

## PERBANDINGAN METODOLOGI WATERFALL DAN RAD (RAPID APPLICATION DEVELOPMENT) DALAM PENGEMBANGAN SISTEM INFORMASI

**Deni Murdiani<sup>1</sup>, Muhamad Sobirin<sup>2</sup>**

Program Studi Teknik Informasi, universitas saintek muhammadiyah  
 Email : [denimurdiani@stmikmj.ac.id](mailto:denimurdiani@stmikmj.ac.id)<sup>1</sup>, [muhamadsobirin26@gmail.com](mailto:muhamadsobirin26@gmail.com)<sup>2</sup>

### ABSTRAK

SDLC atau *Software Development Life Cycle* atau sering disebut juga *System Development Life Cycle* adalah proses mengembangkan atau mengubah suatu sistem perangkat lunak dengan menggunakan model-model dan metodologi yang digunakan orang untuk mengembangkan sistem-sistem perangkat lunak sebelumnya (berdasarkan *best practice* atau cara-cara yang sudah teruji baik). Jurnal ini berusaha memaparkan analisa metodologi pengembangan perangkat lunak antara model *waterfall* (air terjun), dan model RAD (*Rapid Application Development*). Dari jurnal ini diharapkan dapat memberikan pertimbangan dalam pemilihan metodologi yang tepat berdasarkan kebutuhan, kelebihan dan kekurangan, juga faktor pertimbangan lainnya

**Kata Kunci:** *Metodologi pengembangan perangkat lunak, Waterfall, RAD*

### I. PENDAHULUAN

Berkembangnya kebutuhan perangkat lunak membuat perancangan sebuah sistem menjadi sebuah kebutuhan pula, terutama untuk sistem yang besar, sulit dan memiliki fitur kompleks. Berbagai metodologi pun mulai digunakan untuk memudahkan dalam pengembangan sistemnya (Nukman et al., 2014).

Terdapat beberapa metode yang dapat kita gunakan dalam pengembangan sistem diantaranya Waterfall dan RAD (Rapid Application Development). Masing-masing

metode memiliki kelebihan dan kekurangan, sehingga terkadang membutuhkan studi khusus agar dapat menentukan metode mana yang paling tepat untuk digunakan dalam mengembangkan sebuah sistem informasi (Nukman et al., 2014).

Terdapat beberapa aspek yang dapat dijadikan acuan untuk penentuan metode pengembangan ini, misalnya ukuran sistem, jumlah tim, resiko pengembangan, dana, waktu dan aspek lainnya (Nukman et al., 2014).

Pemilihan metode yang tepat tentunya dapat menguntungkan bagi

programmer, terutama untuk penghematan cost, waktu, dan SDM (Nukman et al., 2014).

Dalam penulisan ini, penulis mencoba untuk membandingkan dua metode pengembangan antara metode Waterfall dan metode RAD (Rapid Application Development).

## II. METODOLOGI PENELITIAN

Metode penelitian pada dasarnya merupakan cara ilmiah untuk mendapatkan data dengan tujuan dan kegunaan tertentu. Berdasarkan hal tersebut terdapat empat kata kunci yang perlu diperhatikan yaitu cara ilmiah, data, tujuan dan kegunaan (Sugiyono, 2010) pada jurnal (Widiyanto, 2018)

Dalam penelitian ini, penulis menggunakan data dari penelitian terdahulu yang dikomparasi dalam arti lain penulis menggunakan metode deskriptif. Metode deskriptif yaitu suatu metode penelitian dalam meneliti status dari sekelompok manusia, suatu obyek, suatu sistem pemikiran, suatu set kondisi, ataupun suatu kelas peristiwa pada masa saat ini (Widiyanto, 2018).

Penelitian deskriptif bertujuan untuk membuat deskripsi secara sistematis, faktual dan akurat mengenai fakta-fakta dan sifat populasi di wilayah tertentu. Data kualitatif yang dihasilkan akan dapat memberikan jawaban terhadap penelitian yang dilakukan (Wahid, 2020).

Langkah awal pengumpulan data menggunakan studi pustaka. Studi pustaka merupakan metode pengumpulan data yang diarahkan kepada pencarian data dan informasi

melalui dokumen baik tertulis, foto, gambar, maupun dokumen elektronik yang mendukung dalam proses penulisan (Wahid, 2020).

