

STUDI ANALISIS *RAPID APPLICATION DEVELOPMENT* SEBAGAI SALAH SATU ALTERNATIF METODE PENGEMBANGAN PERANGKAT LUNAK

Agustinus Noertjahyana

Fakultas Teknologi Industri, Jurusan Teknik Informatika, Universitas Kristen Petra

e-mail : agust@peter.petra.ac.id

ABSTRAK: *Rapid Application Development (RAD)* sebagai salah satu alternatif dari *System Development Life Cycle* belakangan ini seringkali digunakan untuk mengatasi keterlambatan yang terjadi apabila menggunakan metode konvensional. Adapun keunggulan yang bisa didapatkan dengan menggunakan metode ini adalah kecepatan, ketepatan, dan biaya yang relatif lebih rendah dibanding dengan metode konvensional. Di samping itu dengan melibatkan user pada proses desain menyebabkan kebutuhan user dapat terpenuhi dengan baik dan secara otomatis kepuasan user sebagai pengguna sistem semakin meningkat. Akan tetapi di dalam menggunakan metode *Rapid Application Development* perlu untuk memperhatikan hal-hal yang penting, terutama kesiapan tim, ruang lingkup sistem, kebutuhan user, dan kinerja sistem. Pada akhirnya, sebagai salah satu alternatif dari *System Development Life Cycle*, maka *Rapid Application Development* dapat dijadikan acuan untuk menghasilkan sistem informasi yang dapat memenuhi kebutuhan user.

Kata kunci: *Rapid Application Development (RAD)*, *System Development life Cycle (SDLC)*.

ABSTRACT: *Rapid Application Development is one of the alternatives of System Development Life Cycle which is lately used to cope with the "slowness" of the conventional method. The Strength of using this method is the speed, accuracy and relatively lower cost than the conventional method. Moreover, the user's needs can be fulfilled well by involving the user in the design process. As a result, the user's satisfaction will increase. However, there are several things that should be considered in using the Rapid Application Development such as the team's preparation, the system territory, user's needs, and system operation. Finally, as one of the alternatives of System Development Life Cycle, Rapid Application Development can be used as the fundamental to produce an information system which is able to fulfill the user's needs.*

Keywords: *Rapid Application Development (RAD)*, *System Development life Cycle (SDLC)*.

1. LATANG BELAKANG

Rapid Application Development (RAD) adalah salah satu metode pengembangan suatu sistem informasi dengan waktu yang relatif singkat. 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.

Tujuan utama dari semua metode sistem development adalah memberikan suatu sistem yang dapat memenuhi harapan dari para pemakai, akan tetapi sering kali di dalam melakukan pengembangan suatu sistem tidak melibatkan para pemakai sistem secara langsung, sehingga hal ini menyebabkan sistem informasi yang dibuat jauh dari harapan pemakai yang dapat berakibat

sistem tersebut walaupun dapat diterima tetapi para pemakai enggan untuk menggunakannya atau bahkan para pemakai menolak untuk menggunakannya.

Pada saat *RAD* diimplementasikan, maka para pemakai bisa menjadi bagian dari keseluruhan proses pengembangan sistem dengan bertindak sebagai pengambil keputusan pada setiap tahapan pengembangan. *RAD* bisa menghasilkan suatu sistem dengan cepat karena sistem yang dikembangkan dapat memenuhi keinginan dari para pemakai sehingga dapat mengurangi waktu untuk pengembangan ulang setelah tahap implementasi.

2. KELEMAHAN-KELEMAHAN METODE KONVENSIONAL

Adapun kelemahan-kelemahan yang terdapat pada metode konvensional adalah sebagai berikut:

- Dengan metode konvensional, maka terdapat batas waktu yang cukup lama mulai dari pembuatan sistem sampai dengan konsumen dapat menggunakan sistem tersebut.
- Dengan metode konvensional, apabila proses pengembangan suatu sistem membutuhkan waktu yang lama maka kebutuhan konsumen pada sistem akan mengalami perubahan seiring dengan perubahan proses bisnis yang dilakukan oleh konsumen.
- Dengan metode konvensional, sistem yang dikembangkan tidak akan mempunyai manfaat apabila belum diselesaikan seluruhnya.

