

Survey Paper: Perbandingan Metode Pengembangan Perangkat Lunak (Waterfall, Prototype, RAD)

Titania Pricillia¹, Zulfachmi²

^{1,2} STT Indonesia Tanjungpinang

Jln. Pompa Air No. 28 Tanjungpinang Kepulauan Riau Indonesia

¹titania@sttindonesia.ac.id

²zulfachmi@sttindonesia.ac.id

Intisari—*System Development Life Cycle* atau yang lebih dikenal dengan istilah SDLC adalah metodologi umum yang digunakan untuk mengembangkan sistem informasi. SDLC terdiri dari beberapa fase yang dimulai dari fase perencanaan, analisis, perancangan, implementasi hingga pemeliharaan sistem. Pembahasan yang dilakukan adalah terhadap metode pengembangan *waterfall*, *prototype* dan *rapid application development (RAD)*. Hasil perbandingan dari ketiga model tersebut menyatakan bahwa model *waterfall* lebih cocok untuk sistem atau perangkat lunak yang bersifat generik, sedangkan *prototype* dan *RAD* lebih cocok untuk sistem atau perangkat lunak yang bersifat *customize*. Sesuai dengan analisa yang didapatkan, ketiga metode memiliki kelebihan dan kekurangan masing-masing, sehingga pengembang dapat memilih metode mana yang lebih cocok untuk pengembangan perangkat lunaknya.

Kata kunci— SDLC, Waterfall, Prototype, RAD.

Abstract— *The System Development Life Cycle or better known as SDLC is a general methodology used to develop information systems. SDLC consists of several phases starting from the planning, analysis, design, implementation to system maintenance phases. The discussion is about the waterfall development method, prototype and rapid application development (RAD). The results of the comparison of the three models state that the waterfall model is more suitable for generic systems or software, while prototype and RAD are more suitable for customized systems or software. According to the analysis obtained, the three methods have their respective advantages and disadvantages, so that developers can choose which method is more suitable for software development.*

Keywords— SDLC, Waterfall, Prototype, RAD

I. PENDAHULUAN

SDLC (Systems Development Life Cycle, Siklus Hidup Pengembangan Sistem) atau Systems Life Cycle (Siklus Hidup Sistem), dalam rekayasa sistem dan rekayasa perangkat lunak, adalah proses pembuatan dan perubahan sistem serta model dan metodologi yang digunakan untuk mengembangkan sistem-sistem tersebut. Konsep ini umumnya merujuk pada sistem komputer atau informasi. SDLC juga merupakan pola yang diambil untuk mengembangkan sistem perangkat lunak, yang terdiri dari tahap-tahap: rencana (*planning*), analisis (*analysis*), desain (*design*), implementasi (*implementation*), uji coba (*testing*) dan pengelolaan (*maintenance*) [1].

Dalam rekayasa perangkat lunak, konsep SDLC mendasari berbagai jenis metodologi pengembangan perangkat lunak. Metodologi-metodologi ini membentuk suatu kerangka kerja untuk perencanaan dan pengendalian pembuatan sistem informasi, yaitu proses pengembangan perangkat lunak. Terdapat 3 jenis metode siklus hidup sistem yang paling banyak digunakan, yakni: siklus hidup sistem tradisional (*traditional system life cycle*), siklus hidup menggunakan prototyping (*life cycle using prototyping*), dan siklus hidup sistem orientasi objek (*object-oriented system life cycle*). [1] SDLC juga merupakan metodologi umum dalam pengembangan sistem

yang menandai kemajuan usaha analisis dan desain. Fase-fase yang terdapat didalam SDLC meliputi :

- a. Perencanaan Sistem (*Systems Planning*)
- b. Analisis Sistem (*Systems Analysis*)
- c. Perancangan Sistem (*Systems Design*)
- d. Implementasi Sistem (*Systems Implementation*)
- e. Pemeliharaan Sistem (*Systems Maintenance*)

Konsep SDLC mendasari model pengembangan perangkat lunak lainnya. Model pengembangan perangkat lunak tersebut antara lain waterfall, prototype, iterative, spiral, rapid application development (RAD) dan lainnya. Dalam penelitian ini hanya akan dibahas tiga model yaitu Waterfall, Prototyping, dan RAD.

