unified Process dan metodologi pengembangan perangkat lunak lainnya untuk memahami keunggulan dan keterbatasannya.
- Mendeskripsikan alat bantu yang mendukung penerapan Rational Unified Process dalam pengembangan perangkat lunak

BAB II

KAJIAN TEORITIS

A. Definisi dan Sejarah

Rational Unified Process (RUP) adalah suatu kerangka proses rekayasa perangkat lunak yang memberikan pendekatan disiplin dalam pembagian tugas dan tanggung jawab di dalam organisasi pengembangan. Tujuannya adalah memastikan terciptanya perangkat lunak berkualitas tinggi sesuai kebutuhan pengguna, dalam jadwal dan anggaran yang dapat diprediksi. RUP bersifat iteratif dan incremental, berbasis objek, serta berperan sebagai panduan praktik terbaik melalui basis pengetahuan yang berisi pedoman, template, dan contoh pada setiap tahap pengembangan perangkat lunak [2].

Sejarah Rational Unified Process berawal dari Objectory Process yang dikembangkan oleh Ivar Jacobson pada tahun 1987 di Swedia berdasarkan pengalamannya di Ericsson. Setelah Rational Software Corporation mengakuisisi Objectory AB pada tahun 1995, lahirlah Rational Objectory Process versi 4 pada tahun 1996. Proses ini kemudian berkembang menjadi RUP dengan integrasi konsep iterasi dari pendekatan Rational dan konsep *use-case* dari Objectory. Selain itu, RUP juga diperkaya dengan praktik manajemen kebutuhan dari Requisite Inc., proses pengujian dari SQA Inc., serta diintegrasikan dengan Unified Modeling Language (UML) yang pada awalnya dikembangkan oleh Rational Software. Seiring waktu, RUP menjadi salah satu standar industri dalam metodologi pengembangan perangkat lunak [2].

B. Penerapan Model

Rational Unified Process (RUP) diterapkan melalui empat fase utama, yaitu *inception*, *elaboration*, *construction*, dan *transition*.

- **Inception**

Tahap ini lebih pada memodelkan proses bisnis yang dibutuhkan (business modeling), mendefinisikan kebutuhan sistem yang akan dibuat (requirement) serta analisis dan desain [3].

- **Elaboration**

Tahap ini lebih pada analisis dan desain sistem serta implementasi sistem yang fokus pada purwarupa sistem (prototype) [3].

- **Construction**

Tahap ini lebih pada implementasi dan pengujian sistem yang fokus pada implementasi perangkat lunak pada kode program [3].

- **Transition**

Tahap ini lebih pada deployment atau instalasi sistem agar dapat dimengerti oleh user [3].

Setiap fase memiliki tujuan spesifik dan selalu melibatkan proses pengujian sehingga kebutuhan sistem dapat divalidasi secara bertahap. Pendekatan iteratif dan incremental ini memungkinkan pengembang untuk mengidentifikasi kebutuhan secara lebih jelas, mengurangi risiko kesalahan, serta meningkatkan kualitas perangkat lunak yang dihasilkan. Dengan demikian, penerapan RUP tidak hanya mengikuti alur berurutan, tetapi juga melibatkan siklus umpan balik (*feedback loop*) untuk penyempurnaan di setiap tahap [3].

Sebagai contoh, pada pengembangan **Sistem Informasi Medical Check-Up di Citra Medical Centre**, tahap *inception* digunakan untuk pemodelan bisnis dengan *Business Model Canvas*, identifikasi kebutuhan fungsional, dan pembuatan *use case diagram*. Tahap *elaboration* melibatkan perancangan *activity diagram*, *class diagram*, *sequence diagram*, serta *Entity Relationship Diagram (ERD)*, dilanjutkan dengan validasi fitur bersama pengguna. Pada tahap *construction*, dilakukan implementasi database, pengembangan aplikasi berbasis web dengan konsep Model-View-Controller (MVC), serta pengujian menggunakan *black box testing*. Terakhir, pada tahap *transition* dilakukan uji *usability* yang melibatkan pegawai klinik dan masyarakat sebagai pengguna, dengan hasil kelayakan sebesar 86,12% yang dikategorikan "sangat layak" [3].

