

Implementasi Metode Rapid Application Development (RAD) Dalam Membangun E-Retail

Supratman Zakir

Institut Agama Islam Negeri (IAIN) Bukittinggi, Indonesia
e-mail : supratman@iainbukittinggi.ac.id,

Efmi Maiyana

Akademi Manajemen Informatika dan Komputer (AMIK) Boekittinggi, Indonesia
e-mail : efmi_maiyana@yahoo.com

Abstract

The appearance of Internet technology in recent years requires a lot of people to do all the activities such as economic activity quickly, accurately and efficiently as possible in order to obtain maximum results. One of the economic activity was much interested in is e-retail. E-Retail is the retail activities (retail trade) is carried out electronically via the Internet or also called virtual shopping. E-retail advantages compared with conventional retail scope is broader, faster receiving and processing orders, customer access to information increasingly intensive and always up to date and area of buying and selling are always changing. E-retail excellence makes e-retail to be one of the promising solutions for economic activity. However, building an e-retailers need a comprehensive and in-depth study so that economic activities through e-retail became one of the drivers of economic growth. One of the approaches or methods that can be used in building e-retail is a Rapid Application Development (RAD). RAD is a method of developing an information system that has the advantage of speed, accuracy and cost relatively lower than with conventional methods.

Keyword : e-retail, RAD, ekonomi, virtual shopping, e-commerce

Abstrak

Munculnya teknologi Internet dalam beberapa tahun belakangan ini menuntut banyak orang untuk melakukan segala aktifitas seperti aktivitas perekonomian secara cepat, tepat dan seefisien mungkin demi mendapatkan hasil yang maksimal. Salah satu aktivitas perekonomian yang sedang banyak minati adalah e-retail. *E-Retail* adalah kegiatan retail (perdagangan eceran) yang dilaksanakan secara elektronik melalui internet atau disebut juga virtual shopping. Kelebihan e-retail dibandingkan dengan retail konvensional adalah ruang lingkup lebih luas, lebih cepat menerima dan memproses pesanan, akses pelanggan terhadap informasi semakin intensif dan selalu up to date dan area jual beli yang selalu berubah. Keunggulan e-retail ini membuat e-retail menjadi salah satu solusi aktivitas perekonomian yang menjanjikan. Namun demikian membangun sebuah e-retail butuh kajian yang komprehensif dan mendalam sehingga aktivitas perekonomian melalui e-retail menjadi salah satu pendorong pertumbuhan perekonomian. Salah satu pendekatan atau metode yang dapat digunakan dalam membangun e-retail adalah *Rapid Application Development* (RAD). RAD adalah satu metode pengembangan suatu sistem informasi yang memiliki keunggulan kecepatan, ketepatan dan biaya yang relatif lebih rendah dibanding dengan metode konvensional.

Keyword : e-retail, RAD, ekonomi, virtual shopping, e-commerce

Latar Belakang

Pesatnya perkembangan teknologi saat ini menuntut banyak orang untuk melakukan segala aktifitas secara cepat, tepat dan seefisien mungkin demi mendapatkan hasil yang semaksimal mungkin.

Untuk mendukung semuanya itu penggunaan internet dalam setiap kegiatan sangatlah diperlukan untuk memperlancar aktifitas dalam mendapatkan informasi yang ada di sekitar kita dan sebagai alat komunikasi dengan orang lain.

Dengan munculnya teknologi Internet dalam beberapa tahun ini, banyak istilah baru yang menggunakan awalan e-xxx, seperti halnya: e-mail, e-commerce, e-goverment, e-society, e-retail dll. **E-Retail** adalah kegiatan retail (perdagangan eceran) yang dilaksanakan secara elektronik melalui internet. Kata "e" dalam e-retail ini berarti "elektronik". e-retail atau disebut juga virtual shopping muncul di internet pada tahun 1994.