II. METODE PENELITIAN

Dalam penelitian ini, penulis menggunakan data dari review beberapa jurnal dan di komparasi, dengan kata lain penulis menggunakan metode penelitian metode deskriptif. Metode deskriptif merupakan suatu metode penelitian dalam meneliti setatus dari sekelompok manusia, suatu obyek, suatu sistem pemikiran, suatu set kondisi, ataupun suatu kelas peristiwa pada masa saat ini. adapun tujuan dari penelitian deskriptif ini yaitu untuk membuat gambaran, deskripsi atau lukisan secara sistematis, faktual dan akurat mengenai fakta, sifat serta hubungan antar fenomena yang sedang diselidiki.

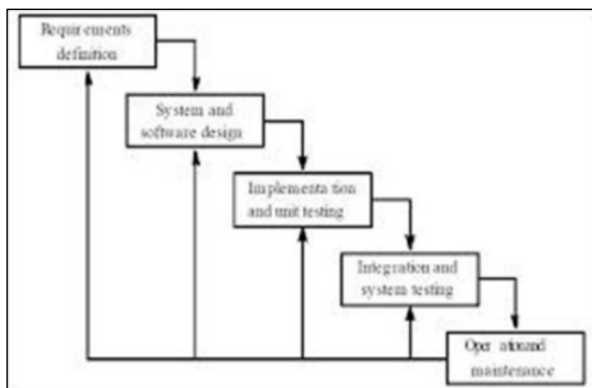
Penulis mengambil beberapa metode pengembangan sebagai objek penelitian yang akan dijadikan perbandingan antara lain :

- Waterfall
- Prototype
- RAD

III. HASIL DAN PEMBAHASAN

A. Waterfall

Model waterfall adalah model pengembangan perangkat lunak yang paling sering digunakan. Model pengembangan ini bersifat linear dari tahap awal pengembangan system yaitu tahap perencanaan sampai tahap akhir pengembangan system yaitu tahap pemeliharaan. Tahapan berikutnya tidak akan dilaksanakan sebelum tahapan sebelumnya selesai dilaksanakan dan tidak bisa kembali atau mengulang ke tahap sebelumnya. Menurut [2] “model SDLC air terjun (*waterfall*) sering disebut juga model sekuensial linier (*sequential linear*) atau alur hidup klasik (*classic life cycle*). Model air terjun menyediakan pendekatan alur hidup perangkat lunak secara sekuensial atau terurut dimulai dari analisis, desain, pengodean, pengujian, dan tahap pendukung (*support*)”. Berikut adalah gambar model air terjun:



Gambar 1. Metode *Waterfall*

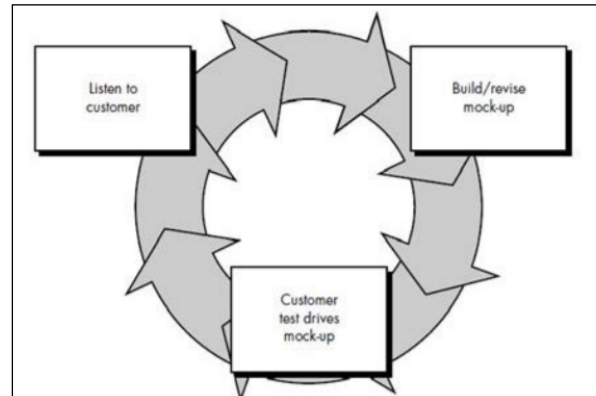
B. Prototype

Model prototyping merupakan suatu teknik untuk mengumpulkan informasi tertentu mengenai kebutuhan-kebutuhan informasi pengguna secara cepat. Berfokus pada penyajian dari aspek-aspek perangkat lunak tersebut yang akan nampak bagi pelanggan atau pemakai. Prototipe tersebut akan dievaluasi oleh pelanggan/pemakai dan dipakai untuk menyaring kebutuhan pengembangan perangkat lunak.

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[6]

Bentuk dari prototype pada gambar 2 merupakan versi awal dari tahapan sebuah sistem software yang digunakan dalam mempresentasikan gambaran dari ide, eksperimen dari sebuah rancangan, mencari sebanyak mungkin masalah yang ada serta penyelesaian terhadap masalah tersebut [3]. Sistem dengan model prototype memungkinkan pengguna agar mengetahui seperti apa tahapan sistem dibuat sehingga sistem mampu beroperasi dengan baik.