C. Karakteristik Model

Model Rational Unified Process (RUP) memiliki sejumlah karakteristik utama yang membedakannya dari metodologi pengembangan perangkat lunak lainnya:

- **Berbasis Objek**

Rational Unified Process (RUP) mendukung pengembangan perangkat lunak berorientasi objek, di mana sistem dimodelkan sebagai kumpulan objek yang saling berinteraksi [4].

- **Iteratif dan Inkremental**

Proses pengembangan dipecah menjadi serangkaian iterasi, dan setiap iterasi menghasilkan tambahan fungsionalitas sehingga sistem berkembang secara bertahap [4].

- **Fase-Fase Pengembangan**

Rational Unified Process (RUP) terdiri atas empat fase utama: *Inception* (pembentukan ruang lingkup dan tujuan proyek), *Elaboration* (analisis dan perancangan sistem), *Construction* (implementasi dan pengujian), serta *Transition* (penyerahan sistem ke pengguna) [4].

- **Disiplin dan Artefak**

Rational Unified Process (RUP) memiliki berbagai disiplin (disciplines) seperti analisis bisnis, manajemen proyek, pengujian, dan pemeliharaan. Setiap disiplin menghasilkan artefak berupa dokumen atau model, misalnya spesifikasi kebutuhan, desain sistem, dan dokumentasi pengujian [4].

- **Arsitektur-Centric**

Arsitektur sistem menjadi pusat pengembangan, dirancang dan dipelihara dengan hati-hati sepanjang siklus hidup perangkat lunak [4].

- **Orientasi pada Risiko**

Rational Unified Process (RUP) secara eksplisit mengidentifikasi dan mengelola risiko sejak tahap awal, sehingga dapat mengurangi ketidakpastian proyek dan meningkatkan peluang keberhasilan [4].

- **Berbasis Dokumen**

Rational Unified Process (RUP) menghasilkan dokumentasi yang lengkap, mencakup kebutuhan, desain, implementasi, serta pengujian perangkat lunak. Dokumentasi ini berfungsi sebagai acuan untuk pengembangan berkelanjutan [4].

Karakteristik tersebut menjadikan Rational Unified Process (RUP) sebagai metodologi yang sistematis, terstruktur, dan sesuai untuk proyek perangkat lunak skala menengah hingga besar yang membutuhkan dokumentasi kuat, kontrol risiko, serta fokus pada kualitas arsitektur [4].

D. Kelebihan dan Kelemahan

- **Kelebihan**

1. Menyediakan akses mudah terhadap pengetahuan dasar bagi anggota tim [5].
2. Memberikan panduan penggunaan UML secara efektif [5].
3. Mendukung proses iteratif dalam pengembangan perangkat lunak [5].
4. Memungkinkan adanya penambahan atau perluasan proses sesuai kebutuhan [5].
5. Memfasilitasi kontrol perubahan perangkat lunak secara sistematis sepanjang pengembangan [5].

6. Mendukung pengujian perangkat lunak secara berulang untuk meningkatkan kualitas [5].

- **Kelemahan**

1. Hanya sesuai untuk pengembangan perangkat lunak berorientasi objek dengan fokus pada UML [5].
2. Membutuhkan waktu lebih lama dibandingkan metode lain [5].
3. Biaya implementasi relatif lebih tinggi karena mencakup analisis risiko yang mendalam, dokumentasi terperinci, serta penggunaan sumber daya yang lebih besar [5].

E. Perbedaan dengan Model atau Metodologi Lainnya

- **Perbandingan Rational Unified Process (RUP) dan Waterfall Model**

Rational Unified Process (RUP) dan Waterfall merupakan dua model pengembangan perangkat lunak dengan karakteristik yang berbeda. RUP bersifat fleksibel karena pengguna dapat melakukan perubahan pada setiap tahapan pengembangan. Kelebihan ini membuat RUP sesuai untuk proyek yang kebutuhan sistemnya masih mungkin berubah. Namun, fleksibilitas tersebut juga menjadi kelemahan karena dapat memperpanjang waktu pengembangan akibat terlalu banyak revisi [3].