Kelebihan e-retail dibandingkan dengan retail konvensional adalah ruang lingkup lebih luas, lebih cepat menerima dan memproses pesanan, akses pelanggan terhadap informasi semakin intensif dan selalu up to date, area jual beli yang selalu berubah, kemandirian eksekusi; selain ketersediaan produk dan kemudahan pembayaran, konsumen juga menuntut kecepatan pengiriman produk.

Untuk membangun e-retail kita dapat menggunakan metode Rapid Application Development (RAD) yang merupakan salah satu alternatif dari *system Development Life Cycle* (SDLC) yang belakangan ini seringkali digunakan untuk mengatasi keterlambatan yang terjadi apabila menggunakan metode konvensional.

Keunggulan metode ini adalah ; kecepatan, ketepatan dan biaya yang relatif lebih rendah dibanding dengan metode konvensional. Jika untuk pengembangan suatu sistem informasi yang normal membutuhkan waktu minimal 180 hari, dengan menggunakan metode RAD suatu sistem dapat diselesaikan hanya dalam waktu 30-90 hari. Selain itu metode ini melibatkan user pada proses desain sehingga kebutuhan user dapat terpenuhi dengan baik dan otomatis kepuasan user sebagai pengguna sistem semakin meningkat

Pembahasan

Dalam membangun sistem E-Retail berbasis RAD dapat dikelompokkan ke dalam 12 langkah yaitu: *set goal and scope, select lifecycle, set organization/team form, start team selection, determine risks, create WBS, identify tasks, estimate size, estimate effort, identify task dependencies, assign resources dan schedule work*

Kelemahan-kelemahan Metode Konvensional

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

- a. Terdapat batas waktu yang cukup lama mulai dari pembuatan sistem sampai dengan konsumen dapat menggunakan sistem tersebut.
- b. 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.
- c. Sistem yang dikembangkan tidak akan mempunyai manfaat apabila belum diselesaikan seluruhnya.

Rapid Application Development

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.

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.

Ide Dasar dari RAD

Beberapa ide yang mendasari penggunaan RAD dalam membangun e-retail adalah sebagai berikut :

- a. Lebih melibatkan user dari sistem secara aktif dalam kegiatan analisis, desain, dan konstruksi.
- b. Untuk mengatur pengembangan sistem kepada serangkaian workshop yang intense dan fokus yang melibatkan pemilik sistem, user, analis, designer, dan pembuat.
- c. Untuk mempercepat analisis kebutuhan dan fase desain melalui pendekatan konstruksi yang iteratif.
- d. Untuk mengurangi jumlah waktu yang diperlukan sampai user mulai melihat sistem yang bekerja.

RAD menggunakan prototipe untuk mempercepat analisis kebutuhan dan desain sistem. Prototipe adalah representasi/working model dari kebutuhan user atau design yang diusulkan untuk sebuah sistem informasi.

Prinsip dasar dari prototyping adalah user mengetahui apa keinginan mereka ketika prototipe tersebut berfungsi. Pada RAD, sebuah prototipe biasanya pada akhirnya menjadi sistem informasi final.

Alasan Memilih Metode RAD

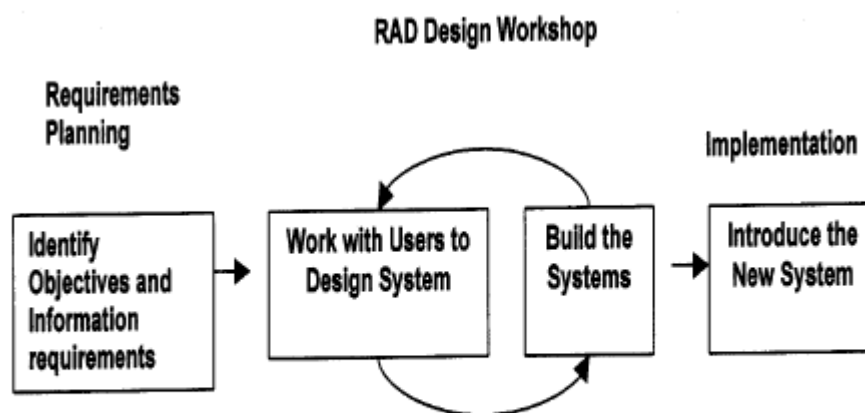
Di dalam memilih metode RAD harus memperhatikan alasan-alasan berikut ini:

- a. RAD dapat 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.