Metode prototype digunakan dimaksudkan untuk mendapatkan representasi dari pemodelan aplikasi yang akan dibuat. Rancangan aplikasi awal mulanya berbentuk mockup selanjutnya akan dievaluasi oleh pengguna. Setelah mockup dievaluasi pengguna tahap selanjutnya mockup menjadi bahan rujukan bagi pengembang software untuk merancang aplikasi.



Gambar 2. Model *Prototype*[4]

C. RAD

Rapid Application Development (RAD) adalah sebuah model proses perkembangan perangkat lunak sekuensial linier yang menekankan siklus perkembangan yang sangat pendek (kira-kira 60 sampai 90 hari). Model RAD ini merupakan sebuah adaptasi “kecepatan tinggi” dari model sekuensial linier dimana perkembangan cepat dicapai dengan menggunakan pendekatan konstruksi berbasis komponen.

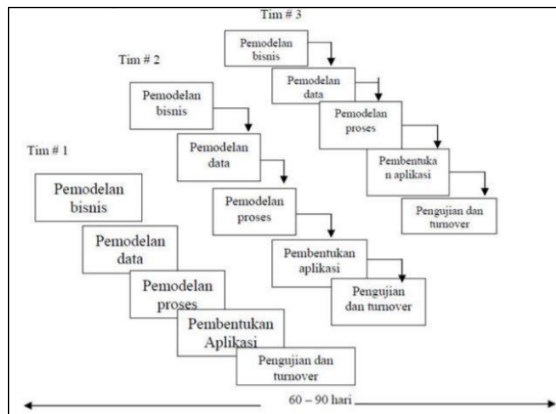
RAD adalah suatu pendekatan berorientasi objek terhadap pengembangan sistem yang mencakup suatu metode pengembangan serta perangkatperangkat lunak.[5] RAD bertujuan mempersingkat waktu yang biasanya diperlukan dalam siklus hidup pengembangan sistem tradisional antara perancangan dan penerapan suatu sistem informasi. Pada akhirnya, RAD sama-sama berusaha memenuhi syarat-syarat bisnis yang berubah secara cepat

RAD menggunakan metode iteratif (berulang) dalam mengembangkan sistem dimana working model (model bekerja) sistem dikonstruksikan di awal tahap pengembangan dengan tujuan menetapkan kebutuhan (requirement) pengguna dan selanjutnya disingkirkan. Dalam pengembangan sistem informasi normal, memerlukan waktu minimal 180 hari, namun dengan menggunakan metode RAD, sistem dapat diselesaikan dalam waktu 30-90 hari [10]. Model RAD memiliki 3 tahapan sebagai berikut:

- 1) Rencana Kebutuhan (*Requirement Planning*): User dan analyst melakukan pertemuan untuk mengidentifikasi tujuan dari sistem dan kebutuhan informasi untuk mencapai tujuan. Pada tahap ini merupakan hal terpenting yaitu adanya keterlibatan dari kedua belah pihak.
- 2) Proses Desain Sistem (*Design System*): Pada tahap ini keaktifan user yang terlibat menentukan untuk mencapai tujuan karena pada proses ini melakukan proses desain dan melakukan perbaikan-perbaikan apabila masih terdapat ketidaksesuaian desain antara user dan analyst. Seorang user dapat langsung memberikan komentar apabila terdapat ketidaksesuaian pada desain, merancang sistem dengan mengacu pada dokumentasi kebutuhan user yang dibuat pada tahap sebelumnya. Keluaran dari tahapan ini adalah

spesifikasi software yang meliputi organisasi sistem secara umum, struktur data dan yang lain.

- 3) Implementasi (*Implementation*): Tahapan ini adalah tahapan programmer yang mengembangkan desain suatu program yang telah disetujui oleh user dan analyst. Sebelum diaplikasikan pada suatu organisasi terlebih dahulu dilakukan proses pengujian terhadap program tersebut apakah ada kesalahan atau tidak. Pada tahap ini user biasa memberikan tanggapan akan sistem yang sudah dibuat serta mendapat persetujuan mengenai sistem tersebut.