Sebaliknya, model Waterfall memiliki tahapan yang pasti dan berurutan sehingga lebih mudah dipahami dan diterapkan. Kelemahannya adalah sifatnya yang kaku, karena tidak mendukung revisi atau perubahan dalam jumlah besar setelah tahapan tertentu dilalui. Dengan demikian, RUP lebih unggul untuk proyek yang membutuhkan fleksibilitas, sedangkan Waterfall lebih sesuai untuk proyek yang sederhana dan kebutuhan sistemnya relatif stabil [3].

- **Perbandingan Rational Unified Process (RUP) dan Prototyping Model**

Rational Unified Process (RUP) dan Prototyping merupakan dua metode pengembangan perangkat lunak dengan pendekatan yang berbeda. RUP menggunakan empat fase utama, yaitu *inception*, *elaboration*, *construction*, dan *transition*, yang menekankan analisis kebutuhan, perancangan sistem, implementasi, serta pengujian dan penyerahan. Sebaliknya, Prototyping menekankan siklus cepat melalui tahap pengumpulan kebutuhan, desain awal, pembangunan *prototype*, evaluasi bersama pengguna, serta perbaikan hingga produk akhir [5].

Dari sisi kebutuhan, RUP lebih cocok untuk proyek dengan spesifikasi yang jelas sejak awal, sedangkan Prototyping sesuai untuk kebutuhan yang masih

belum pasti dan memerlukan validasi pengguna. Dari aspek risiko, RUP memiliki analisis risiko yang lebih sistematis, sementara Prototyping cenderung terbatas. Dari segi waktu dan biaya, Prototyping lebih singkat dan murah, sedangkan RUP relatif lebih lama dan mahal karena dokumentasi dan analisis risiko yang mendalam. RUP juga unggul dalam dokumentasi, keamanan, serta penggunaan UML sebagai alat bantu, sedangkan Prototyping lebih menekankan keterlibatan intensif pengguna dengan alat bantu seperti DFD [5].

Dengan demikian, RUP lebih unggul dalam pengelolaan proyek yang kompleks, berskala besar, dan membutuhkan dokumentasi serta pengendalian risiko, sementara Prototyping lebih sesuai untuk proyek berskala kecil hingga menengah dengan kebutuhan yang belum jelas dan membutuhkan umpan balik pengguna secara cepat [5].

- **Perbandingan Rational Unified Process (RUP) dan Agile Unified Process (AUP)**

Agile Unified Process (AUP) merupakan versi penyederhanaan dari Rational Unified Process (RUP) yang dikembangkan oleh Scott Ambler dengan mengintegrasikan praktik *Agile Modeling* ke dalam kerangka RUP. AUP tetap mempertahankan empat fase utama RUP, yaitu *inception*, *elaboration*, *construction*, dan *transition*, namun dijalankan dalam iterasi singkat dengan durasi sekitar dua minggu [6].

Jika RUP dikenal sebagai proses yang iteratif, *use-case driven*, dan *architecture-centric*, dengan cakupan dokumentasi yang relatif besar, maka AUP menawarkan pendekatan yang lebih ringan dengan dokumentasi yang ringkas dan fokus pada nilai-nilai *Agile Manifesto*. Hal ini mencakup kolaborasi dengan pengguna, perangkat lunak yang berfungsi sebagai ukuran utama kemajuan, serta kemampuan merespons perubahan secara cepat [6].

- **Perbandingan Rational Unified Process (RUP) dan Rapid Application Development (RAD)**

Rational Unified Process (RUP) dan Rapid Application Development (RAD) memiliki pendekatan yang berbeda dalam pengembangan perangkat lunak. RUP bersifat iteratif, *use-case driven*, dan *architecture-centric*, dengan empat fase utama (*inception*, *elaboration*, *construction*, *transition*). Model ini cocok digunakan untuk proyek berskala besar karena menghasilkan dokumentasi lengkap, mendukung analisis risiko secara sistematis, serta lebih terstruktur. Kelemahannya terletak pada waktu dan biaya implementasi yang relatif tinggi [7].

