MENGAPA PROYEK PERANGKAT LUNAK GAGAL
(PENERAPAN MANAJEMEN RESIKO DALAM PROYEK PERANGKAT LUNAK)

Yasmi Afrizal
Dosen Jurusan Manajemen Informatika
Universitas Komputer Indonesia

ABSTRAK

Tingkat kegagalan proyek perangkat lunak terbukti sangat tinggi. Hal ini memungkinkan terjadinya kegagalan proyek dalam mengembangkan perangkat lunak. Manajemen resiko adalah suatu pendekatan yang mengarahkan untuk meminimalkan atau mengurangi dampak kegagalan tersebut. Tujuan penelitian ini adalah menjelaskan tentang prinsip dasar model manajemen resiko dengan membuat kesimpulan yang penting dari setiap konsep yang direkomendasikan dari beberapa jurnal yang berbeda. Manajemen resiko memililki langkah penyelesaian yang terdiri dari identifikasi resiko, analisis, strategi pengurangan resiko dan pengendalian resiko. Penelitian ini sangat bermanfaat untuk membantu akademisi, peneliti dan praktisi dalam menerapkan manajemen resiko dalam proyek perangkat lunak

Key Words : Kegagalan proyek, manajemen resiko, analisis resiko

I. PENDAHULUAN

Pada saat kita mengerjakan pengembangan perangkat lunak sering kita menghadapi berbagai situasi yang tidak nyaman seperti keterlambatan pengembangan atau pengeluaran biaya pengembangan yang melebihi anggaran. Hal ini dikarenakan kurang siapnya kita menghadapi berbagai kemungkinan resiko yang akan terjadi. Untuk itu perlu dilakukan identifikasi tindakan yang harus dilakukan untuk mencegah ataupun meminimalkan resiko tersebut.

Tujuan utama manajemen resiko adalah mengenali semua kemungkinan kegagalan dari suatu proyek perangkat lunak dengan melihat kekomplekan dalam memutuskan langkah solusi yang akan dibuat secara alami [Boehm, B. W. 1998]. Solusi pemecahan masalah dilakukan dengan meminimalkan segala macam ketidakjelasan yang muncul di awal proyek dan melakukan evaluasi terhadap pemecahan tersebut

Beberapa resiko lebih penting dibandingkan resiko lainnya. Baik penting maupun tidak sebuah resiko tertentu bergantung pada sifat resiko

tersebut, pengaruhnya pada aktifitas tertentu dan kekritisan aktifitas tersebut. Aktifitas beresiko tinggi pada jalur kritis pengembangan biasanya merupakan penyebabnya.

Untuk mengurangi bahaya tersebut, maka harus ada jaminan untuk meminimalkan resiko atau paling tidak mendistribusikannya selama pengembangan tersebut dan idealnya resiko tersebut dihapus dari aktifitas yang mempunyai jalur yang kritis.

Resiko dari sebuah aktifitas yang sedang berlangsung sebagian bergantung pada siapa yang mengerjakan atau siapa yang mengelola aktifitas tersebut. Evaluasi resiko dan alokasi staf dan sumber daya lainnya erat kaitannya.

I. ANALISIS RESIKO

Barry Boehm merupakan peneliti pertama yang berhasil membawa manajemen resiko ke dalam area rekayasa perangkat lunak (Boehm 1981; Boehm 1987; Boehm 1988; Boehm 1989; Boehm 1991; Boehm 1992). Sumbangan utama dari Boehm adalah mengenalkan teknik pengukuran resiko dalam suatu model proses pengembangan perangkat lunak spiral (Boehm 1988). manajemen resiko Boehm meletakan standar pengukuran resiko yang disebut risk explosure dengan bentuk :

Risk Exposure = Probability (Outcome) * Loss(Outcome)……...(1)

Probabilitas terjadinya resiko sering disebut dengan *risk likelihood*; sedangkan dampak yang akan terjadi jika resiko tersebut terjadi dikenal dengan *risk impact* dan tingkat kepentingan resiko disebut dengan *risk value* atau *risk exposure*.