Penulis mengambil beberapa metode pengembangan sistem informasi sebagai objek penelitian dan dijadikan perbandingan, yaitu :

- a. Metode Waterfall
- b. Metode RAD

## III. HASIL DAN PEMBAHASAN

### SDLC (*Software Development Life Cycle*)

Pada awal pengembangan perangkat lunak, para *progammer* langsung melakukan pengodean perangkat lunak tanpa menggunakan prosedur atau tahapan pengembangan perangkat lunak. Dan ditemuilah kendala-kendala seiring perkembangan skala sistem-sistem perangkat yang semakin besar (AS & Shalahuddin, 2015).

SDLC dimulai dari tahun 1960-an, untuk mengembangkan sistem skala usaha besar secara fungsioanal untuk para konglomerat pada jaman itu. Sistem-sistem yang dibangun mengelola informasi kegiatan dan rutinitas dari perusahaan-perusahaan yang berpotensi memiliki data yang besar dalam perkembangannya (AS & Shalahuddin, 2015).

SDLC atau *Software Development Life Cycle* atau sering disebut juga *System Development Life Cycle* adalah proses mengembangkan atau mengubah suatu sistem perangkat lunak dengan menggunakan model-model dan metodologi yang digunakan orang

untuk mengembangkan sistem-sistem perangkat lunak sebelumnya (berdasarkan *best practice* atau cara-cara yang sudah teruji baik) (AS & Shalahuddin, 2015).

System Development Life Cycle (SDLC) adalah sebuah proses logika yang digunakan oleh seorang system analyst untuk mengembangkan sebuah sistem informasi yang melibatkan requirements, validation, training, dan pemilik system (Wahid, 2020).

SDLC (*System Development Life Cycle*) adalah salah satu metode pengembangan sistem informasi yang populer pada saat sistem informasi pertama kali dibuat (Susanto, 2004).

Tahapan-tahapan yang ada pada SDLC secara global adalah sebagai berikut (AS & Shalahuddin, 2015):

- Inisiasi (*initiation*) tahap ini biasanya ditandai dengan pembuatan proposal proyek perangkat lunak.
- Pengembangan konsep sistem (*system concept development*) mendefinisikan lingkup konsep termasuk dokumen lingkup sistem, analisis manfaat biaya, manajemen rencana, dan pembelajaran kemudahan sistem.
- Perencanaan (*Planning*) mengembangkan rencana manajemen proyek dan dokumen perencanaan lainnya. Menyediakan dasar untuk mendapatkan sumber daya (*resources*) yang dibutuhkan untuk memperoleh solusi.
- Analisis kebutuhan (*requirements analysis*)

Menganalisa kebutuhan pemakai sistem perangkat lunak (*user*) dan mengembangkan kebutuhan *user*. Membuat dokumen kebutuhan fungsional.

- Desain (*design*) mentransformasikan kebutuhan detail menjadi kebutuhan yang sudah lengkap, dokumen desain sistem fokus pada bagaimana dapat memenuhi fungsi-fungsi yang dibutuhkan.
- Pengembangan (*development*) mengonversi desain ke sistem informasi yang lengkap termasuk bagaimana memperoleh dan melakukan instalasi lingkungan sistem yang dibutuhkan; membuat basis data dan mempersiapkan prosedur kasus pengujian; mempersiapkan berkas atau *file* pengujian, pengodean, pengompilasian, memperbaiki dan membersihkan program; peninjauan pengujian.
- Integrasi dan pengujian (*integration and test*) mendemonstrasikan sistem perangkat lunak bahwa telah memenuhi kebutuhan yang dispesifikasikan pada dokumen kebutuhan fungsional. Dengan diarahkan oleh staf penjamin kualitas (*quality assurance*) dan *user*. Menghasilkan laporan analisis pengujian.
- Implementasi (*implementation*) termasuk pada persiapan implementasi, implementasi perangkat lunak pada lingkungan produksi (lingkungan pada *user*) dan menjalankan resolusi dari permasalahan yang teridentifikasi dari fase integrasi dan pengujian.