- b. RAD dapat 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
- c. RAD dapat menghasilkan suatu desain yang dapat diterima oleh konsumen dan dapat dikembangkan dengan mudah.
- d. RAD dapat memberikan batasan-batasan pada suatu sistem supaya tidak mengalami perubahan.

Tahapan-tahapan pada RAD

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



Gambar 1. Tahapan RAD

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. Disamping 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*.

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.

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.

Tahapan keseluruhan

Dengan berdasarkan pada tahapan-tahapan tersebut di atas maka proses utama pengembangan suatu sistem dengan menggunakan metode RAD adalah sebagai berikut :

- a. Pengembang membuat *prototype* berdasarkan kebutuhan-kebutuhan yang sudah didefinisikan sebelumnya
- b. Desainer melakukan penilaian terhadap *prototype*
- c. User melakukan uji coba pada *prototype* dan memberikan masukan mengenai kebutuhan-kebutuhan yang kurang.
- d. *User* dan *developer* melakukan pertemuan untuk memberikan penilaian terhadap produk secara bersama-sama, menyesuaikan kebutuhan serta memberikan komentar apabila diperlukan perubahan.

- e. Semua kebutuhan akan sistem dan perubahan-perubahan yang terjadi dilakukan proses “*timeboxed*” dengan mempunyai 2 kemungkinan :
 - i. Perubahan yang tidak dapat ditampung seperti yang sudah direncanakan harus dihilangkan.
 - ii. Jika diperlukan, kebutuhan-kebutuhan yang bersifat sekunder ditiadakan.

Tahapan dalam membangun e-retail berbasis RAD

1. Menetapkan tujuan

Tujuan akhir dari project ini adalah untuk membangun perangkat lunak e-retail. Tujuan Pengembangan:

- a. Meminimalisir kesalahan karena konsumen dilibatkan dalam proses.
- b. Meningkatkan kepuasan konsumen karena keinginan konsumen dapat di akomodir.
- c. Biaya yang lebih murah karena pengembangan menggunakan komponen yang telah ada.
- d. Jangkauan lebih luas

2. Menentukan Life Cycle

Pada tahap ini kita menetapkan model pengembangan perangkat lunak e-retail yaitu : Model Rapid Application Development (RAD) RAD adalah model pengembangan perangkat lunak yang menekankan siklus perkembangan yang sangat pendek (Pressman: 2002)

3. Menentukan Team Work

Team work terdiri dari :

- a. *Manager Senior*, yang menentukan isu-isu bisnis yang memiliki pengaruh penting di dalam proyek: yaitu Pimpinan/Manager Perusahaan
- b. *Manager (teknik) Proyek*, yang merencanakan, memotivasi, mengorganisasi, dan mengontrol sebuah produk atau aplikasi : Pimpinan / Manager Proyek
- c. *Pelaksana*, yang menyampaikan ketrampilan teknik yang diperlukan untuk merekayasa aplikasi pembangunan sistem e-retail: Sistem Analis, Pendesain sistem, Programmer, Teknisi, operator
- d. *Pelanggan*, yang menentukan jenis kebutuhan bagi perangkat lunak yang akan direkayasa. Dalam hal ini adalah masyarakat yang membutuhkan produk / pemakai/stake holder.

4. Start Team Selection

Merekrut anggota-anggota tim yang “benar” dapat membuat sebuah proyek. Berikut panduan untuk menyeleksi dan merekrut tim :

- a. Orang-orang punya motivasi tinggi dan memiliki talenta.