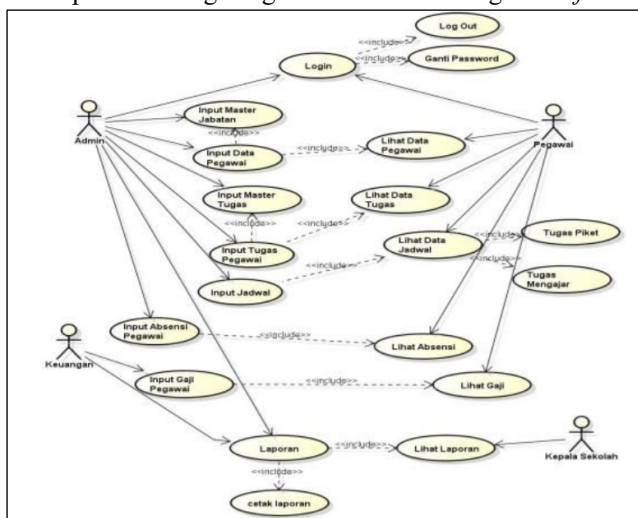


Gambar 3. Metode *Rapid Application Development* (RAD)

D. Penelitian Terdahulu

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

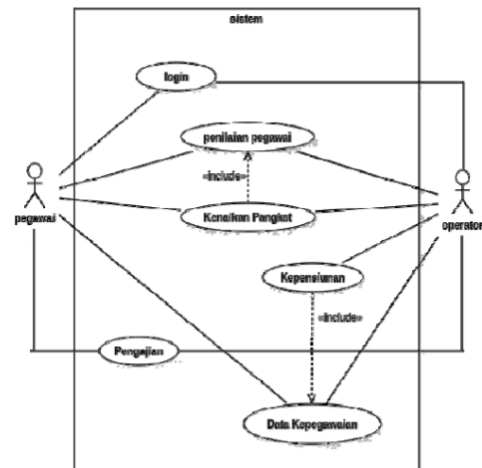
Menurut [7] oleh Nia Oktaviani yang berjudul “Sistem Informasi Pegawai Berbasis Web Dengan Metode Waterfall Pada SMA Aisyiyah 1 Palembang” dalam mengembangkan sistem informasi kepegawaian dalam menentukan kebutuhan sistem dapat dirancang dengan sebuah metodologi *waterfall*.



Gambar 4. Use Case Diagram Penelitian [7]

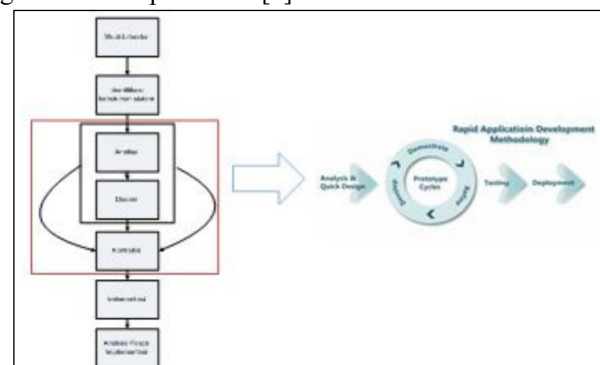
Menurut [8] oleh Mohamad Reza Fachlevi yang berjudul “Perancangan Sistem Informasi Kepegawaian Berbasis

Website Di Bagian Kepagawaian SDN Binakarya I Kabupaten Garut” menggunakan metode *prototype*, karena harus menyesuaikan dengan keinginan pengguna yaitu dengan memberikan contoh dari tiap modul yang dibuat untuk ditunjukkan pada pengguna (admin kepegawaian), jika sudah sesuai akan dilanjutkan pada tahap selanjutnya, jika tidak dilakukan perbaikan sesuai dengan permintaan pengguna.



Gambar 5. Use Case Diagram Penelitian [8]

Berdasarkan dari penelitian [9] oleh Rian Ardika, Ahmad Luthfi, dan Kurniawan Yang Berjudul “Sistem Informasi Data Pegawai Dengan Menggunakan Metode *Rapid Application Development* (Studi Kasus Kantor Kehutanan Prabumulih)”, metodologi yang digunakan dalam menentukan kebutuhan sistem yaitu menggunakan metodologi *Rapid Application Development* (RAD). Berikut gambaran metodologi yang digunakan oleh penelitian [9]



Gambar 6. Metode RAD Penelitian [9]

Sistem Informasi Data Pegawai 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.