Sebaliknya, RAD menekankan kecepatan pengembangan melalui siklus singkat sekitar 60–90 hari dengan keterlibatan intensif pengguna. RAD berfokus pada pembangunan komponen yang dapat digunakan kembali (*reusable components*) dan menghasilkan produk awal dengan cepat. Kelebihannya adalah waktu pengembangan yang singkat dan biaya yang lebih rendah, tetapi kelemahannya terletak pada dokumentasi yang terbatas, risiko kegagalan komitmen antara pengguna dan pengembang, serta kurang sesuai untuk proyek berskala besar [7].

Dengan demikian, RUP lebih unggul untuk sistem kompleks yang membutuhkan dokumentasi kuat, struktur jelas, dan pengelolaan risiko, sedangkan RAD lebih sesuai untuk proyek kecil hingga menengah yang membutuhkan kecepatan pengembangan dan keterlibatan pengguna secara intensif [7].

F. Alat Bantu Model atau Metode

Dalam penerapan Rational Unified Process (RUP), penggunaan alat bantu (*tools*) sangat penting untuk mendukung aktivitas analisis, perancangan, hingga simulasi proses pengembangan perangkat lunak.

Salah satu alat bantu yang digunakan dalam penerapan Rational Unified Process (RUP) adalah **Enterprise Architect**, sebuah perangkat lunak pemodelan yang mendukung **Unified Modeling Language (UML)**. UML berfungsi sebagai bahasa pemodelan standar untuk merepresentasikan kebutuhan fungsional maupun non-fungsional sistem melalui berbagai diagram, seperti *use case*, *class*, *sequence*, dan *activity diagram*. Dalam penelitian terkait pengembangan **Detector Control System (DCS)** untuk eksperimen *High-Energy Physics (HEP)*, Enterprise Architect digunakan untuk menyusun *use case* baik dalam bentuk diagram maupun tabular, sehingga mendukung spesifikasi kebutuhan dan dokumentasi sistem secara sistematis [8].

Selain itu, penerapan Rational Unified Process (RUP) juga dapat didukung oleh perangkat lunak simulasi. Salah satu contoh yang digunakan dalam penelitian adalah **Simphony.NET**, sebuah aplikasi simulator yang dirancang untuk memodelkan proses pengembangan perangkat lunak. Melalui Simphony.NET, dapat dilakukan pemodelan berbagai sumber daya seperti analis bisnis, desainer, programmer, dan tester, serta penentuan probabilitas kesalahan dan jumlah iterasi proyek. Aplikasi ini juga memungkinkan simulasi pada berbagai skala proyek, baik kecil, menengah, maupun besar, sehingga memberikan gambaran yang lebih realistis tentang jalannya proses pengembangan perangkat lunak [9].

Dengan demikian, alat bantu seperti Enterprise Architect dan Symphony.NET berperan penting dalam mendukung penerapan RUP, baik dari sisi analisis dan pemodelan kebutuhan, maupun simulasi alur kerja dan sumber daya dalam proyek perangkat lunak.

BAB III

PENUTUP

A. Kesimpulan

Berdasarkan pembahasan, Rational Unified Process (RUP) merupakan kerangka proses pengembangan perangkat lunak yang iteratif, incremental, berbasis objek, serta menekankan dokumentasi dan arsitektur sebagai pusat pengembangan. Sejarahnya berawal dari *Objectory Process* hingga menjadi metodologi standar industri yang terintegrasi dengan UML. Penerapan RUP dilakukan melalui empat fase utama yaitu *inception*, *elaboration*, *construction*, dan *transition* yang memungkinkan identifikasi kebutuhan, pengendalian risiko, serta validasi sistem secara bertahap.