- b. Memilih tugas terbaik untuk tiap orang.
- c. Mendorong harmoni tim.
- d. Merencanakan masa depan.
- e. Jaga jumlah anggota tim tetap kecil.

5. Menentukan Resiko

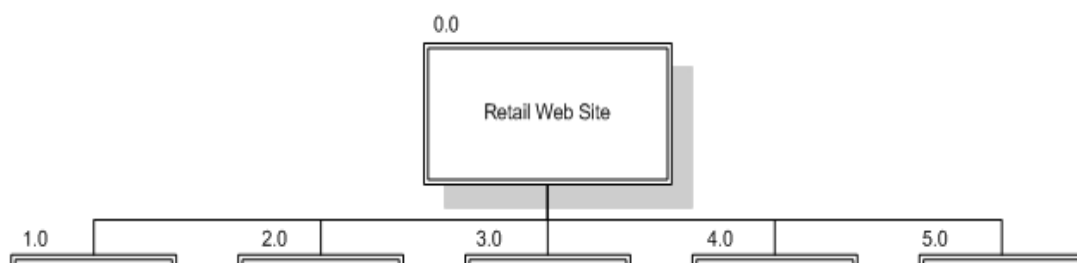
Dalam pembangunan e-retail ini resiko yang harus diperhitungkan adalah :

- a. *Resiko proyek* yaitu resiko yang mempengaruhi schedule, atau sumber daya proyek(keluarnya staff, perubahan manajemen, tidak tersediannya perangkat keras dll)
- b. *Resiko product* yaitu resiko yang mempengaruhi kualitas atau kinerja perangkat lunak yang dikembangkan (penaksiran ukuran yang terlalu kecil, kinerja case tool yang tidakmencukupi)
- c. *Resiko Bisnis* yaitu resiko yang mempengaruhi organisasi yang sedang mengadakan /mengembangkan PL(perubahan teknologi, kompetisi produk))

6. Membuat Work Breakdown Structure (WBS) Uraian Pekerjaan/Aktifitas E-Retail

- 1. Project Management
- 2. Requirements Gathering
- 3. Analysis & Design
- 4. Site Software Development
 - 4.1 HTML Design and Creation
 - 4.2 Backend Software
 - 4.2.1 Database Implementation
 - 4.2.2 Middleware Development
 - 4.2.3 Security Subsystems
 - 4.2.4 Catalog Engine
 - 4.2.5 Transaction Processing
- 4.3 Graphics and Interface
- 4.4 Content Creation
- 5. Testing and Production

WBS CHART E-RETAIL



7. Menentukan Task/Aktifitas

Task/aktivitas yang harus dilakukan dalam pembangunan e-retail ini adalah :

a. *Project Management*

Manajemen Proyek dimulai sebelum aktivitas teknis diinisiasi dan berlanjut pada keseluruhan batasan, perkembangan dan pemeliharaan perangkat lunak komputer. Menurut Pressman, manajemen proyek dipengaruhi "3 P" yaitu: People, Problem dan Procces.

- i. *People*: "manusia" yaitu dimulai dari proses pembentukan tim-tim yang efektif yang harus diorganisasi (mempertemukan orang-orang dengan tugas-tugas)
- ii. *Problem* : harus di komunikasikan dari pelanggan dari pengembang.
- iii. *Process* : harus disesuaikan dengan manusia dan masalah.

b. *Requirements Gathering*

Tahap yang paling utama dan sulit yaitu meramalkan /mengantisipasi kebutuhan-kebutuhan masa depan dari pelanggan-pelanggan

c. *Analysis & Design*

Pada tahap analysis dan design meliputi aktifitas sbb:

- i. **Survey** : berfungsi untuk mengetahui kebutuhan pemakai, kesalahan-kesalaan dalam sistem lama, menetapkan tujuan perancangan, mengajukan usulan otomasi sistem yang layak dan dapat diterima, dan menyiapkan laporan survey yang berisi tentang segala sesuatunya.
- ii. **Analisa Sistem** : menggabungkan laporan survey dan kebijakan pemakai menjadi spesifikasi yg terstruktur dengan menggunakan permodelan.
- iii. **Desain** : mengimplementasikan model yang diinginkan pemakai.
- iv. **Instalasi** : aspek terakhir yang mesti dilakukan mencakup, serta terimamanual, perangkat keras dan pelatihan pemakaian.

d. Site Software Development

Tahap menentukan lokasi Pengembangan Perangkat Lunak. Pada tahap ini kegiatan yang dilakukan meliputi:

1. HTML Design and Creation
2. Backend Software
 - 2.1. Database Implementation
 - 2.2. Middleware Development
3. Security Subsystems
 - 3.1. Catalog Engine
 - 3.2. Transaction Processing
4. Graphics and Interface
5. Content Creation

Perancangan HTML, Mengimplementasikan model yang dipakai dan mempresentasikan hasil desain ke dalam pemrograman. Pengembang dapat menyisipkan kebutuhan-kebutuhan/kegiatan pada saat kegiatan pengembangan berlangsung.

Pada tahap ini juga, dilakukan penentuan security pada bagian-bagian yang dianggap penting. Perlunya membuat daftar yang terorganisir dari elemen data yang berhubungan dengan sistem, dengan definisi yang jelas sehingga pemakai dan analis sistem akan memiliki pemahaman yang umum mengenai input, output, komponen penyimpanan dan bahkan kalkulasi intermediate sampai dengan bentuk tampilan antar muka.

e. Testing and Production

Harus dilaksanakan pemeriksaan dan perbaikan proyek secara berkesinambungan, diantaranya :

- a. Apakah produk telah memenuhi atau lebih dari harapan-harapan pengguna ?
- b. Apakah Proyek dapat diselesaikan tepat waktu ?
- c. Apakah Proyek sesuai dengan anggaran yang telah ditetapkan?

8. Estimate Size

- a. Sebelum kegiatan Perencanaan pengembangan sistem e-retail dilakukan, harus memperhitungkan 3 hal: berapa lama proyek akan berlangsung, berapa banyak usaha yang diperlukan dan berapa banyak manusia yang terlibat.

- b. Pengukuran langsung dari *proses* rekayasa perangkat lunak menyangkut biaya dan usaha yang diaplikasikan. Pengukuran dari produk menyangkut deretan kode yang diproduksi, kecepatan eksekusi, ukuran memori dan cacat yang dilaporkan pada sejumlah periode tertentu.
- c. Pengukuran tidak langsung dari produk menyangkut fungsionalitas, kualitas, kompleksitas, efisiensi, rehabilitas, kemampuan pemeliharaan.

9. Estimate Effort

- a. Menurut McConnell, pada estimation effort dapat ditunjukkan dengan tabel-tabel yang mencantumkan daftar setiap proyek pengembangan perangkat lunak yang sudah diselesaikan pada tahun-tahun terakhir dan pengukuran yang sesuai untuk proyek tersebut. Estimation effort dapat dibuat secara matematika. Seringkali langkah-langkah penilaian ukuran dan usaha dikombinasikan.
- b. Tabel dibawah ini dapat menunjukkan estimasi effort :
- c. Perkiraan proyek perangkat lunak tidak pernah dapat menjadi ilmu yang pasti, tetapi suatu kombinasi data historis yang baik serta teknik yang sistematis akan dapat meningkatkan ketepatan perkiraan (Pressman).

Perkiraan proyek perangkat lunak tidak pernah dapat menjadi ilmu yang pasti, tetapi suatu kombinasi data historis yang baik serta teknik yang sistematis akan dapat meningkatkan ketepatan perkiraan (Pressman).