Idealnya risk impact diestimasi dalam batas moneter dan likelihood dievaluasi sebagai sebuah probabilitas. Dalam hal ini risk exposure akan menyatakan besarnya biaya yang diperlukan berdasarkan perhitungan analisis biaya manfaat. Risk exposure untuk berbagai resiko dapat dibandingkan antara satu dengan lainnya untuk mengetahui tingkat kepentingan masing-masing resiko.

Akan tetapi, estimasi biaya dan probabilitas tersebut sulit dihitung, subyektif, menghabiskan waktu dan biaya. Untuk mengatasi hal ini maka diperlukan beberapa pengukuran yang kuantitatif untuk menilai risk likelihood dan risk impact, karena tanpa ini sulit untuk membandingkan atau meranking resiko tersebut untuk berbagai keperluan. Akan tetapi, usaha yang dilakukan untuk medapatkan sebuah estimasi kuantitatif yang baik akan menghasilkan pemahaman yang mendalam dan bermanfaat atas terjadinya suatu permasalahan.

Beberapa manajer resiko mempergunakan sebuah metode penilaian yang sederhana untuk menghasilkan ukuran yang kuantitatif pada saat mengevaluasi masing-masing resiko. Beberapa manajer memberikan kategori pada likelihood dan impact dengan high, medium atau low. Akan tetapi bentuk ini tidak memungkinkan

untuk menghitung risk exposure. Sebuah pendekatan yang lebih baik dan populer adalah memberikan skor pada likelihood dan impact dengan skala tertentu misal 1-10. Jika suatu resiko kemungkinan besar akan terjadi diberi skor 10, sedangkan jika kecil jika kemungkinan terjadinya kecil maka akan diberi nilai 1.

Penilaian likelihood dan impact dengan skala 1-10 relatif mudah, akan tetapi kebanyakan manajer resiko akan berusaha untuk memberikan skor yang lebih bermakna, misal skor likelihood 8 akan dipertimbangkan dua kali likelihood dengan skor 4.

Hasil pengukuran impact, dapat diukur dengan skor yang serupa, harus dimasukkan pada perhitungan total risk dari proyek tersebut. Untuk itu harus melibatkan beberapa biaya potensial seperti :

a. Biaya yang diakibatkan keterlambatan penyerahan atas jadwal yang sudah ditentukan
b. Biaya yang berlebihan dikarenakan harus menambah sumber daya atau dikarenakan mempergunakan sumber daya yang lebih mahal
c. Biaya yang tidak terlihat pada beberapa komponen kualitas atau fungsionalitas sistem

Tabel 1 berikut ini memperlihatkan contoh hasil evaluasi nilai resiko. Perhatikan bahwa resiko yang bernilai tertinggi tidak selalu akan menjadi resiko yang pasti terjadi maupun akan menjadi resiko dengan potensi impact yang terbesar.

Tabel 1 . Contoh evaluasi nilai risk exposure

Resiko	Keterangang	Prob.	loss	RE
R1	Perubahan spesifikasi kebutuhan selama coding	1	8	8
R2	Spesifikasi perlu lebih lama dibandingkan yang diperlukan	3	7	21
R3	Staf sakit yang berpengaruh pada aktifitas yang kritis	5	7	35
R4	Staf sakit yang berpengaruh pada aktifitas yang tidak kritis.	10	3	30
R5	Pengkodean modul lebih lama dibandingkan yang diharapkan	4	5	20
R6	Pengujian modul memperlihatkan kesalahan atau ketidakefisiensian dalam desain.	1	10	10

II.

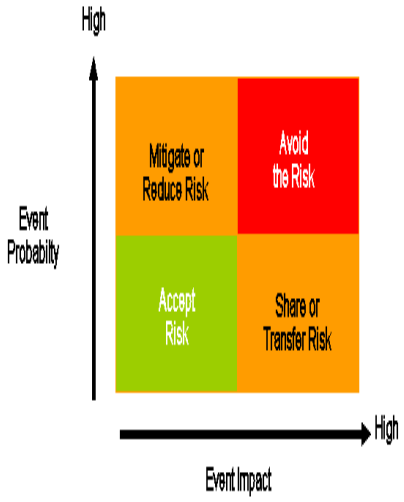
III. PENGENDALIAN RESIKO

Sebarang usaha untuk mengurangi sebuah risk exposure atau untuk melakukan sebuah rencana kontigensi akan berhubungan dengan biaya yang berkaitan dengan usaha tersebut. Merupakan hal yang penting untuk menjamin bahwa usaha ini dilaksanakan dengan cara yang paling efektif dan diperlukan cara untuk memprioritaskan resiko sehingga usaha yang lebih penting dapat menerima perhatian yang lebih besar. Pengelolaan terhadap pengendalian resiko melibatkan penggunaan dua strategi:

1. Risk exposure dapat dikurangi dengan mengurangi likehood atau impact
2. Pembuatan rencana kontingensi berkaitan dengan kemungkinan resiko yang terjadi.

Estimasi nilai likehood dan impact dari masing-masing usaha tersebut akan menentukan nilai risk exposure. Setelah risk exposure dapat dihitung maka resiko dapat diberi prioritas high, medium atau low sesuai dengan besar kecilnya nilai risk exposure.

Risk exposure yang berdasarkan pada metode penilaian perlu diberikan beberapa perhatian. Hasil evaluasi pada tabel 1, contoh, tidak memperlihatkan resiko R5 adalah dua kali lebih penting dibandingkan R6. Pada kasus ini, kita tidak bias mengintepretasikan nilai risk exposure secara kuantitatif disebabkan nilai tersebut didasarkan pada metode penilaian yang non-cardinal. Pada kasus kedua, nilai exposure yang terlalu berjauhan akan mampu untuk membedakan antara resiko tersebut. Akan tetapi risk exposure akan memungkinkan kita untuk memperoleh suatu ranking sesuai dengan kepentingannya. Pertimbangkan resiko pada tabel 1, R3 dan R4 merupakan resiko yang paling penting dan kita dapat mengklasifikasikannya dengan high risk. Tingkat kepentingan yang berbeda dapat membedakan antara skor exposure satu dan dua ini dengan exposure tertinggi berikutnya yaitu R2. R2 dan R5 mempunyai skor yang hampir sama dan dapat dikelompokkan pada resiko dengan prioritas medium. Dua resiko lainnya, R1 dan R6 mempunyai nilai exposure yang rendah sehingga dapat dikelompokan pada prioritas rendah. Dalam kenyataannya, secara umum ada beberapa factor lain, selain nilai risk exposure, yang harus diperhitungkan pada saat menentukan prioritas resiko. Kemungkinan resiko yang muncul dalam suatu proyek terlepas dari dampak yang dihasilkan dan tindakan yang diambil dari resiko tersebut (gambar 1).



Gambar 1. Hubungan kemungkinan resiko yang muncul dan dampat kerugian yang dihasilkan

Jika kemungkinan kejadian resiko kecil dan dampak kerugian kecil, kita dapat melakukan tindakan untuk menerima resiko, tetapi jika kemungkinan resiko yang muncul tinggi dan memilki dampak kerungian yang tinggi, maka kita dapat mengindari resiko walaupun kadang resiko yang besar dapat menghasilkan keuntungan yang besar.

IV. METODELOGI MANAJEMEN RESIKO PERANGKAT LUNAK

Beberapa pendekatan dari arsitektur dan model manajemen resiko perangkat lunak diperkenalkan dan dikembangkan oleh beberapa peneliti. Kebanyakan pendekatan yang ada digunakan untuk memperkirakan resiko yang muncul selama tahapan dari pembangunan perangkat lunak. Manajemen resiko perangkat lunak dilakukan oleh manajemen di proyek, sebelum suatu perangkat lunak dianalisis dan dan dirancang. Beberapa pendekatan manajemen resiko telah dikembangkan oleh beberapa peneliti antara :

1. Model Manajemen Resiko Boehm (Boehm, B. W, 1991; Boehm, B. W & Bose, P, 1994).
2. Deursen, A. and Model Manajemen Resiko SEI [Williams, R.C, et al. 1999]
3. Model Manajemen Resiko Hall [Hall, E. M, 1998]
4. Software Just-In-Time [Karolak, D. 1996; Karolak, D. 1998)
5. Model Riskit [Kontio, J. 2001]