- Operasi dan pemeliharaan (*operations and maintenance*) mendeskripsikan pekerjaan untuk mengoperasikan dan memelihara sistem informasi pada lingkungan produksi (lingkungan pada *user*), termasuk implementasi akhir dan masuk pada proses peninjauan.
- Disposisi (*disposition*) mendeskripsikan aktifitas akhir dari pengembangan sistem dan membangun data yang sebenarnya sesuai dengan aktifitas *user*.

### Model SDLC

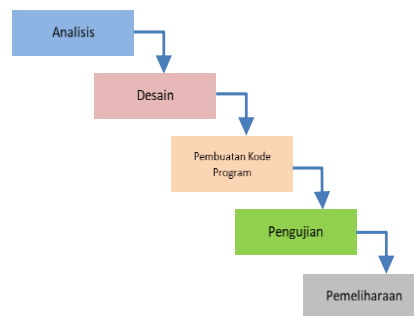
#### 1) Model Waterfall

Nama lain dari model ini adalah “*Linear Sequential Model*” dimana hal ini menggambarkan pendekatan yang sistematis dan juga berurutan pada pengembangan perangkat lunak, mulai dengan spesifikasi kebutuhan pengguna lalu berlanjut melalui tahapan-tahapan perencanaan (*planning*), permodelan (*modelling*), konstruksi (*contruction*), serta penyerahan sistem ke para pengguna (*deployment*), yang diakhiri dengan dukungan pada perangkat lunak lengkap yang dihasilkan (Pressman, S, 2012) pada jurnal (Wahid, 2020).

Model waterfall pertama kali diperkenalkan oleh Winston Royce sekitar tahun 1970 sehingga sering dianggap kuno, tetapi merupakan model yang paling banyak dipakai didalam Software Engineering (SE). saat ini model waterfall merupakan model pengembangan perangkat lunak yang sering digunakan. Model pengembangan ini melakukan pendekatan secara sistematis dan

berurutan. Disebut waterfall karena tahap demi tahap yang dilalui harus menunggu selesainya tahap sebelumnya dan berjalan berurutan. Model pengembangan ini bersifat linear dari tahap awal pengembangan sistem yaitu tahap perencanaan sampai tahap akhir pengembangan sistem yaitu tahap pemeliharaan. Tahapan berikutnya tidak akan dilaksanakan sebelum tahapan sebelumnya selesai dilaksanakan dan tidak bisa kembali atau mengulang ke tahap sebelumnya (Wahid, 2020).

Model SDLC air terjun (*waterfall*) sering juga disebut model sekuensial linier (*sequential linier*) 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 (AS & Shalahuddin, 2015) :



Gambar. 1. Model Waterfall

- Analisis kebutuhan perangkat lunak proses pengumpulan kebutuhan dilakukan secara intensif untuk mespesifikasikan kebutuhan perangkat lunak agar dapat dipahami perangkat lunak seperti apa yang dibutuhkan oleh

*user*. Spesifikasi kebutuhan perangkat lunak pada tahap ini perlu untuk didokumentasikan.

- **Desain**

desain perangkat lunak adalah proses multi langkah yang fokus pada desain pembuatan program perangkat lunak termasuk struktur data, arsitektur perangkat lunak, representasi antarmuka, dan prosedur pengodean.

- **Pembuatan kode program**

desain harus ditranslasikan ke dalam program perangkat lunak. Hasil dari tahap ini adalah program komputer sesuai dengan desain yang telah dibuat pada tahap desain.

- **Pengujian**

pengujian fokus pada perangkat lunak secara dari segi logik dan fungsional dan memastikan bahwa semua bagian sudah diuji. Hal ini dilakukan untuk meminimalisir kesalahan (*error*) dan memastikan keluaran yang dihasilkan sesuai dengan yang diinginkan.

- **Pendukung (*support*) atau pemeliharaan (*maintenance*)**

tidak menutup kemungkinan sebuah perangkat lunak mengalami perubahan ketika sudah dikirimkan ke *user*. Perubahan bisa terjadi karena adanya kesalahan yang muncul dan tidak terdeteksi saat pengujian atau perangkat lunak harus beradaptasi dengan lingkungan baru.

## **2) Model *Rapid Application Development RAD***

*Rapid Application Development* (RAD) adalah suatu pendekatan berorientasi objek terhadap pengembangan sistem yang mencakup suatu metode

pengembangan serta perangkat lunak (Kendall, 2010) pada jurnal (Pricillia & Zulfachmi, 2021).

