MANAJEMEN RESIKO DALAM PENGEMBANGAN PERANGKAT LUNAK

MANAJEMEN RESIKO

Manajemen resiko adalah proses pengukuran atau penilaian resiko serta pengembangan strategi pengelolaannya. Strategi yang dapat diambil antara lain adalah memindahkan resiko kepada pihak lain, menghindari resiko, mengurangi efek negatif resiko, dan menampung sebagian atau semua konsekuensi resiko tertentu. Manajemen resiko tradisional terfokus pada resiko-resiko yang timbul oleh penyebab fisik atau legal (seperti bencana alam atau kebakaran, kematian serta tuntutan hokum). (Wikipedia)
Manajemen resiko adalah rangkaian langkah-langkah yang membantu suatu perangkat lunak untuk memahami dan mengatur ketidak pastian (Roger S. Pressman).

Rekayasa Perangkat lunak
SOFTWARE ENGINGEERING

Dalam rekayasa perangkat lunak tahap awal adalah pendefinisian tentang rekayasa system apa yang akan dibuat. Diperlukan proses perencanaan dan analisis kebutuhan. Setelah pendefinisian tahap selanjutnya adalah pengembangan, dalam tahap ini adalah bagaimana produk yang telah didefinisikan dengan jelas kemudian akan mulai diimplementasikan. Maka pada proses pengembangan ini akan dilakukan design software, kemudian mengenerate koding-koding pembangun program, hingga program siap dites kebenarannya.
Proses rekayasa perangkat lunak adalah proses yang terus berulang, karena karakteristik perangkat lunak yang membutuhkan pemeliharaan dan continue development agar perangkat lunak tidak kadarluasa. Dalam proses pemelihataan kita melakukan koreksi kesalahan, adaptasi kebutuhan, peningkatan kemampuan atau fungsi dan bentuk pencegahan lainnya agar perangkat lunak tersebut tidak kadarluasa.
Penyebab kegagalan rekayasa perangkat lunak adalah :

- Perencanaan yang tidak realistik, terlalu optimis dalam perhitungan.
- sistem pemantauan kerja yang tidak berjalan dengan seharusnya.
- Perubahan kebutuhan.
- Resiko-resiko lainnya

METODOLOGI PROSES

Metodologi adalah cara sistematis atau cara yang didefinisikan secara jelas untuk mencapai tujuan akhir. Juga merupakan sebuah system tata tertib dalam berpikir atau bertindak. Metodologi yang baik adalah sebuah peta atau jalan yang menjadi panduan untuk menemukan jalan yang tepat untuk mencapai tujuan.
Metodologi dalam proses rekayasa perangkat lunak masih menjadi objek penelitian, tapi sekarang ada banyak model umum atau paradigma yang berbeda dari pengembangan perangkat lunak, yaitu :

- Pendekatan Waterfall
- Pengembangan secara evolusioner
- Trasformasi Formal
- Reuse

MODEL WATERFALL

Pertama kali diperkenalkan oleh Winston Royce tahun1970. Model ini merupakan model klasik yang sederhana dengan aliran system yang linear. **Output dari setiap tahap menjadi input bagi tahap berikutnya.** Model ini melibatkan SQA (Software Quality Assurance) dengan tahapan yang setiap tahapannya dilakukan verivikasi dan testing. Tahapa-tahapannya adalah :

Requirement

Pada tahapan ini dilakukan analisa kebutuhan, kemudian diverivikasi oleh klien dan tim SQA. Jasa, kendala, tujuan dihasilkan dari konsultasi dengan pengguna system, kemudian semuanya dibuat dalam bentuk yang dapat dimengerti oleh staf pengembang.

Specification

Jika dokumentasi spesifikasi disetujui oleh klien, maka dokumen tersebut merupakan kontrak kerja antara klien dengan pengembang software. Selanjutnya adalah merencanakan jadwal pengembangan software. Jika disetujui, maka tahap desain akan dilakukan.

Desain

Pada proses desain, system membagi kebutuhan-kebutuhan untuk menghasilkan sebuah arsitektur system keseluruhan. Yang dilakukan pada tahap ini adalah :

- Dekomposisi modul system
- Rancangan masukan dan keluaran
- Penetapan struktur data
- Penetapan prosedur kerja
- Penetapan formula pengolahan data

Implementasi dan Integrasi

Adalah tahap dilakukannya konversi dari hasil rancangan (spesifikasi program) menjadi source code yang terpisah masih dalam bentuk modul, kemudian setiap modul akan diuji terlebih dahulu sebelum digabungkan dengan modul lainnya, kemudian system yang terbentuk dari proses integrasi akan diuji untuk meyakinkan persyaratan perangkat lunak terpenuhi.