Kesimpulan pendekatan penelitian yang telah dilakukan di atas tidak dapat kita perbandingkan antara satu dengan yang lainnya, karena setiap pendekatan dari manajemen resiko dibangun didasakan kondisi dan sudut pandang peneliti mengenai manajemen resiko, akibatnya berdampak pada cara peneliti dalam memecahkan masalah (Yasmi afrizal & Agus Harjoko, 2009). Contoh : Model manajemen resiko Hall mengembangkan pemecahan resiko berdasarkan kemampuan model yang dikembangkan. Sedangkan Model Manajemen Resiko Boehm mengunakan metode *Win-Win solution* yang dikembangkan dari model proses pada pembangunan perangkat lunak yang disebut disebut dengan spiral model, Boehm memasukan teknik resiko untuk mengevaluasi terhadap setiap langkah dalam pembangunan perangkat lunak.

V. KESIMPULAN

Pada paper ini, peneliti memcoba untuk menjelaskan secara keseluruhan konsep fundamental dari metodologi manajemen resiko dalam area rekayasa perangkat lunak. Manajemen resiko merupakan bidang keilmuan yang masih cukup baru, tetapi bidang ini sudah banyak mendapat perhatian peneliti, mekipun masih terdapat kekurangan dalam memahami area tersebut dalam praktisi rekayasa perangkat lunak.

Tinjauan paper ini membantu akademisi, peneliti, dan praktisi untuk dengan cepat memahami dengan prinsip dasar dalam bidang manajemen resiko dan memutuskan metodologi apa yang mereka gunakan dalam proyek. Bagaimanapun, perlu dicatat bahwa paper itu diarahkan untuk menyediakan satu kesimpulan dari pentingnya manajemen resiko untuk menghindari kegagalan proyek perangkat lunak sesuai dengan referensi yang ada.

Banyak dari pendekatan penelitian dalam pembahasan ini, dibatasi dengan dukungan data empiris. Hal ini merupakan yang penting bagi penelitian manajemen resiko pada masa yang akan depan. Selain itu fokus penelitian ke depan harus dapat melakukan evaluasi dari masing–masing metodelogi. Karena kebanyakan dari pendekatan yang diusulkan, memerlukan studi kasus yang nyata untuk melakukan penilaian dari efektivitas dari masing-masing metodologi .

DAFTAR PUSTAKA

Boehm, B. W. 1988. A Spiral Model of Software Development and Enhancement, Computer, May, 61-72.

Boehm, B. W. 1991. Software Risk Management: Principles and Practices, IEEE Software, 8(1):32-41 .

Boehm, B. W. and Bose, P. 1994. A Collaborative Spiral Software Process Model Based on Theory W, Proceedings of the 1994 International Conference on Software Process, IEEE Computer Society, Washington.

Deursen, A. and Kuipers, T. 2003. Source-Based Software Risk Assessment. Proceedings of the 2003 International Conference on Software Maintenance, Amsterdam, Netherlands.

Hall, E. M. 1998. Managing Risk: Methods for Software Systems Development, Addison-Wesley, Reading, U.K.

Higuera R. P. and Haimes, Y. Y. 1996. Software Risk Management, Technical Report, Report # CMU/SEI-96-TR-012, Software Engineering Institute, Carnegie Mellon University, Pittsburgh, PA, USA.

Karolak, D. 1996. Software Engineering Risk Management, IEEE Computer Society Press, Los Alamitos, CA, USA.

Karolak, D. 1998. Software Engineering Risk and Just-In-Time Development, International Journal of Computer Science and Information Management, 1

Kontio, J. 2001. Software Engineering Risk Management: A Method, Improvement Framework, and Empirical Evaluation. Ph.D. Thesis, Department of Computer Science and Engineering, Hensinki University of Technology, Finland, 2001.

Yasmi Afrizal and Agus Harjoko. 2009. Perangkat Lunak JIT (Just In Time) untuk Memprediksi Resiko Proyek Perangkat Lunak**,** Jurnal Sistem Informasi Universitas Kristen Maranatha, Volume 4 Nomor 1 Maret 2009