

Komparasi Metodologi Penentuan Kebutuhan Spesifikasi Sistem Dalam Pengembangan Sistem Informasi Akademik

Wahyu Wijaya Widiyanto¹⁾, Robi Wariyanto²⁾, Sri Wulandari³⁾, Fendy Prasetyo Nugroho⁴⁾,
Muqorobin⁵⁾

Program Magister Teknik Informatika, Universitas AMIKOM Yogyakarta
Jalan Ring Road Utara Condong Catur Depok-Sleman Telpn : 0274 – 884201
e-mail: wahyuwijaya8@gmail.com¹⁾, robiwariyanto@gmail.com²⁾, oel_and@yahoo.com³⁾,
vendyscenter@yahoo.com⁴⁾, cepat12@gmail.com⁵⁾

Abstrak

Pada saat dihadapkan pada pemilihan metodologi pengembangan sistem, banyak diantara pengembang proyek perangkat lunak yang akhirnya bingung. Padahal salah menentukan metodologi, dapat mempengaruhi penyusunan jadwal, staffing proyek, biaya dan lain-lain. Oleh karena itu, pemilihan metodologi merupakan bagian yang penting, tidak hanya pada saat pengembangan sistem informasi namun juga pada pengembangan proyek-proyek perangkat lunak lainnya. Dalam makalah ini membahas mengenai pengembangan sistem informasi akademik dengan 4 model/metode yaitu waterfall, Rapid Application Development (RAD), Prototype, dan Spiral dari segi kelebihan dan kelemahan, tujuan dari pembahasan ini agar pengembang dapat melakukan implementasi pengembangan sistem khususnya sistem informasi akademik dengan pemilihan metodologi yang tepat.

Kata kunci: komparasi, Rapid Application Development (RAD), waterfall, Prototype, Spiral

1. Pendahuluan

Latar Belakang

Perkembangan ilmu pengetahuan dan teknologi memberikan pengaruh yang besar dalam pengelolaan data, khususnya institusi pendidikan dalam mengelola data akademik. Dalam mengelola data akademik diperlukan suatu sistem untuk mengelola data agar lebih cepat dan akurat. Untuk pengembangan sistem informasi akademik diperlukan suatu analisa yang matang dan tepat dalam menentukan suatu kebutuhan spesifikasi kebutuhan sistem yang akan dikembangkan sehingga dapat meminimalkan resiko kegagalan pengembangan sistem. Oleh karena itu dalam makalah ini akan dibahas komparasi terhadap beberapa metodologi dalam menentukan kebutuhan sistem sehingga dapat diterapkan serta mendapat gambaran metodologi yang tepat dalam pengembangan sistem informasi akademik yang akan dikembangkan.

Analisis kebutuhan merupakan langkah awal untuk menentukan gambaran perangkat yang akan dihasilkan

ketika pengembang melaksanakan sebuah proyek pembuatan perangkat lunak. Perangkat lunak yang baik dan sesuai dengan kebutuhan pengguna sangat tergantung pada keberhasilan dalam melakukan analisis kebutuhan. Untuk proyek-proyek perangkat lunak yang besar, analisis kebutuhan dilaksanakan setelah aktivitas sistem *information engineering* dan *software project planning*.

Analisa kebutuhan yang baik belum tentu menghasilkan perangkat lunak yang baik, tetapi analisa kebutuhan yang tidak tepat menghasilkan perangkat yang tidak berguna. Mengetahui adanya kesalahan pada analisis kebutuhan pada tahap awal memang jauh lebih baik, tapi kesalahan analisis kebutuhan yang diketahui ketika sudah memasuki penulisan kode atau pengujian, bahkan hampir masuk dalam tahap penyelesaian merupakan malapetaka besar bagi pembuat perangkat lunak. Biaya dan waktu yang diperlukan akan menjadi sia sia.