Operation Mode dan Retriment

Merupakan tahap terpanjang. Sistem dipasang dan digunakan. Dilakukan pemeliharaan termasuk pembetulan kesalahan yang tidak ditemukan pada langkah sebelumnya.

Jika tim SQA tidak menyetujui maka tahapan pada model waterfall tidak dianggap selesai. Jika terdapat ketidaksesuaian dengan dokumen tahap sebelumnya, maka proses harus kembali ke tahap sebelumnya untuk penyesuaian dan peninjauan ulang. Model ini mengijinkan untuk kembali ketahap sebelumnya.

Permasalahan pada linear model atau waterfall adalah :

- Penanganan perubahan pada saat proses sedang berlangsung menjadi lebih sulit.
- Semua kebutuhan sudah terefenisi sejak awal.
- Software yang diberikan adalah versi terakhir dari setiap tahap, perubahan dalam proses biasanya tidak dilakukan.
- Blocking state

MODEL PROTOTYPING

Terdiri atas tiga bentuk model proses yaitu :

- Diatas kertas berbasis komputer menggambarkan interaksi manusia.
- Working prototype : mengimplementasikan sebagian fungsi perangkat lunak.
- Program jadi : melakukan sebagian atau keseluruhan fungsi yang akan dilakukan. Ada beberapa feature yang belum dikembangkan.

Tahapan proses prototyping adalah :

- Requirement : pengumpulan kebutuhan dan perbaikan
- Quick design
- Pembentukan Prototyping
- Evaluasi Pelanggan
- Perbaikan Prototyping

* Iterasi dilakukan terus menerus mulai dari tapa Quick Design hingga perbaikan Prototype sampai didapat produk akhir.

Permasalahan pada model Prototyping adalah :

- Pelanggan yang melihat working version kemungkinan tidak menyadari bahwa mungkin prototype dibuat terburu-buru dengan rancangan yang disusun tidak terstruktur.
- Pembuatan kadang membuat implementasi sembarang karena ingin working version bekerja lebih cepat.

MODEL SPIRAL BOEHM

Model ini diusulkan oleh Boehm pada tahun 1988 sebagai pendekatan alternative dari model waterfall. Model ini menggunakan fitur yang ada pada model waterfall dan prototype. Setiap tahapan model ini selalu dilakukan risk analisys dan verivikasi atau testing. ***Model ini memiliki empat aktivitas yaitu :***

- Determine objectives : penentuan tujuan, alternative dan batasan dalam proses.
- Risk Analysis : analisa alternatif terhadap resiko yang mungkin terjadi.
- Engineering/develop : pengembangan produk.
- Plant next phase : penentuan rencana-rencana untuk tahap selanjutnya.

Perbedaan yang mendasar antara model spiral dengan model lainnya adalah bahwa model spiral dengan eksplisit menyadari resiko-resiko yang ada. ***Risk Analysis yang dilakukan model spiral*** ini adalah :

- Project Risk : Hal-hal yang mempengaruhi tahap proyek, contoh : kekurangan sumber daya.
- Technical Risk : Hal-hal yang mempengaruhi tahap actual, contoh :; personil tidak terlatih ditahap tersebut.
- Bussiness Risk : Hal-hal yang mempengaruhi keinginan perusahaan untuk membuat software, contoh : software ternyata tidak dibutuhkan lagi.

Prioritas resiko dapat dikategorikan sebagai berikut :

- Catastrophic (luar biasa), contoh : penurunan kualitas yang luar biasa, biaya yang tidak terkontrol.
- Critical (kritis), contoh : tidak tepat waktu, biaya diluar perkiraan.
- Marginal (ringan), contoh : penjadwalan yang terlambat.
- Negligible (tidak berarti), contoh : penggunaan waktu proyek yang tidak optimal.

Model spiral sangat baik digunakan untuk pengembangan sistem yang besar, karena sistem ini meminimalisasi resiko lewat mekanisme yang baik. kelemahan sistem ini ada pada pengontrollannya dan belum banyak cerita sukses tentang penggunaan model ini.