3. ALASAN MEMILIH METODE RAD

Di dalam memilih metode RAD harus memperhatikan alasan-alasan berikut ini:

3.1 Alasan yang Buruk

- Apabila menggunakan RAD hanya untuk menghemat biaya pengembangan suatu sistem. Hal ini disebabkan karena dengan menggunakan metode RAD membutuhkan suatu tim yang mengerti betul mengenai manajemen biaya. Sebab bila tidak, maka biaya yang dikeluarkan akan menjadi lebih besar.
- Apabila menggunakan RAD hanya untuk menghemat waktu pengembangan suatu sistem. Hal ini disebabkan karena dengan menggunakan metode RAD membutuhkan suatu tim yang mengerti betul mengenai manajemen waktu. Sebab bila tidak maka waktu yang dibutuhkan akan menjadi lebih lama.

3.2 Alasan yang Baik

- Apabila menggunakan RAD untuk mendapatkan suatu desain yang dapat diterima oleh konsumen dan dapat dikembangkan dengan mudah.

- Apabila menggunakan RAD untuk memberikan batasan-batasan pada suatu sistem supaya tidak mengalami perubahan.
- Apabila menggunakan RAD untuk menghemat waktu, dan kalau memungkinkan bisa menghemat biaya serta menghasilkan produk yang berkualitas.

4. *SCHEDULE* vs EKONOMI vs KUALITAS PRODUK

Ada beberapa hal yang perlu diperhatikan di dalam mengembangkan suatu sistem dengan menggunakan metode RAD dengan berdasarkan pada *schedule*, ekonomi dan kualitas produk antara lain model pengembangan, negosiasi, dan tujuan.

4.1 Model Pengembangan

Pada saat mengembangkan suatu sistem pasti dihadapkan dengan 3 pilihan model yaitu:

- *Efficient Development* (model pengembangan yang mengutamakan *schedule*, ekonomi dan kualitas produk secara seimbang).
 - *Schedule* → lebih cepat dari rata-rata
 - Ekonomi → biaya lebih murah dari rata-rata
 - Produk → lebih baik daripada kualitas rata-rata
- *Sensible RAD* (model pengembangan yang mengutamakan *schedule* dibandingkan dengan ekonomi dan kualitas produk).
 - *Schedule* → lebih cepat dari rata-rata
 - Ekonomi → biaya lebih murah sedikit dari rata-rata
 - Produk → lebih baik sedikit dari kualitas rata-rata
- *All-out RAD* (model pengembangan yang mengutamakan *schedule* dengan mengorbankan ekonomi dan kualitas produk).
 - *Schedule* → paling cepat
 - Ekonomi → biaya lebih mahal dari rata-rata
 - Produk → lebih buruk dari kualitas rata-rata

4.2 Negosiasi

Untuk menghasilkan suatu sistem yang sesuai dengan kebutuhan dan keinginan user maka perlu melakukan suatu negosiasi dan bukan hanya lebih mementingkan *schedule*.

- RAD dapat dilakukan dengan cukup sukses apabila konsumen mampu melakukan negosiasi untuk menentukan ekonomi atau kualitas dari suatu sistem.
- RAD bisa memperoleh kesuksesan yang lebih baik apabila konsumen mampu melakukan negosiasi untuk menentukan ekonomi dan kualitas dari suatu sistem.
- Akan tetapi hal yang perlu diperhatikan adalah bahwa yang dimaksud negosiasi mengenai kualitas adalah bukan berarti konsumen bisa menerima kesalahan yang semakin banyak, tetapi yang dimaksud negosiasi adalah produk yang diterima mempunyai kekurangan baik itu pada penggunaan, kelengkapan fasilitas atau kurang efisien.