E. Perbandingan

Bagian ini menjelaskan kelebihan dan kekurangan dari masing-masing metode pengembangan perangkat lunak

berdasarkan kajian dari penelitian terdahulu yang disajikan dalam bentuk tabel:

1) Waterfall

TABEL I
KELEBIHAN DAN KEKURANGAN METODE WATERFALL

No	Kelebihan Metode <i>Waterfall</i>	Kekurangan Metode <i>Waterfall</i>
1	Tahapan proses pengembangannya tetap (pasti), mudah diaplikasikan, dan prosesnya teratur	Proyek yang sebenarnya jarang mengikuti alur sekuensial seperti diusulkan, sehingga perubahan yang terjadi dapat menyebabkan hasil yang sudah didapatkan tim pengembang harus diubah kembali/iterasi sering menyebabkan masalah baru.
2	Cocok digunakan untuk produk software/program yang sudah jelas kebutuhannya di awal, sehingga minim kesalahannya.	Terjadinya pembagian proyek menjadi tahap-tahap yang tidak fleksibel, karena komitmen harus dilakukan pada tahap awal proses.
3	Software yang dikembangkan dengan metode ini biasanya menghasilkan kualitas yang baik.	Sulit untuk mengalami perubahan kebutuhan yang diinginkan oleh customer/pelanggan.
4	Dokumen pengembangan sistem sangat terorganisir, karena setiap fase harus terselesaikan dengan lengkap sebelum melangkah ke fase berikutnya	Pelanggan harus sabar untuk menanti produk selesai, karena dikerjakan tahap per tahap, dan proses pengerjaannya akan berlanjut ke setiap tahapan bila tahap sebelumnya sudah benar-benar selesai.
5		Perubahan ditengah-tengah pengerjaan produk akan membuat bingung tim pengembang yang sedang membuat produk
6		Adanya waktu kosong (menganggur) bagi pengembang, karena harus menunggu anggota tim proyek lainnya menuntaskan pekerjaannya

2) Prototype

TABEL II
KELEBIHAN DAN KEKURANGAN METODE PROTOTYPE

No	Kelebihan Metode <i>Prototype</i>	Kekurangan Metode <i>Prototype</i>
1	Pelanggan berpartisipasi aktif dalam pengembangan sistem, sehingga hasil produk pengembangan akan semakin mudah disesuaikan dengan keinginan dan kebutuhan pelanggan.	Proses analisis dan perancangan terlalu singkat.
2	Penentuan kebutuhan lebih mudah diwujudkan.	Biasanya kurang fleksibel dalam menghadapi perubahan.
3	Mempersingkat waktu pengembangan produk perangkat lunak.	Walaupun pemakai melihat berbagai perbaikan dari setiap versi prototype, tetapi pemakai mungkin tidak menyadari bahwa versi tersebut dibuat tanpa memperhatikan kualitas dan pemeliharaan jangka panjang.
4	Adanya komunikasi yang baik antara pengembang dan pelanggan.	Pengembang kadang-kadang membuat kompromi implementasi dengan menggunakan sistem operasi yang tidak relevan dan algoritma yang tidak efisien
5	Pengembang dapat bekerja lebih baik dalam menentukan kebutuhan pelanggan.	
6	Lebih menghemat waktu dalam pengembangan sistem.	
7	Penerapan menjadi lebih mudah karena pelanggan mengetahui apa yang diharapkannya	

3) RAD

TABEL III

KELEBIHAN DAN KEKURANGAN METODE RAPID APPLICATION DEVELOPMENT (RAD)