Rational Unified Process (RUP) memiliki karakteristik unik seperti pendekatan berbasis risiko, disiplin dan artefak yang terstruktur, serta dukungan dokumentasi yang kuat. Kelebihannya mencakup kontrol perubahan sistematis, dukungan UML, dan kualitas perangkat lunak yang terjamin, sementara kelemahannya terletak pada waktu dan biaya implementasi yang relatif lebih besar. Dibandingkan dengan model lain, RUP lebih unggul dalam hal struktur, dokumentasi, dan manajemen risiko, namun kalah fleksibel dan cepat dibandingkan Prototyping, AUP (Agile Unified Process), maupun RAD (Rapid Application Development).

Selain metodologi, keberhasilan penerapan Rational Unified Process (RUP) juga didukung oleh alat bantu seperti Enterprise Architect untuk pemodelan UML dan Symphony.NET untuk simulasi proses pengembangan. Dengan demikian, RUP dapat dipandang sebagai metodologi yang relevan dan komprehensif bagi proyek perangkat lunak berskala menengah hingga besar, terutama ketika kualitas, dokumentasi, dan pengelolaan risiko menjadi prioritas utama.

DAFTAR PUSTAKA

- [1] P. Kruchten, *The Rational Unified Process: An Introduction*. 2003. [Online]. Available:
[https://books.google.co.id/books?id=RYCMx6o47pMC&lpg=PR13&ots=h4frEWP9Sd&dq=rational unified process \(rup\)&lr&hl=id&pg=PR13#v=onepage&q=rational unified process \(rup\)&f=false](https://books.google.co.id/books?id=RYCMx6o47pMC&lpg=PR13&ots=h4frEWP9Sd&dq=rational+unified+process+(rup)&lr&hl=id&pg=PR13#v=onepage&q=rational+unified+process+(rup)&f=false)
- [2] A. Anwar, "A Review of RUP (Rational Unified Process)," no. 5, pp. 8–24, 2014.
- [3] S. Endang Anjarwani, "PENERAPAN METODE RATIONAL UNIFIED PROCESS (RUP) DALAM PENGEMBANGAN SISTEM INFORMASI MEDICAL CHECK UP PADA CITRA MEDICAL CENTRE (The Application of RationalUnifiedProcess (RUP) in Development of a Medical CheckUpInformation System at Citra Medical Centre)," *Jtika*, vol. 2, no. 1, pp. 76–88, 2020, [Online]. Available:
<http://jtika.if.unram.ac.id/index.php/JTIKA/>
- [4] S. Widayati and Y. I. Chandra, "Perancangan Aplikasi Rekomendasi Shade Complexion Make Up Untuk Menentukan Jenis Foundation Menggunakan Model Rational Unified Process (RUP) Berbasis Web," *J. Penerapan Sist. Inf. (Komputer Manajemen)*, vol. 5, no. 3, pp. 1355–1368, 2024, [Online]. Available:
<https://www.pkm.tunasbangsa.ac.id/index.php/kesatria/article/view/454/449>
- [5] F. Mubarak, H. Harliana, and I. Hadijah, "Perbandingan Antara Metode RUP dan Prototype Dalam Aplikasi Penerimaan Siswa Baru Berbasis Web," *Creat. Inf. Technol. J.*, vol. 2, no. 2, p. 114, 2015, doi: 10.24076/citec.2015v2i2.42.
- [6] C. Edeki, "Agile Unified Process," vol. 1, pp. 13–17, 2013.
- [7] C. Ramos, S. Ganesan, and R. Caytiles, "The Integration of MVC Framework in Rapid Application Development (RAD) Process Model," vol. 12, no. 1, pp. 57–66, 2018.
- [8] J. Carlos and C. Noris, "Analysis of Requirements for the Design of a Detector Control System in a High Energy Physics (HEP) Experiment," pp. 1–5.
- [9] T. K. Tia and W. A. K, "MODEL SIMULASI PENGEMBANGAN PERANGKAT LUNAK MENGGUNAKAN RATIONAL UNIFIED PROCESS (RUP)," vol. 2, pp. 33–40, 2018.