*Rapid Application Development* adalah model proses pengembangan perangkat lunak yang bersifat inkremental terutama untuk waktu pengerjaan yang pendek. Model RAD adalah adaptasi dari model air terjun versi kecepatan tinggi dengan menggunakan model air terjun untuk pengembangan setiap komponen perangkat lunak (AS & Shalahuddin, 2015).

Tahapan RAD terdiri dari 3 tahap yang terstruktur dan saling bergantung disetiap tahap, yaitu (Putri & Effendi, 2018) :

### 1) **Requirements Planning** (Perencanaan Persyaratan).

- Pengguna dan analisis bertemu untuk mengidentifikasi tujuan dari aplikasi atau sistem
- Berorientasi pada pemecahan masalah bisnis.

### 2) **Design Workshop.**

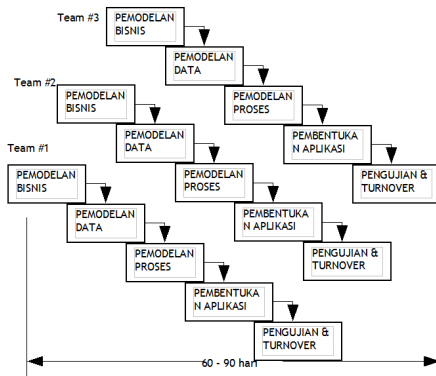
- Fase desain dan menyempurnakan.
- Gunakan kelompok pendukung keputusan sistem untuk membantu pengguna setuju pada desain.
- Programmer dan analis membangun dan menunjukkan tampilan visual desain dan alur kerja pengguna.
- Pengguna menanggapi prototipe kerja aktual.
- Analisis menyempurnakan modul dirancang berdasarkan tanggapan pengguna.

### 3) Implementation (Penerapan).

- Sebagai sistem yang baru dibangun, sistem baru atau parsial diuji dan diperkenalkan kepada organisasi.
- Ketika membuat sistem baru, tidak perlu untuk menjalankan sistem yang lama secara parallel.

Model RAD membagi tim pengembang menjadi beberapa tim untuk mengerjakan beberapa komponen masing-masing tim pengerjaan dapat dilakukan secara paralel.

Berikut adalah gambar dari model RAD (AS & Shalahuddin, 2015) :



Gambar. 2. Model Rapid Application Development (RAD)

- **Pemodelan bisnis**  
pemodelan yang dilakukan untuk memodelkan fungsi bisnis untuk mengetahui informasi apa yang terkait proses bisnis, informasi apa saja yang harus dibuat, siapa yang harus membuat informasi itu, bagaimana alur informasi itu, proses apa saja yang terkait informasi itu.
- **Pemodelan data**  
memodelkan data apa saja yang dibutuhkan berdasarkan pemodelan bisnis dan mendefinisikan atribut-atributnya beserta relasinya dengan data-data yang lain.
- **Pemodelan proses**

mengimplementasikan fungsi bisnis yang sudah didefinisikan terkait dengan pendefinisian data.

- **Pembuatan aplikasi**  
mengimplementasikan pemodelan proses dan data menjadi program. Model RAD sangat menganjurkan pemakaian komponen yang sudah ada jika dimungkinkan.

- **Pengujian dan pergantian**  
Menguji komponen-komponen yang dibuat. Jika sudah teruji maka tim pengembang komponen dapat beranjak untuk mengembangkan komponen berikutnya.

### Perbandingan

**Kelebihan dan Kekurangan Waterfall** (Pricillia & Zulfachmi, 2021) :

Tabel 1  
Kelebihan Kekurangan Metode Waterfall

No.	Kelebihan	Kekurangan
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 tahaptahap 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 benarbenar selesai.
5		Perubahan ditengahengah pengerjaan produk akan membuat bingung tim pengembang yang sedang membuat produk

6		Adanya waktu kosong (mengganggu) bagi pengembang, karena harus menunggu anggota tim proyek lainnya menuntaskan pekerjaannya
---	--	---

		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	RAD tidak cocok digunakan untuk sistem yang mempunyai resiko teknik yang tinggi.