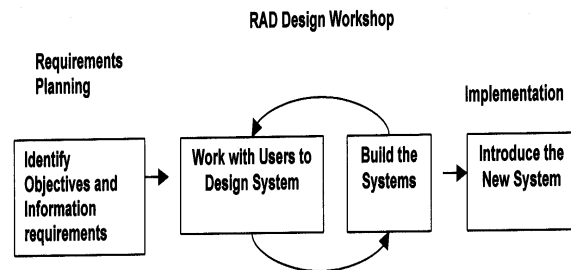
4.3 Tujuan

Dengan menggunakan RAD maka ada satu atau beberapa tujuan berikut ini yang tidak akan dapat dicapai secara bersama-sama yaitu:

- Kemungkinan terjadi kesalahan yang kecil, karena pihak pengembang tidak mempunyai hak untuk mengubah komponen-komponen yang digunakan dalam mengembangkan suatu sistem.
- Tingkat kepuasan konsumen yang tertinggi, karena kebutuhan-kebutuhan sekunder dari konsumen harus dikorbankan supaya suatu sistem dapat diselesaikan sesuai jadwal.
- Biaya pengembangan yang termurah, karena dengan menggunakan komponen yang sudah ada dapat menyebabkan biaya yang lebih besar apabila dibandingkan dengan mengembangkan komponen sendiri.

5. TAHAPAN-TAHAPAN PADA RAD

Metode RAD mempunyai 3 tahapan utama seperti yang terlihat pada gambar 1.



Gambar 1. Tahapan RAD

5.1 Rencana Kebutuhan (*Requirement Planning*)

Pada tahap ini, *user* dan *analyst* melakukan semacam pertemuan untuk melakukan identifikasi tujuan dari aplikasi atau sistem dan melakukan identifikasi kebutuhan informasi untuk mencapai tujuan. Pada tahap ini hal terpenting adalah adanya keterlibatan dari kedua belah pihak, bukan hanya sekedar persetujuan akan proposal yang sudah dibuat. Untuk lebih jauh lagi, keterlibatan *user* bukan hanya dari satu tingkatan pada suatu organisasi, melainkan beberapa tingkatan organisasi sehingga informasi yang dibutuhkan untuk masing-masing *user* dapat terpenuhi dengan baik. Di samping itu, dapat juga melakukan koordinasi dengan *Chief Information Office* (CIO) atau bagian perencana strategis terutama untuk mengembangkan suatu aplikasi *E-commerce* berbasis *Web* untuk mendapatkan informasi yang lebih detail akan tujuan dari suatu organisasi. Pertemuan semacam ini seringkali disebut *Joint Application Development*.

5.2 Proses Desain (*Design Workshop*)

Pada tahap ini adalah melakukan proses desain dan melakukan perbaikan-perbaikan apabila masih terdapat ketidaksesuaian desain antara *user* dan *analyst*. Untuk tahap ini maka keaktifan *user* yang terlibat sangat menentukan untuk mencapai tujuan, karena *user* bisa langsung memberikan komentar apabila terdapat ketidaksesuaian pada desain. Biasanya, *user* dan *analyst* berkumpul menjadi satu dan duduk di meja melingkar dimana masing-masing orang bisa melihat satu dengan yang lain tanpa ada halangan.

Apabila memungkinkan, maka masing-masing *user* diberikan satu komputer yang terhubung satu dengan yang lain, sehingga masing-masing bisa melihat desain yang dibuat dan langsung memberikan komentar. Hal ini sering kali disebut dengan *Group Decision Support System* (GDSS). Pada beberapa kasus, GDSS ini merupakan suatu langkah yang ideal, karena user dan analyst dapat menyetujui desain yang dibuat untuk kemudian dilanjutkan oleh *programmer* dalam pembuatan *prototype* dari aplikasi yang dimaksud dengan langsung menampilkan kepada user hasilnya dengan cepat.