No	Kelebihan Metode <i>Rapid Application Development</i>	Kekurangan Metode <i>Rapid Application Development</i>
1	Lebih efektif dari Pengembangan Model waterfall/sequential linear dalam menghasilkan sistem yang memenuhi kebutuhan langsung dari pelanggan	Model RAD menuntut pengembangan dan pelanggan memiliki komitmen di dalam aktivitas rapid-fire yang diperlukan untuk melengkapi sebuah sistem, di dalam kerangka waktu yang sangat diperpendek. Jika komitmen tersebut tidak ada, proyek RAD akan gagal.
2	Cocok untuk proyek yang memerlukan waktu yang singkat.	Tidak semua aplikasi sesuai untuk RAD, bila system tidak dapat dimodulkan dengan teratur, pembangunan komponen penting pada RAD akan menjadi sangat bermasalah
3	Model RAD mengikuti tahap pengembangan sistem seperti pada umumnya, tetapi mempunyai kemampuan untuk menggunakan kembali komponen yang ada sehingga pengembang tidak perlu membuatnya dari awal lagi sehingga waktu pengembangan menjadi lebih singkat dan efisien	RAD tidak cocok digunakan untuk sistem yang mempunyai resiko teknik yang tinggi.
4		Membutuhkan tenaga kerja yang banyak untuk menyelesaikan sebuah proyek dalam skala besar
5		Jika ada perubahan di tengah-tengah pengerjaan maka harus membuat kontrak baru antara pengembang dan pelanggan

F. Hasil Perbandingan

Bagian ini melakukan penjabaran sesuai dengan tahapan pengembangan untuk evaluasi metode pengembangan perangkat lunak yang dipilih. Ini terdiri dari kriteria yang membahas tahapan pengembangan secara umum. Ini mencakup lima tahapan utama dari metode pengembangan: Perencanaan,

Analisis, Perancangan, Implementasi dan Pemeliharaan yang disajikan dalam bentuk tabel.

TABEL IV
PERBANDINGAN METODE PENGEMBANGAN PERANGKAT LUNAK

Tahapan Pengembangan Perangkat Lunak	<i>Waterfall</i>	<i>Prototype</i>	<i>RAD</i>
Perencanaan Sistem (<i>Systems Planning</i>)	Berawal dari kebutuhan	Berawal dari Kebutuhan	Berawal dari kebutuhan
Analisis Sistem (<i>Systems Analysis</i>)	Kebutuhan data harus dianalisis diawal secara lengkap dan menyeluruh	Kebutuhan data dapat ditambah ataupun dikurangi sesuai dengan kebutuhan user, ketika dilakukan testing.	Kebutuhan data dapat ditambah ataupun dikurangi sesuai dengan kebutuhan user, ketika dilakukan testing .
	Perubahan data ataupun fungsional akan merubah keseluruhan proses pada tahapan berikutnya.	Perubahan dapat dilakukan selama sistem atau perangkat lunak masih dalam bentuk prototype	Kebutuhan fungsi mayor dapat dimodulkan dalam waktu tertentu dan dapat dibicarakan oleh tim RAD yang terpisah
Perancangan Sistem (<i>Systems Design</i>)	Testing dilakukan ketika semua tahapan pada model sudah selesai.	Testing dapat dilakukan ketika prototype telah dibangun, sehingga hasil testing dapat merubah rancangan sistem.	Testing dapat dilakukan ketika prototype telah dibangun, sehingga hasil testing dapat merubah rancangan sistem.
	Tidak dapat memberikan gambaran yang jelas	Memberikan prototype sebagai gambaran sistem yang akan	Memberikan prototype sebagai gambaran sistem

	mengenai sistem yang dibangun, karena sistem bisa dilihat jika semua tahapan telah dilakukan.	dibangun, sehingga user dapat melihat dan berinteraksi langsung dengan gambaran sistem.	yang akan dibangun, sehingga user dapat melihat dan berinteraksi langsung dengan gambaran sistem.	telah dibangun	telah dibangun.	telah dibangun.
		User berperan aktif dalam pengembangan sistem	User berperan aktif dalam pengembangan sistem	Pemeliharaan Sistem (<i>Systems Maintenance</i>)	Dilakukan sesuai kesepakatan	Dilakukan sesuai kesepakatan
		Sistem yang dibangun akan sesuai dengan keinginan user	Sistem yang dibangun akan sesuai dengan keinginan user			
			Mempunyai kemampuan untuk menggunakan kembali komponen yang ada (<i>reusable object</i>) sehingga pengembangan tidak perlu membuat dari awal lagi dan waktu lebih singkat			
Implementasi Sistem (<i>Systems Implement</i>)	Menerapkan proses perancangan yang baik	Tidak menerapkan proses perancangan yang baik	Tidak menerapkan proses perancangan yang baik			
	Evaluasi dilakukan setelah sistem	Evaluasi dilakukan ketika prototype	Evaluasi dilakukan ketika prototype			