Ada tiga faktor yang harus dipenuhi ketika melakukan analisis kebutuhan ini, yaitu lengkap, detail, dan benar. Lengkap artinya semua yang diharapkan oleh klien telah didapatkan oleh pihak yang melakukan analisis. Detail maksudnya adalah berhasil mengumpulkan informasi yang terperinci. Semua data dari analisis kebutuhan ini haruslah benar, sesuai apa yang dimaksud oleh klien, bukan benar menurut apa yang dipikirkan oleh pihak analisis.

Rumusan Masalah

Masalah yang muncul berdasarkan latar belakang diatas adalah sebagai berikut:

Apakah persamaan dari metode *waterfall*, *RAD*, *Prototype*, dan *Spiral*? Dan pertimbangan apa dalam penentuan metode tersebut?

Tujuan

Tujuan dari penelitian ini adalah untuk mengetahui persamaan dari metode *waterfall*, *RAD*, *Prototype*, dan *Spiral*, serta pertimbangan apa dalam penentuan metode tersebut dalam menentukan kebutuhan spesifikasi sistem informasi akademik.

Tinjauan Pustaka

Rekayasa Kebutuhan (*Requirement Engineering*) adalah bagian yang tak terpisahkan dari kegiatan rekayasa perangkat lunak. Rekayasa Kebutuhan mempunyai peran yang cukup penting, bahkan akan menentukan keberhasilan dari suatu proyek rekayasa perangkat lunak. Mengenai peran penting rekayasa kebutuhan tersebut telah banyak dikemukakan oleh para pakar.

- a. *Requirement Engineering* adalah proses menentukan properti tertentu dari sistem yang harus ada, dengan kata lain, menentukan komponen-komponen sistem. Kebutuhan proses menghasilkan informasi tentang desain yang akan menjadi dasar. Untuk itu, harus mengetahui dimana sebuah sistem akan digunakan, oleh siapa, dan layanan apa yang harus disediakan. Juga penting untuk menentukan kompromi apa yang dapat dilakukan jika terjadi konflik kebutuhan. Asumsi bahwa setiap sistem memiliki kumpulan fungsi yang berguna, yang penting untuk keberhasilan [1].
- b. *Software requirements* berisikan kebutuhan dan kendala yang ditempatkan pada produk perangkat lunak yang memberikan kontribusi pada solusi dari beberapa masalah dunia nyata [2].
- c. Rekayasa Kebutuhan membantu para ahli perangkat lunak untuk lebih memahami masalah dan menyelesaikannya. Ini meliputi kumpulan dari tugas-tugas yang mengarah ke pemahaman tentang apa yang akan menjadi dampak dari bisnis perangkat lunak, apa yang diinginkan oleh pelanggan, dan bagaimana pemakai akan berinteraksi dengan perangkat lunak [3].

Salah satu turunan dari SDLC adalah metode air terjun (*waterfall*). Metode air terjun atau yang sering disebut metode *waterfall* sering dinamakan siklus hidup klasik (*classic life cycle*), dimana hal ini menggambarkan pendekatan yang sistematis dan juga berurutan pada pengembangan perangkat lunak, dimulai dengan spesifikasi kebutuhan pengguna lalu berlanjut melalui tahapan-tahapan perencanaan (*planning*), permodelan (*modeling*), konstruksi (*construction*), serta penyerahan sistem ke para pelanggan/pengguna (*deployment*), yang diakhiri dengan dukungan pada perangkat lunak lengkap yang dihasilkan [3].

Rapid Application Development (RAD) merupakan proses pengembangan *software incremental*, yang menekankan pada siklus pengembangan yang sangat pendek. Untuk pengembangan suatu sistem informasi yang normal membutuhkan waktu minimal 180 hari, akan tetapi dengan menggunakan metode RAD suatu sistem dapat diselesaikan hanya dalam waktu 30-90 hari [4].