**Kelebihan dan Kekurangan *Rapid Application Development (RAD)***

(Pricillia & Zulfachmi, 2021) :

Tabel 2

Kelebihan Kekurangan Metode RAD

N o.	Kelebihan	Kekurangan
1	Lebih efektif dari Pengembangan Model waterfall/sequentia l 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.



	kembali komponen yang ada sehingga pengembang tidak perlu membuatnya dari awal lagi sehingga waktu pengembangan menjadi lebih singkat dan efisien	
4		Mebutuhkan 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

1) Model *Waterfall* cenderung digunakan pada sistem informasi dimana sistem tersebut dapat diidentifikasi semua kebutuhannya dari awal, dengan spesifikasi umum dan sesuai untuk perangkat lunak yang memiliki tujuan untuk membangun sebuah sistem dari awal, mengumpulkan kebutuhan dari sebuah sistem yang akan dibuat sesuai dengan penelitian yang dipilih sampai produk tersebut diuji

2) Model *Rapid Application Development (RAD)* dapat digunakan untuk sistem atau perangkat lunak yang berskala besar dan memerlukan waktu lebih singkat, dimana software dibuat berdasarkan permintaan dan kebutuhan tertentu dan sesuai untuk perangkat lunak yang memiliki tujuan untuk menerapkan sebuah metode tertentu pada suatu kasus, juga adanya kemungkinan untuk kebutuhan pengembangan selanjutnya dalam jangka waktu yang cukup panjang.

3) Metode-metode yang dianalisa diatas memiliki kelebihan dan kekurangan masing-masing, disini penulis tidak dapat menentukan mana yang lebih baik. Dari kelebihan dan kekurangan masing-masing metode, diharapkan kita dapat memilih metode mana yang paling cocok untuk digunakan.

**IV. KESIMPULAN**

Berdasarkan penelitian dan perbandingan yang telah dilakukan, penulis mendapat kesimpulan sebagai berikut:

**V. DAFTAR PUSTAKA**

AS, R., & Shalahuddin, M. (2015). *Rekayasa Perangkat Lunak* (3rd ed.). Informatika Bandung.

- Kendall. (2010). *Analisis dan Perancangan Sistem*. PT Index.
- Nukman, H., Aceh, B., & Informasi, M. T. (2014). *Perbandingan Metodology Klasik Dan Agile Dalam Isbn : 978-602-70467-0-2*.
- Pressman, S, R. (2012). *Rekayasa Perangkat Lunak*. ANDI.
- Pricillia, T., & Zulfachmi. (2021). Perbandingan Metode Pengembangan Perangkat Lunak (Waterfall, Prototype, RAD). *Jurnal Bangkit Indonesia*, 10(1), 6–12.  
<https://doi.org/10.52771/bangkitindonesia.v10i1.153>
- Putri, M. P., & Effendi, H. (2018). *Implementasi Metode Rapid Application Development Pada Website Service Guide “Waterfall Tour South Sumatera .”* 07(September), 130–136.
- Susanto, A. (2004). *Sistem Informasi Manajemen*. Lingga Jaya.
- Wahid, A. A. (2020). Analisis Metode Waterfall Untuk Pengembangan Sistem Informasi. *Jurnal Ilmu-Ilmu Informatika Dan Manajemen STMIK*, November, 1–5.  
[https://www.researchgate.net/profile/Aceng\\_Wahid/publication/346397070\\_Analisis\\_Metode\\_Waterfall\\_Untuk\\_Pengembangan\\_Sistem\\_Informasi/links/5fbfa91092851c933f5d76b6/Analisis-Metode-Waterfall-Untuk-Pengembangan-Sistem-Informasi.pdf](https://www.researchgate.net/profile/Aceng_Wahid/publication/346397070_Analisis_Metode_Waterfall_Untuk_Pengembangan_Sistem_Informasi/links/5fbfa91092851c933f5d76b6/Analisis-Metode-Waterfall-Untuk-Pengembangan-Sistem-Informasi.pdf)
- Widiyanto, W. W. (2018). Analisa Metodologi Pengembangan Sistem Dengan Perbandingan Model Perangkat Lunak Sistem Informasi Kepegawaian Menggunakan Waterfall Development Model, Model Prototype, Dan Model Rapid Application Development (Rad). *Jurnal Informa Politeknik Indonusa Surakarta ISSN*, 4(1), 34–40.  
<http://www.informa.poltekindonusa.ac.id/index.php/informa/article/view/34>