MODEL INCREMENTAL

Model ini menerapkan rekayasa perangkat lunak perbagian, hingga menghasilkan perangkat lunak yang lengkap. Proses membangun berhenti jika produk telah mencapai seluruh fungsi yang diharapkan. Pada awal tahapan dilakukan penentuan, kebutuhan dan spesifikasi. Kemudian dilakukan perancangan arsitektur software yang terbuka, agar dapat diterapkan pembangunan per-bagian pada tahapan selanjutnya.

Tahapan pada model incremental adalah requirement, specification, architektur design, kemudian tahapan membangun tiap bagian secara berurutan. Setiap bagian yang telah selesai testing, maka akan dikirim kepemakai untuk lansung dapat digunakan. Pada model incremental, tahapan requirement, specification dan architektur design harus dikerjakan terlebih dahulu sebelum tahapan pembagian tiap modul.

Kembali lagi kepada **MANAJEMEN RESIKO PADA REKAYASA PERANGKAT LUNAK**

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.

Mengapa manajemen resiko itu penting? Sikap orang ketika menghadapi resiko berbeda-beda. Ada orang yang berusaha untuk menghindari resiko, namun ada juga yang sebaliknya sangat senang menghadapi resiko sementara yang lain mungkin tidak terpengaruh dengan adanya resiko. Pemahaman atas sikap orang terhadap resiko ini dapat membantu untuk mengerti betapa resiko itu penting untuk ditangani dengan baik.

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.

Resiko dalam perangkat lunak memiliki dua karakteristik:

- Uncertainty : tidak ada resiko yang 100% pasti muncul.
- Loss : resiko berimbas pada kehilangan.

Dan resiko memiliki tiga kategori:

- Resiko proyek : berefek pada perencanaan proyek.
- Resiko teknikal : berefek pada kualitas dan waktu pembuatan perangkat lunak.
- Resiko bisnis : berefek pada nilai jual produk

Contoh : Seorang programmer yang sangat pintar keluar. Resiko yang mana?

1. **Analisa resiko**

Setelah melakukan identifikasi resiko, maka tahap berikutnya adalah pengukuran resiko dengan cara melihat potensial terjadinya seberapa besar severity (kerusakan) dan probabilitas terjadinya risiko tersebut. Penentuan probabilitas terjadinya suatu event sangatlah subyektif dan lebih berdasarkan nalar dan pengalaman. Beberapa risiko memang mudah untuk diukur, namun sangatlah sulit untuk memastikan probabilitas suatu kejadian yang sangat jarang terjadi. Sehingga, pada tahap ini sangtalah penting untuk menentukan dugaan yang terbaik supaya nantinya kita dapat memprioritaskan dengan baik dalam implementasi perencanaan manajemen risiko. Kesulitan dalam pengukuran risiko adalah menentukan kemungkinan terjadi suatu risiko karena informasi statistik tidak selalu tersedia untuk beberapa risiko tertentu. Selain itu, mengevaluasi dampak severity (kerusakan) seringkali cukup sulit untuk asset immateriil. Dampak adalah efek biaya, waktu dan kualitas yang dihasilkan suatu resiko.

Dampak	**Biaya**	**Waktu**	**Kualitas**
Sangat rendah	Dana mencukupi	Agak menyimpang dari target	Kualitas agak berkurang namun masih dapat digunakan
Rendah	Membutuhkan dana tambahan	Agak menyimpang dari target	Gagal untuk memenuhi janji pada stakeholder
Sedang	Membutuhkan dana tambahan	Penundaan berdampak terhadap stakeholder	Beberapa fungsi tidak dapat dimanfaatkan
Tinggi	Membutuhkan dana tambahan yang signifikan	Gagal memenuhi deadline	Gagal untuk memenuhi kebutuhan banyak stakeholder
Sangat tinggi	Membutuhkan dana tambahan yang substansial	Penundaan merusak proyek	Proyek tidak efektif dan tidak berguna

Setelah mengetahui probabilitas dan dampak dari suatu resiko, maka kita dapat mengetahui potensi suatu resiko. Untuk mengukur bobot resiko kita dapat menggunakan skala dari 1 – 5 sebagai berikut seperti yang disarankan oleh JISC InfoNet:

Skala	**Probabilitas**	**Dampak**
Sangat rendah	Hampir tidak mungkin terjadi	Dampak kecil
Rendah	Kadang terjadi	Dampak kecill pada biaya,

		waktu dan kualitas
Sedang	Mungkin tidak terjadi	Dampak sedang pada biaya, waktu dan kualitas
Tinggi	Sangat mungkin terjadi	Dampak substansial pada biaya, waktu dan kualitas
Sangat tinggi	Hampir pasti terjadi	Mengancam kesuksesan proyek