Pada tahap desain ini membutuhkan waktu beberapa hari, akan tetapi bisa semakin lebih lama, tergantung dari besar kecilnya sistem yang dibuat. Pada selang waktu tersebut, *user* bisa memberikan tanggapan akan sistem yang sudah dikembangkan untuk selanjutnya dilakukan perbaikan-perbaikan. Dengan demikian proses pengembangan suatu sistem membutuhkan waktu yang cepat.

5.3 Implementasi (*Implementation*)

Setelah desain dari sistem yang akan dibuat sudah disetujui baik itu oleh user dan *analyst*, maka pada tahap ini *programmer* mengembangkan desain menjadi suatu program. Setelah program selesai baik itu sebagian maupun secara keseluruhan, maka dilakukan proses pengujian terhadap program tersebut apakah terdapat kesalahan atau tidak sebelum diaplikasikan pada suatu organisasi. Pada saat ini maka user bisa memberikan tanggapan akan sistem yang sudah dibuat serta persetujuan mengenai sistem tersebut.

Adapun hal terpenting adalah bahwa keterlibatan *user* sangat diperlukan supaya sistem yang dikembangkan dapat memberikan kepuasan kepada *user*, dan di samping itu, sistem yang lama tidak perlu dijalankan secara paralel dengan sistem yang baru.

5.4 Tahapan keseluruhan

Dengan berdasarkan pada tahapan-tahapan tersebut di atas maka proses utama pengembangan suatu sistem dengan meng-

gunakan metode RAD adalah sebagai berikut :

- Pengembang membuat *prototype* berdasarkan kebutuhan-kebutuhan yang sudah didefinisikan sebelumnya
- Desainer melakukan penilaian terhadap *prototype*
- User melakukan uji coba pada *prototype* dan memberikan masukan mengenai kebutuhan-kebutuhan yang kurang.
- *User* dan *developer* melakukan pertemuan untuk memberikan penilaian terhadap produk secara bersama-sama, menyesuaikan kebutuhan serta memberikan komentar apabila diperlukan perubahan.
- Semua kebutuhan akan sistem dan perubahan-perubahan yang terjadi dilakukan proses "*timeboxed*" dengan mempunyai 2 kemungkinan :
 - Perubahan yang tidak dapat ditampung seperti yang sudah direncanakan harus dihilangkan.
 - Jika diperlukan, kebutuhan-kebutuhan yang bersifat sekunder ditiadakan.

6. KONDISI-KONDISI YANG MEM-PENGARUHI RAD

Pada saat akan menggunakan metode RAD perlu memperhatikan kondisi-kondisi yang bisa menunjang dan menghambat keberhasilan dari suatu sistem.

6.1 Kondisi Penunjang

Beberapa kondisi yang dapat menunjang keberhasilan dari RAD adalah sebagai berikut :

- Sistem berjalan sendiri (*standalone*).
- Kinerja dari sistem bukan faktor terpenting.
- Distribusi produk yang bersifat sempit.
- Ruang lingkup yang terbatas.
- Keandalan dari sistem bukan faktor terpenting.
- Membutuhkan teknologi yang tidak terlalu baru (lebih dari 1 tahun).
- Sistem dapat dipecah-pecah menjadi bagian-bagian yang lebih kecil.

6.2 Kondisi Penghambat

Beberapa kondisi yang dapat menghambat keberhasilan dari RAD adalah sebagai berikut :

- Sistem harus dapat berjalan secara bersamaan dengan sistem yang lama.
- Komponen-komponen penunjang sangat langka untuk didapatkan.
- Kinerja yang optimal merupakan faktor terpenting.
- Distribusi produk yang bersifat luas.
- Ruang lingkup yang luas.
- Apabila digunakan untuk membuat Sistem Operasi, dimana membutuhkan sistem yang handal.
- Sistem tidak dapat dipecah-pecah menjadi bagian-bagian yang lebih kecil.

7. KEUNTUNGAN DAN KERUGIAN RAD

Dalam menggunakan RAD ada beberapa hal yang perlu diperhatikan terutama berkaitan dengan keuntungan dan kerugian.