Menurut Ian Sommerville dalam bukunya yang berjudul *Software Engineering* 9th Edition, pengembangan perangkat lunak dengan model spiral adalah proses

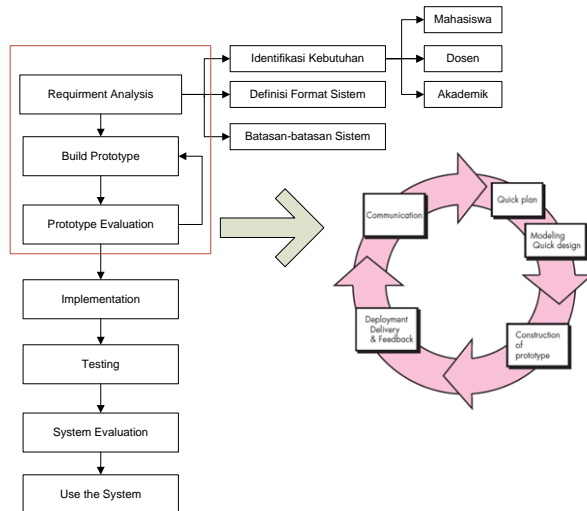
pengembangan perangkat lunak yang digambarkan sebagai sebuah spiral, dimana tiap putaran menggabungkan tahap yang berbeda dalam prosesnya. Perpindahan dari satu spiral ke putaran lain, berarti mengulang semua tahapan dari proses pengembangan perangkat lunak. Tiap putaran dalam spiral model mewakili sebuah fase dari proses pengembangan perangkat lunak. Dengan demikian, putaran paling dalam/ terdekat mungkin bersangkutan dengan kelayakan sistem, putaran selanjutnya dengan definisi kebutuhan, putaran selanjutnya dengan perancangan sistem, dan seterusnya.

Menurut Raymond McLeod, *prototype* didefinisikan sebagai alat yang memberikan ide bagi pembuat maupun pemakai potensial tentang cara sistem berfungsi dalam bentuk lengkapnya, dan proses untuk menghasilkan sebuah *prototype* disebut *prototyping*. *Prototyping* adalah proses pembuatan model sederhana *software* yang mengijinkan pengguna memiliki gambaran dasar tentang program serta melakukan pengujian awal. *Prototyping* memberikan fasilitas bagi pengembang dan pemakai untuk saling berinteraksi selama proses pembuatan, sehingga pengembang dapat dengan mudah memodelkan perangkat lunak yang akan dibuat. *Prototyping* merupakan salah satu metode pengembangan perangkat lunak yang banyak digunakan.

2. Pembahasan

Dalam pembahasan makalah ini mengambil 4 penelitian pengembangan sistem informasi akademik terdahulu menggunakan beberapa model sebagai komparasi, yaitu dengan metode *prototyping*, metode *waterfall*, metode RAD (*Rapid Application Development*), dan metode/model spiral.

Menurut penelitian Ahmad Homaidi dalam mengembangkan sistem informasi akademik menggunakan metode *prototyping*, karena harus menyesuaikan dengan keinginan pengguna yaitu dengan memberikan contoh dari tiap modul yang dibuat untuk ditunjukkan pada pengguna (akademik), jika sudah sesuai akan dilanjutkan pada tahap selanjutnya, jika tidak dilakukan perbaikan sesuai dengan permintaan pengguna [5]. Berikut gambaran metodologi penelitian dengan menggunakan metode *prototype*:



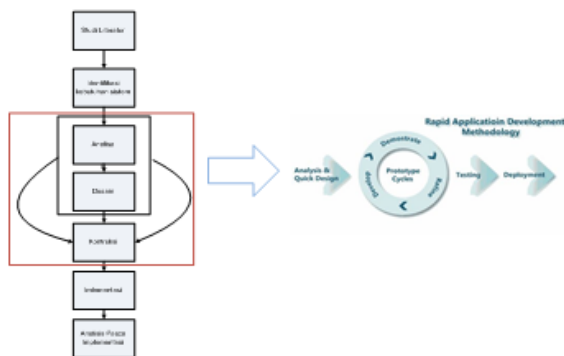
Gambar 1. Metode *prototype* penelitian Ahmad Homaidi

Berdasarkan metodologi yang dijabarkan di atas maka dapat dijabarkan jenis kebutuhan sistem informasi akademik bagi pengguna:

Tabel 1. Jenis Kebutuhan Sistem Informasi Akademik Bagi Pengguna

Mahasiswa	Dosen	Akademik
<ul style="list-style-type: none"> - Perubahan profil - KRS - KHS 	<ul style="list-style-type: none"> - Nilai - Distribusi mata kuliah 	<ul style="list-style-type: none"> - KRS - KHS - Nilai - Distribusi mata kuliah - Dosen - Mahasiswa - Absensi - Jadwal

Berdasarkan dari penelitian Pramana Yoga Saputra dalam pengembangan sistem informasi akademik metodologi yang digunakan dalam menentukan kebutuhan sistem yaitu menggunakan metodologi *Rapid Application Development (RAD)* [6]. Berikut gambaran metodologi penelitian yang digunakan dengan mengadopsi dari metode RAD:



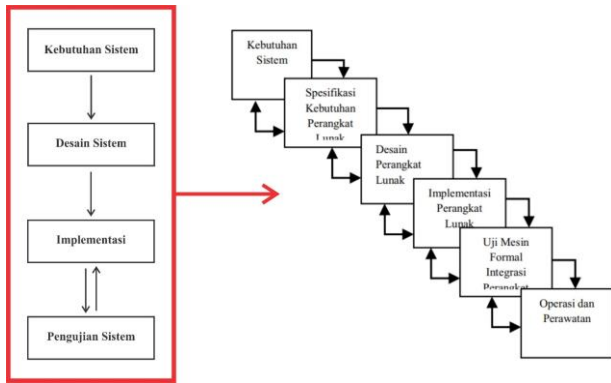
Gambar 2. Metode *prototype* penelitian Pramana Yoga Saputra

Sistem informasi akademik dengan metodologi RAD dibagi-bagi menjadi beberapa modul dan dikerjakan beberapa tim dalam waktu yang hampir bersamaan dalam waktu yang sudah ditentukan. Model ini melibatkan banyak tim, dan setiap tim mengerjakan tugas yang selevel, namun berbeda sesuai dengan pembagian modul sistem. Dengan menggunakan metode RAD sistem yang dikembangkan lebih cepat tersampaikan ke *user* dan tidak diperlukan dalam menunggu fitur yang lain terselesaikan.

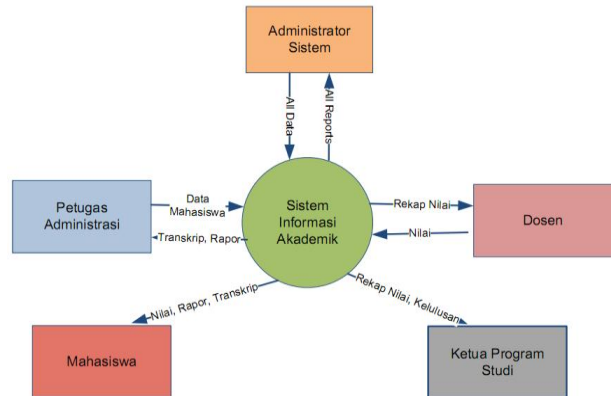
Tabel 2. Kelebihan dan Kelemahan Metode *prototype* penelitian Pramana Yoga Saputra

Kelebihan	Kelemahan
<p>a. Waktu dalam pengembangan sistem relatif singkat jika dibandingkan dengan pengembangan sendiri</p> <p>b. Pembiayaan pengembangan sistem lebih murah</p> <p>c. Mampu meminimalkan kesalahan-kesalahan dengan menggunakan <i>tool case</i></p> <p>d. Sistem mudah dimengerti dengan menggunakan <i>prototyping</i> dari sistem</p> <p>e. lebih fleksibel karena pengembang dapat melakukan proses desain ulang.</p> <p>f. Lebih fleksibel karena pengembang dapat melakukan proses desain ulang pada saat yang bersamaan</p>	<p>a. Dengan melakukan pembelian belum tentu bisa menghemat biaya dibandingkan dengan mengembangkan sendiri.</p> <p>b. Membutuhkan biaya tersendiri untuk membeli peralatan-peralatan penunjang seperti misalnya <i>software</i> dan <i>hardware</i>.</p> <p>c. Kesulitan melakukan pengukuran mengenai kemajuan proses.</p> <p>d. Kurang efisien karena apabila melakukan pengkodean dengan menggunakan tangan bisa lebih efisien.</p> <p>e. Ketelitian menjadi berkurang karena tidak menggunakan metode yang formal dalam melakukan pengkodean.</p> <p>f. Lebih banyak terjadi kesalahan apabila hanya mengutamakan kecepatan dibandingkan dengan biaya dan kualitas.</p> <p>g. Fasilitas-fasilitas banyak yang dikurangi karena terbatasnya waktu yang tersedia.</p>