IV. KESIMPULAN

Hasil yang dicapai berdasarkan penelitian dan perbandingan yang telah dilakukan diatas, maka dapat disimpulkan:

- 1) Dapat diketahui karakteristik dari ketiga model pengembangan perangkat lunak Waterfall, Prototype dan RAD
- 2) Model pengembangan Waterfall cocok digunakan untuk sistem atau perangkat lunak yang bersifat generik, artinya sistem dapat diidentifikasi semua kebutuhannya dari awal dengan spesifikasi yang umum serta sesuai untuk perangkat lunak yang memiliki tujuan untuk membangun sebuah sistem dari awal yang mengumpulkan kebutuhan sistem yang akan dibangun sesuai dengan topik penelitian yang dipilih sampai dengan produk tersebut diuji
- 3) Model pengembangan Prototype lebih cocok untuk sistem atau perangkat lunak yang bersifat customize, artinya software yang diciptakan berdasarkan permintaan dan kebutuhan (bahkan situasi atau kondisi) tertentu dan sesuai untuk perangkat lunak memiliki tujuan untuk mengimplementasikan sebuah metode atau algoritma tertentu pada suatu kasus.
- 4) Model pengembangan RAD lebih cocok untuk sistem atau perangkat lunak yang bersifat customize, berskala besar dan memerlukan waktu yang lebih singkat artinya software yang diciptakan berdasarkan permintaan dan kebutuhan (bahkan situasi atau kondisi) tertentu dan sesuai untuk perangkat lunak memiliki tujuan untuk mengimplementasikan sebuah metode atau algoritma tertentu pada suatu kasus, serta memiliki kemungkinan untuk kebutuhan pengembangan kembali dalam jangka waktu yang cukup panjang.
- 5) 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.

REFERENSI

- [1] Britton, Carol, Jill Doake, "Object Oriented Systems Development". McGraw-Hill, 2001, ISBN 0-07- 709544-8.
- [2] Rosa, A. S. dan M Shalahuddin. "Rekayasa Perangkat Lunak Terstruktur dan Berorientasi Objek", 2013, Bandung, Informatika
- [3] I. Sommerville, "Software Engineering, 9th ed", 2007, Boston: Addison-Wesley.
- [4] M. Khosrow-Pour, "Encyclopedia of Information Science and

- [5] Technology. Hershey", PA, 2005, USA: IGI Global Publishing
Kendall, "Analisis dan Perancangan Sistem", 2010, Jakarta , PT Indeks.
- [6] McLeod, Jr., Raymond, " Sistem Informasi Managemen (Terjemahan) , 2011, Jakarta: Salemba Empat.
- [7] Nia Oktaviani, "Sistem Informasi Pegawai Berbasis Web Dengan Metode Waterfall Pada Sma Aisyiyah 1 Palembang", Seminar Nasional Inovasi Teknologi, ISSN: 2549-7952 UN PGRI Kediri, 2017, Kediri
- [8] Mohamad Reza Fachlevi, "Perancangan Sistem Informasi Kepegawaian Berbasis Website Di Bagian Kepagawaian Sdn Binakarya I Kabupaten Garut", Jurnal SIMETRIS, Vol 8 No 2, 2017, ISSN: 2252- 4983.
- [9] Rian Ardika, Ahmad Luthfi, dan Kurniawan, "Sistem Informasi Data Pegawai Dengan Menggunakan Metode Rapid Application Development (Studi Kasus Kantor Kehutanan Prabumulih)", 2015, Student Colloquium Sistem Informasi & Teknik Informatika (SC-SITI) Palembang
- [10] Wahyuningrum, dkk, "Perancangan WEB e-Commerce dengan Metode Rapid Application Development (RAD) untuk Produk Unggulan Desa", Seminar Teknologi Informasi dan Komunikasi Terapan (Semantik), 2014, Semarang