Setelah resiko yang dapat mempengaruhi pengembangan teridentifikasi maka diperlukan cara untuk menentukan tingkat kepentingan dari masing-masing resiko. Beberapa resiko secara relatif tidak terlalu fatal (misal resiko keterlambatan penyerahan dokumentasi) sedangkan beberapa resiko lainnya berdampak besar. (misal resiko keterlambatan penyerahan software). Beberapa resiko sering terjadi (salah satu anggota tim sakit sehingga tidak bisa bekerja selama beberapa hari). Sementara itu resiko lainnya jarang terjadi (misal kerusakan perangkat keras yang dapat mengakibatkan sebagian program hilang).
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.* Risk value dapat dihitung dengan formula :

EXPOSURE = RISK LIKELIHOOD X RISK IMPACT

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 :

- Biaya yang diakibatkan keterlambatan penyerahan atas jadwal yang sudah ditentukan
- Biaya yang berlebihan dikarenakan harus menambah sumber daya atau dikarenakan mempergunakan sumber daya yang lebih mahal

2. Prioritas resiko

PENGELOLAAN RESIKO MELIBATKAN PENGGUNAAN DUA STRATEGI:

RISK EXPOSURE DAPAT DIKURANGI DENGAN MENGURANGI LIKEHOOD ATAU IMPACT
PEMBUATAN RENCANA KONTINGENSI BERKAITAN DENGAN KEMUNGKINAN RESIKO YANG AKAN TERJADI.
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.
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.
Kepercayaan terhadap penilaian resiko

Beberapa penilaian risk exposure relative kurang. Untuk diperlukan investigasi lebih lanjut sebelum tindakan diambil.

Penggabungan resiko

Beberapa resiko saling bergantung dengan lainnya. Dalam hal ini maka beberapa resiko tersebut perlu dianggap sebagai satu resiko.

Jumlah resiko

Perlu batas jumlah resiko yang dapat dipertimbangkan secara efektif dan dapat diambil tindakannya oleh seorang manajer proyek. Untuk itu perlu dibatasi ukuran daftar prioritas.

Biaya tindakan

Beberapa resiko, yang suatu saat dapat dikenali, dapat dikurangi atau dicegah segera dengan biaya atau usaha yang sedikit tanpa menganggap nilai resikonya. Untuk resiko lainnya perlu dilakukan perbandingan antara biaya yang diperlukan dengan benefit yang diperoleh dengan mengurangi resiko tersebut. Satu metode untuk melaksanakan perhitungan ini adalah dengan menghitung risk reduction leverage (RRL) dengan mempergunakan persamaan sebagai berikut:

$$\mathbf{RRL = RE_{before} - RE_{after}}$$

Risk reduction cost

RE_{before} adalah nilai risk exposure semula, RE_{after} adalah nilai risk exposure yang diharapkan setelah diambil tindakan dan risk education cost adalah biaya untuk implementasi tindakan pengurangan resiko. Risk reduction cost harus dinyatakan dengan unit yang sama dengan nilai resiko yaitu nilai moneter yang diperlukan atau dengan nilai skor. Jika nilai yang diharapkan ternyata lebih besar maka RRL yang lebih besar memperlihatkan bahwa kita perlu berharap untuk meningkatkan rencana pengurangan resiko disebabkan reduksi risk exposure yang diharapkan lebih besar dibandingkan dengan biaya rencana.

3. Pengelolaan resiko

Jenis-jenis cara mengelola resiko :

1. *Risk Avoidance*

Yaitu memutuskan untuk tidak melakukan aktivitas yang mengandung resiko sama sekali. Dalam memutuskan untuk melakukannya, maka harus dipertimbangkan potensial keuntungan dan potensial kerugian yang dihasilkan oleh suatu aktivita

2. *Risk Reduction*

Risk reduction atau disebut juga *risk mitigation* yaitu merupakan metode yang mengurangi kemungkinan terjadinya suatu resiko ataupun mengurangi dampak kerusakan yang dihasilkan oleh suatu resiKo.

3. *Risk Transfer*

Yaitu memindahkan resiko pada pihak lain, umumnya melalui suatu kontrak (asuransi) maupun hedging.

4. *Risk Deferral*

Dampak suatu resiko tidak selalu konstan. *Risk deferral* meliputi menunda aspek suatu proyek hingga saat dimana probabilitas terjadinya resiko tersebut kecil.