Berdasarkan penelitian dari Sunarna Hendra Handaka dalam pengembangan sistem informasi akademik terpadu (SIKADU) dalam menentukan kebutuhan sistem dapat dirancang dengan sebuah metodologi *waterfall* [7]. Berikut gambaran metodologi penelitian yang digunakan dengan mengadopsi dari metode *waterfall*:



Gambar 3. Metodologi Yang Digunakan Dengan Mengadopsi Dari Metode Waterfall



Gambar 4. Model Keterhubungan Pengguna dan Data Pada Sistem Informasi Akademik

Tabel 3. Kelebihan dan Kelemahan penelitian dari Sunarna Hendra Handaka

Kelebihan	Kelemahan
a. Kualitas sistem akan baik karena dikerjakan secara bertahap. b. Pengembangan sistem sangat terorganisir karena setiap fase harus diselesaikan dengan lengkap sebelum melangkah ke fase berikutnya. c. Metode yang lebih baik dan mudah untuk digunakan jika kebutuhan sudah diketahui dengan baik.	a. Proses pengembangan tidak dapat dilakukan secara berulang sebelum terjadinya suatu produk. b. Kesalahan kecil akan menjadi masalah besar jika tidak diketahui sejak awal pengembangan yang berakibat pada tahapan selanjutnya. c. Pembuatan perangkat lunak akan dimulai ketika tahap desain sudah selesai.

Metode ini memungkinkan untuk departementalisasi dan kontrol. proses pengembangan model *fase one by one*, sehingga meminimalis kesalahan yang mungkin akan terjadi. Pengembangan bergerak dari konsep, yaitu melalui desain, implementasi, pengujian, instalasi, penyelesaian masalah, dan berakhir di operasi dan pemeliharaan.

Berdasarkan penelitian dari Humairi, Indri Rahmayuni, Defni dalam perancangan sistim informasi akademik berbasis web menggunakan model spiral (studi kasus AKNP Pelalawan), dalam melakukan desain sistem ini menggunakan model spiral karena Model spiral merupakan kombinasi antara model *iterative* dan model *linear* seperti *waterfall* yang dititik beratkan pada evaluasi analisis resiko. Model spiral terdiri atas empat tahapan. Dalam pembangunannya, sistem informasi atau perangkat lunak yang dibangun secara iterasi melewati empat tahapan tersebut [8].

Tabel 4. Kelebihan dan Kelemahan Model Spiral

Kelebihan	Kekurangan
a. Model ini dapat disesuaikan langsung terhadap perangkat lunak agar perangkat lunak dapat berjalan lancar pada komputer b. Lebih cocok untuk pengembangan sistem dan perangkat lunak skala besar terutama pada siacad sistem akademik. c. Pengembang dan pemakai dapat lebih mudah memahami dan bereaksi terhadap resiko setiap tingkat evolusi karena perangkat lunak terus bekerja selama proses d. Menggunakan prototipe sebagai mekanisme pengurangan resiko dan pada setiap keadaan di dalam evolusi produk e. Tetap mengikuti langkah-langkah dalam siklus kehidupan klasik dan memasukkannya ke dalam kerangka kerja iteratif f. Membutuhkan pertimbangan langsung terhadap resiko teknis sehingga mengurangi resiko sebelum menjadi permasalahan yang serius	a. Sulit untuk menyakinkan pelanggan bahwa pendekatan evolusioner ini bisa dikontrol b. Memerlukan penaksiran resiko yang masuk akal dan akan menjadi masalah yang serius jika resiko mayor tidak ditemukan dan diatur c. Butuh waktu lama untuk menerapkan paradigma ini menuju kepastian yang absolute

