

# Mengapa Proyek Perangkat Lunak Gagal (Penerapan Manajemen Resiko dalam Proyek Perangkat Lunak) *Why Software Project Fails (Applying Risk Management in Software Project)*

Yasmi Afrizal

Universitas Komputer Indonesia, Jl. Dipati Ukur No 112-116, Bandung, Telp:(022) 2504119

yasmi\_afrizal@yahoo.co.id

**Abstract-** The greater and problems complex of software development project enables the event of failure. Risk management is an approach to guide minimization or lessens the failure impact. This paper is explain risk management model base that probe event risk in developing software. Risk management has step of solution consisted of by identification, analysis, mitigation strategy risk. This paper also very useful to assist academicians, researcher and practitioner to be more understanding risk management as one part of the software engineering.

**Keywords :** Project failure, risk management, risk analysis, software engineering

## I. PENDAHULUAN

Pertumbuhan Industri perangkat lunak yang cepat, menuntut pengembang untuk semakin berkompetisi dalam mengambil lompatan-lompatan besar dalam menghasilkan produk perangkat lunak yang berkualitas. Kesuksesan dalam membangun perangkat lunak tidak lepas dari bagaimana pengembang mengelola resiko yang hadapi seperti keterlambatan penerimaan produk atau pengeluaran biaya yang melebihi anggaran yang direncanakan. Permasalahan tersebut lebih disebabkan 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 [1]. Solusi pemecahan masalah dilakukan dengan meminimalkan segala macam ketidakjelasan yang muncul di awal proyek dan melakukan evaluasi terhadap pemecahan tersebut

Suatu resiko yang muncul dalam pengembangan perangkat lunak mungkin memiliki prioritas perhatian yang lebih dibandingkan resiko lainnya. Prioritas sebuah resiko sangat bergantung pada sifat resiko dan dampak yang ditimbulkan jika resiko tersebut terjadi. Untuk mengurangi bahaya tersebut, maka harus ada jaminan untuk meminimalkan resiko

atau paling tidak mendistribusikannya selama pengembangan atau idealnya resiko tersebut harus dihindari.

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.

## II. ANALISIS RESIKO

Barry Boehm merupakan peneliti pertama yang berhasil membawa manajemen resiko ke dalam area rekayasa perangkat lunak ([1]-[2]). Sumbangan utama dari Boehm adalah mengenalkan teknik pengukuran resiko yang kemudian dimasukkan kedalam model proses spiral [1]. Manajemen resiko Boehm meletakkan standar pengukuran resiko yang disebut *risk exposure* dengan bentuk :

Risk Exposure =

$$\text{Probability(Outcome)} * \text{Loss(Outcome)} \dots\dots (1)$$

Probabilitas resiko yang mucul sering disebut dengan *risk likelihood*, dampak yang akan terjadi jika resiko terjadi dikenal dengan *risk impact* dan tingkat kepentingan atau perhatian dari terhadap resiko disebut *risk exposure*. Idealnya *risk likelihood* dan *risk impact* harus diestimasi agar memiliki nilai yang minimal dengan menggunakan nilai probabilitas. Dikarenakan *risk likelihood* dan *risk impact* bernilai probabilitas, maka kita sulit untuk membandingkan antara satu dengan lainnya karena bersifat subyektif dan sulit dihitung. Untuk mengatasi hal ini, maka diperlukan bentuk pengukuran lain yang bersifat kuantitatif. Penggunaan pengukuran kuantitatif akan menghasilkan pemahaman yang mendalam dan bermanfaat jika terjadi suatu permasalahan.

Para manajer proyek perangkat lunak dapat menggunakan metode penilaian yang sederhana untuk menghasilkan pengukuran yang kuantitatif pada saat mengevaluasi masing-masing resiko. Mereka dapat 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 kemungkinan terjadinya kecil maka akan diberi nilai 1.

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

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

TABEL 1  
CONTOH EVALUASI NILAI RISK EXPOSURE

Resiko	Keterangan	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 ketidakefisienan dalam desain.	1	10	10

### III. STRATEGI PENGURANGAN RESIKO

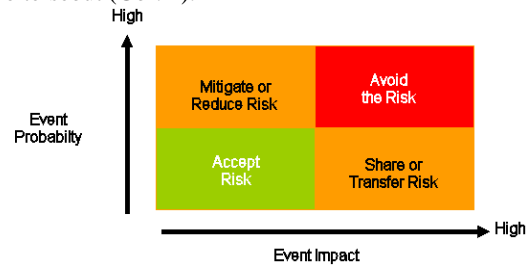
Segala usaha akan kita lakukan untuk mengurangi sebuah *risk exposure* atau rencana kontigensi akan berhubungan dengan biaya yang berkaitan dengan usaha. Merupakan hal yang penting untuk menjamin usaha yang dilakukan sudah yang paling efektif. Pengelolaan terhadap pengurangan resiko melibatkan penggunaan dua strategi:

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

Estimasi nilai *likelihood* 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 digunakan untuk menilai resiko perlu mendapat beberapa perhatian. Hasil evaluasi pada Tabel 1, memperlihatkan bahwa resiko R5 tidak dua kali lebih penting dibandingkan R6. Kejadian ini memperingatkan kita untuk tidak bias menterjemahkan nilai risk exposure secara kuantitatif, disebabkan nilai tersebut didasarkan pada metode penilaian yang non-cardinal. Pada kasus kedua, nilai exposure yang terlalu besar akan mampu untuk membedakan antara resiko yang ada. Akan tetapi risk exposure akan memungkinkan kita untuk memperoleh suatu prioritas (ranking) resiko mana yang kan mendapat perhatian. Berdasarkan kumpulan resiko yang ada pada Tabel 1, R3 dan R4 merupakan resiko yang paling penting dan kita dapat

mengklasifikasikannya sebagai *high risk*. Tingkat kepentingan yang berbeda juga diperlihatkan pada skor exposure R2. R2 dan R5 yang mempunyai skor yang hampir sama yaitu dikelompokkan pada resiko dengan prioritas medium. Dua resiko lainnya, R1 dan R6 mempunyai nilai exposure yang rendah sehingga dapat dikelompokkan pada prioritas rendah. Dalam kenyataannya, secara umum ada beberapa faktor lain, selain nilai risk exposure, yang harus diperhitungkan pada saat menentukan prioritas resiko.

Kemungkinan resiko yang muncul dalam suatu proyek terlepas dari dampak yang dan tindakan yang diambil dari resiko tersebut (Gbr. 1).



Gbr. 1 Hubungan kemungkinan resiko yang muncul dan dampak kerugian yang dihasilkan

Berdasarkan Gbr. 1, kita dapat menerima resiko (*accept risk*) apabila kemungkinan resiko yang muncul (*event probability*) memiliki nilai yang kecil dan dampak kerugian yang kecil, akan tetapi jika kemungkinan resiko yang muncul bernilai tinggi dan memiliki dampak kerugian yang tinggi, maka kita dapat menghindari resiko walaupun kadang resiko yang besar dapat menghasilkan keuntungan yang besar.

### IV. METODOLOGI 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 lain:

1. Model Manajemen Resiko Boehm ([2] – [3]).
2. Deursen, A. and Model Manajemen Resiko SEI [6]
3. Model Manajemen Resiko Hall [5]
4. Software Just-In-Time ([7], [8])
5. Model Riskit [9]

Kesimpulan pendekatan penelitian yang telah dilakukan di atas tidak dapat kita perbandingkan antara satu dengan yang lainnya, karena setiap pendekatan dari manajemen resiko dibangun didasarkan kondisi dan sudut pandang peneliti mengenai manajemen resiko, akibatnya berdampak pada cara peneliti dalam memecahkan masalah [10]. Contoh: Model manajemen resiko Hall mengembangkan pemecahan resiko berdasarkan kemampuan model yang dikembangkan. Sedangkan Model Manajemen Resiko Boehm menggunakan metode *Win-Win solution* yang dikembangkan dari model

proses pada pembangunan perangkat lunak yang disebut dengan spiral model, Boehm memasukan teknik resiko untuk mengevaluasi terhadap setiap langkah dalam pembangunan perangkat lunak.

## V. PENUTUP

Pada paper ini, peneliti mencoba 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 metodologi. Karena kebanyakan dari pendekatan yang diusulkan,

memerlukan studi kasus yang nyata untuk melakukan penilaian dari efektivitas dari masing-masing metodologi .

## DAFTAR PUSTAKA

- [1] Boehm, B. W. 1988. "A Spiral Model of Software Development and Enhancement", *Computer*, May, 61-72.
- [2] Boehm, B. W. 1991. "Software Risk Management: Principles and Practices", *IEEE Software*, 8(1):32-41 .
- [3] 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.
- [4] Deursen, A. and Kuipers, T. 2003. "Source-Based Software Risk Assessment", *Proceedings of the 2003 International Conference on Software Maintenance*, Amsterdam, Netherlands.
- [5] Hall, E. M. 1998. *Managing Risk: Methods for Software Systems Development*, Addison-Wesley, Reading, U.K.
- [6] 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.
- [7] Karolak, D. 1996. *Software Engineering Risk Management*, IEEE Computer Society Press, Los Alamitos, CA, USA.
- [8] Karolak, D. 1998. "Software Engineering Risk and Just-In-Time Development", *International Journal of Computer Science and Information Management*, 1 .
- [9] 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.
- [10] 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.