7.1 Keuntungan RAD

Beberapa keuntungan dalam menggunakan metode RAD adalah sebagai berikut:

- Membeli sistem yang baru memungkinkan untuk lebih menghemat biaya ketimbang mengembangkan sendiri.
- Proses pengiriman menjadi lebih mudah, hal ini dikarenakan proses pembuatan lebih banyak menggunakan potongan-potongan *script*.
- Mudah untuk diamati karena menggunakan model *prototype*, sehingga user lebih mengerti akan sistem yang dikembangkan.
- Lebih fleksibel karena pengembang dapat melakukan proses desain ulang pada saat yang bersamaan.
- Bisa mengurangi penulisan kode yang kompleks karena menggunakan *wizard*.
- Keterlibatan *user* semakin meningkat karena merupakan bagian dari tim secara keseluruhan.
- Mampu meminimalkan kesalahan-kesalahan dengan menggunakan alat-alat bantuan (*CASE tools*).

- Mempercepat waktu pengembangan sistem secara keseluruhan karena cenderung mengabaikan kualitas.
- Tampilan yang lebih standar dan nyaman dengan bantuan *software-software* pendukung.

7.2 Kerugian RAD

Beberapa kerugian dalam menggunakan metode RAD adalah sebagai berikut :

- Dengan melakukan pembelian belum tentu bisa menghemat biaya dibandingkan dengan mengembangkan sendiri.
- Membutuhkan biaya tersendiri untuk membeli peralatan-peralatan penunjang seperti misalnya *software* dan *hardware*.
- Kesulitan melakukan pengukuran mengenai kemajuan proses.
- Kurang efisien karena apabila melakukan pengkodean dengan menggunakan tangan bisa lebih efisien.
- Ketelitian menjadi berkurang karena tidak menggunakan metode yang formal dalam melakukan pengkodean.
- Lebih banyak terjadi kesalahan apabila hanya mengutamakan kecepatan dibandingkan dengan biaya dan kualitas.
- Fasilitas-fasilitas banyak yang dikurangi karena terbatasnya waktu yang tersedia.
- Sistem sulit diaplikasikan di tempat yang lain.
- Fasilitas yang tidak perlu terkadang harus disertakan, karena menggunakan komponen yang sudah jadi, sehingga hal ini membuat biaya semakin meningkat karena harga komponen yang lebih lengkap semakin mahal.

8. KESIMPULAN

Berdasarkan pembahasan di atas, maka di dalam menggunakan metode RAD dapat disimpulkan sebagai berikut:

- Penggunaan RAD harus digunakan secara tepat, sebab bila tidak maka akan menimbulkan kerugian-kerugian seperti misalnya biaya yang semakin membengkak dan waktu yang semakin lama.
- Penggunaan metode RAD harus digunakan dengan mempertimbangkan aspek waktu dan biaya secara seimbang, tidak bisa diprioritaskan satu per satu.

- Sebagai salah satu alternatif dari SDLC maka RAD dapat dijadikan acuan untuk mengembangkan suatu sistem informasi yang unggul dalam hal kecepatan, ketepatan dan biaya yang lebih rendah.
- Dengan menggunakan RAD, maka keterlibatan *user* menjadi semakin meningkat yang pada akhirnya dapat meningkatkan kepuasan *user* terhadap sistem yang dikembangkan.

DAFTAR PUSTAKA

1. Kendal & Kendal. *Systems Analysis and Design Fifth Edition*. Prentice-Hall International, Inc. 2002.
2. Jeffrey A. Hoffer, Joey F. George, Joseph S. Valacich. *Modern Systems Analysis and Design Second Edition*. Addison-Wesley. 1997.
3. Jeffrey L. Whitten, Lonnie D. Bentley, Kevin C. Dittman. *Systems Analysis and Design Methods*. McGraw-Hill. 2001.
4. Ramez El Masri, Shamkant B. Navache. *Fundamental of Database Systems*. The Benjamin/Cummings Publishing Company, Inc. 1994.
5. Raymond McLeod, George Schell. *Management Information Systems 8/e*. Prentice-Hall, Inc. 2001.