Seminar Nasional Sistem Informasi dan Teknologi Informasi 2018

SENSITEK 2018

STMIK Pontianak, 12 Juli 2018

Alasan menerapkan model spiral adalah Model spiral cocok digunakan untuk pengembangan perangkat lunak berikut:

- a. Ketika ada batasan biaya dan evaluasi serta analisis resiko merupakan salah satu factor penting
- b. Digunakan untuk proyek pengembangan dengan resiko menengah sampai besar
- c. Untuk proyek dengan jangka waktu panjang karena adanya potensi perubahan prioritas ekonomi yang sejalan dengan perubahan kebutuhan sistem
- d. Konsumen yang tidak yakin mengenai kebutuhan sistem yang akan dibuat
- e. Kebutuhan sistem cukup kompleks sehingga membutuhkan evaluasi untuk memperjelasnya

3. Kesimpulan

Berdasarkan hasil pembahasan diatas peneliti mengambil kesimpulan pada penelitian ini:

- a. Secara umum ada persamaan pada keempat metode yang dianalisa, yaitu tahapan dilakukan secara sekuensial; artinya satu tahapan selesai baru dilanjutkan dengan tahapan berikutnya.
- b. Metode-metode yang dianalisa mempunyai kelebihan dan kekurangannya masing-masing sehingga tidak dapat ditentukan mana yang lebih baik. Dari kelebihan dan kekurangan masing-masing metode, pengembang dapat memilih metode mana yang paling cocok untuk dirinya.

Saran dari peneliti:

- a. Sebaiknya ada penelitian lanjutan yang akan membandingkan metode-metode tersebut secara nyata, artinya membuat produk sistem informasi akademik menggunakan metode-metode tersebut dan membandingkannya untuk mencari mana yang lebih baik.
- b. Membangun metode pengembangan perangkat lunak sistem informasi akademik yang baru untuk mengakomodasi kekurangan-kekurangan metode yang dianalisa ini.

Daftar Pustaka

- [1]. A. Endres, H.D. Rombach, "A Handbook of Software and Systems Engineering : Empirical Observations, Laws and Theories", Pearson Education Limited, England, 2003.
- [2]. Kotonya, I. Sommerville, Requirements Engineering: Processes and Techniques, John Wiley & Sons, 2000.
- [3]. R.S. Pressman. "Software Engineering: A Practitioner's Approach", 6th Edition. McGraw-Hill. 2005.
- [4]. A. Noertjahyana, "Studi Analisis Rapid Application Development Sebagai Salah Satu Alternatif Metode Pengembangan Perangkat Lunak", *Jurnal Informatika Volume 3, No. 2, November 2002*.
- [5]. A.Homaidi, "Sistem Informasi Akademik Amik Ibrahimy Berbasis Web", *Jurnal Ilmiah Informatika Volume 1 No. 1 / Desember 2016*.
- [6]. P.Y. Saputra, D.O. Siahaan, "Analisis Dan Desain Sistem Informasi Akademik Politeknik Negeri Malang Menggunakan Metode Rapid Application Development (Rad)", *Prosiding*

Seminar Nasional Manajemen Teknologi XX Program Studi MMT-ITS, Surabaya 1 Februari 2014.

- [7]. S.H. Handaka, E. Kuntadi, "Pengembangan Sistem Perwalian Akademik Berbasis Web (Studi Kasus Pada Jurusan Manajemen Fakultas Ekonomi Universitas Negeri Semarang)", *Jurnal Teknologi Informasi dan Komunikasi, ISSN:2087-0868, Volume 6 Nomor 1 Maret 2015*.
- [8]. Humaira, I. Rahmayuni, D. National, "Perancangan Sistem Informasi Akademik Berbasis Web Menggunakan Model Spiral (Studi Kasus Aknp Pelalawan)", *Conference of Applied Sciences, Engineering, Business and Information Technology. Politeknik Negeri Padang, 15 – 16 Oktober 2016 ISSN:2541-111x*.