10. Identify Task Dependencies

Tahap mengidentifikasi ketergantungan antar kegiatan merupakan tahap menyelaraskan tiap-tiap kegiatan, sehingga terjadi keseimbangan untuk setiap kegiatan, misalnya antara ukuran, waktu, skedul, biaya dan sumber daya .

11. Assign Resources

- a. Menentukan siapa (sumber daya) yang akan mengerjakan tiap-tiap kegiatan dalam pembangunan sistem e-retail.
- b. Kegiatan pengembangan sistem harus dibagi ke dalam sejumlah tugas dan aktifitas yang dapat dikendalikan.
 - i. Pimpinan perusahaan bertugas menentukan isu-isu bisnis yang memiliki pengaruh penting di dalam proyek.
 - ii. Manager Proyek mempunyai tugas sebagai penganalisa sistem: merencanakan, memotivasi, mengorganisasi, dan mengontrol produk atau aplikasi pengembangan sistem e-retail. Menelusuri Sistem aktual, Mengembangkan/Usulan Sistem baru, Menjalankan fungsi komunikasi dari semua level (pemakai, manajer, programmer, dan pelaku sistem yang lain)

yang belum mempunyai sikap dan cara pandang yang sama. Penganalisa sistem adalah penanggung-jawab pekerjaan.

- iii. Pendesain sistem bertugas mentransformasikan hasil analisa sistem ke dalam desain arsitektur.
- iv. Programer bertugas sebagai pembuat program
- v. Operator tugasnya sebagai penanggung jawab untuk menjalankan sistem (seperti proses input, jaringan, keamanan perangkat keras, keamanan perangkat lunak, pencetakan dan back-up)

12. Schedule Work

- a. Setelah kegiatan WBS dan Estimasi selesai maka kegiatan berikutnya adalah pembuatan jadwal kerja. Sasarannya adalah: menghitung waktu yang terbaik, biaya rendah dan resiko kecil.
- b. Penjadwalan merupakan titik puncak dari aktivitas perencanaan yang menjadi komponen utama dari manajemen proyek pengembangan e-retail.
- c. Penjadwalan dimulai dengan dekomposisi proyek. Jaringan kerja menggambarkan setiap tugas rekayasa, ketergantungan pada tugas-tugas yang lain dan durasi proyeknya. Jaringan kerja digunakan untuk menghitung jalur proyek kritis, diagram timeline, dan sebagai informasi proyek. Dengan menggunakan jadwal sebagai panduan, manager proyek dapat menelusuri dan mengontrol setiap langkah dalam proyek rekayasa perangkat lunak.

Kondisi-kondisi yang mempengaruhi RAD

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

Kondisi Penunjang

Beberapa kondisi yang dapat menunjang keberhasilan dari RAD seperti sistem berjalan sendiri (*standalone*), kinerja dari sistem bukan faktor terpenting, distribusi produk yang bersifat sempit, ruang lingkup yang terbatas, kehandalan 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.

Kondisi Penghambat

Beberapa kondisi yang dapat menghambat keberhasilan dari RAD seperti 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, mempercepat waktu pengembangan sistem secara keseluruhan karena cenderung mengabaikan kualitas, tampilan yang lebih standar dan nyaman dengan bantuan *software-software* pendukung.

Keuntungan dan kerugian RAD

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

Keuntungan RAD

Beberapa keuntungan dalam menggunakan metode RAD adalah sebagai berikut:

- a. Membeli sistem yang baru memungkinkan untuk lebih menghemat biaya ketimbang mengembangkan sendiri.
- b. Proses pengiriman menjadi lebih mudah, hal ini dikarenakan proses pembuatan lebih banyak menggunakan potonganpotongan *script*.
- c. Mudah untuk diamati karena menggunakan model *prototype*, sehingga user lebih mengerti akan sistem yang dikembangkan.
- d. Lebih fleksibel karena pengembang dapat melakukan proses desain ulang pada saat yang bersamaan.
- e. Bisa mengurangi penulisan kode yang kompleks karena menggunakan *wizard*.
- f. Keterlibatan *user* semakin meningkat karena merupakan bagian dari tim secara keseluruhan.
- g. Mampu meminimalkan kesalahan-kesalahan dengan menggunakan alat-alat bantuan (*CASE tools*).
- h. Mempercepat waktu pengembangan sistem secara keseluruhan karena cenderung mengabaikan kualitas.
- i. Tampilan yang lebih standar dan nyaman dengan bantuan *software-software* pendukung.
- j. Berguna untuk proyek yang kebutuhan usernya tidak pasti
- k. Mendorong partisipasi user & manajemen
- l. Proyek memiliki visibility dan suport yang lebih baik
- m. User & manajemen dapat melihat solusi yang berbasiskan software dengan lebih cepat daripada pada model-driven development
- n. Error dapat dideteksi lebih awal
- o. Testing dan training merupakan satu kesatuan dengan pendekatan prototyping
- p. Pendekatan iteratif lebih alami karena perubahan merupakan faktor yang diperkirakan selama perkembangan
- q. Mengurangi resiko karena adanya pengecekan secara iteratif.

Kerugian RAD

Beberapa kerugian dalam menggunakan metode RAD adalah sebagai berikut :

- a. Dengan melakukan pembelian belum tentu bisa menghemat biaya dibandingkan dengan mengembangkan sendiri.
- b. Membutuhkan biaya tersendiri untuk membeli peralatan-peralatan penunjang seperti misalnya *software* dan *hardware*.
- c. Kesulitan melakukan pengukuran mengenai kemajuan proses.
- d. Kurang efisien karena apabila melakukan pengkodean dengan menggunakan tangan bisa lebih efisien.
- e. Ketelitian menjadi berkurang karena tidak menggunakan metode yang formal dalam melakukan pengkodean.
- f. Lebih banyak terjadi kesalahan apabila hanya mengutamakan kecepatan dibandingkan dengan biaya dan kualitas.
- g. Fasilitas-fasilitas banyak yang dikurangi karena terbatasnya waktu yang tersedia.
- h. Sistem sulit diaplikasikan di tempat yang lain.
- i. 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.
- j. RAD meningkatkan lifetime cost untuk menjaga sistem agar tetap terjaga
- k. Prototipe RAD bisa memecahkan masalah yang salah
- l. Prototipe RAD tidak membuat analisis melihat alternatif lain
- m. RAD dapat merusak kualitas karena adanya shortcut-shortcut yang dibuat.

Kesimpulan

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

- a. 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.
- b. Penggunaan metode RAD harus digunakan dengan mempertimbangkan aspek waktu dan biaya secara seimbang, tidak bisa diprioritaskan satu per satu.
- c. 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.
- d. Dengan menggunakan RAD, maka keterlibatan *user* menjadi semakin meningkat yang pada akhirnya dapat meningkatkan kepuasan *user* terhadap sistem yang dikembangkan.

Referensi

- [1] Candra Ahmadi dan Dadang Hermawan, (2014), *e-Business and e-commerce*, Andi Offset, Yogyakarta
- [2] Dennis, Charles,. Fenech, Tino,. and Merrilees, Bill (2004), *e-Retailing*, Routledge, Australia
- [3] Laudon, Kenneth C and Travis, Jeffrey (2015), *E-commerce: business. technology. society.*, Prentice Hall • Cloth, 944 pp Published 08/14/2003
- [4] Martin, James (1991), *Rapid Application Development*, Macmillan Publishing Company, Third Edition, Michigan
- [5] McMahon, David, (1999), *Rapid Application Development with Visual C++*, McGraw-Hill, USA
- [6] Onno W. Purbo, dan Akhmad Daniel S., (2000), *Membangun Web E-Commerce*, PT. Elex Media Komputindo, Jakarta.
- [7] Wahana Komputer (2002), *Apa & Bagaimana E-Commerce* , Andi Publisher , Yogyakarta.