5. *Risk Retention*

Walaupun resiko tertentu dapat dihilangkan dengan cara mengurangi maupun mentransfernya, namun beberapa resiko harus tetap diterima sebagai bagian penting dari aktivitas.

Penanganan resiko:

1. *High probability, high impact:* resiko jenis ini umumnya dihindari ataupun ditransfer.
2. *Low probability, high impact:* respon paling tepat untuk tipe resiko ini adalah dihindari. Dan jika masih terjadi, maka lakukan mitigasi resiko serta kembangkan *contingency plan.*
3. *High probability, low impact:* mitigasi resiko dan kembangkan *contingency plan.*
4. *Low probability, low impact:* efek dari resiko ini dapat dikurangi, namun biayanya dapat saja melebihi dampak yang dihasilkan. Dalam kasus ini mungkin lebih baik untuk menerima efek dari resiko tersebut.

Contigency plan

Untuk resiko yang mungkin terjadi maka perlu dipersiapkan *contingency plan*seandainya benar-benar terjadi. *Contigency plan* haruslah sesuai dengan proposional terhadap dampak resiko tersebut. Dalam banyak kasus seringkali lebih efisien untuk mengalokasikan sejumlah sumber daya untuk mengurangi resiko dibandingkan mengembangkan *contingency plan* yang jika diimplementasikan akan lebih mahal. Namun beberapa skenario memang membutuhkan full *contingency plan,* tergantung pada proyeknya. Namun jangan sampai tertukar antara *contingency planning* dengan re-planning normal yang memang dibutuhkan karena adanya perubahan dalam proyek yang berjalan.

Tabel resiko proyek software dan strategi mengurangi resiko

Resiko	**Teknik mengurangi resiko**
Kegagalan pada personil	· Memperkerjakan staf yang handal · Job matching · Membangun tim · Mengadakan pelatihan dan peningkatan karir · Membuat jadwal lebih awal bagi personil utama
Estimasi biaya dan waktu yang tidak realistis	· Membuat beberapa estimasi · Desain untuk biaya · Meningkatkan pengembangan · Merekam dan menganalisa proyek sebelumnya · Standarisasi metode
Mengembangkan fungsi software yang salah	· Evaluasi proyek ditingkatkan · Buat metode spesifikasi yang formal · Survey pengguna · Buat prototype · Buat user manual lebih awal

Mengembangkan antarmuka pengguna yang salah	· Membuat prototype · Analisis tugas · Keterlibatan pengguna
Gold plating	· Mengurangi kebutuhan · Membuat prototype · Analisis biaya manfaat · Desain biaya
Terlambat untuk mengubah kebutuhan	· Mengubah prosedur kendali · Membatasi perubahan yang terlalu banyak · Meningkatkan prototype · Meningkatkan pengembangan (akibat perubahan)
Kegagalan pada komponen yang disuplai pihak eksternal	· Melakukan benchmarking · Inspeksi · Spesifikasi formal · Kontrak perjanjian · Prosedur dan sertifikasi jaminan kualitas
Kegagalan menjalankan tugas eksternal	· Prosedur jaminan kualitas · Desain / prototype yang kompetitif · Membangun tim · Kontrak insentif
Kegagalan kinerja real-time	· Simulasi · Benchmarking · Prototipe · Tuning · Analisis teknis
Pengembangnya terlalu sulit secara teknis	· Analisa teknis · Analisis biaya manfaat · Prototipe · Melatih dan mengembangkan staf

4. Implementasi manajemen resiko

Setelah memilih respon yang akan digunakan untuk menangani resiko, maka saatnya untuk mengimplementasikan metode yang telah direncanakan tersebut.

1. Monitoring resiko

Mengidentifikasi, menganalisa dan merencanakan suatu resiko merupakan bagian penting dalam perencanaan suatu proyek. Namun, manajemen resiko tidaklah berhenti sampai disana saja. Praktek, pengalaman dan terjadinya kerugian akan membutuhkan suatu perubahan dalam rencana dan keputusan mengenai penanganan suatu resiko. Sangatlah penting untuk selalu memonitor proses dari awal mulai dari identifikasi resiko dan pengukuran resiko untuk mengetahui keefektifitas respon yang telah dipilih dan untuk mengidentifikasi adanya resiko yang baru maupun berubah. Sehingga, ketika

suatu resiko terjadi maka respon yang dipilih akan sesuai dan diimplementasikan secara efektif.

TAHAP INTEGRASI MANAJEMEN RISIKO DALAM SDLC PADA NIST 800-30

1. Inisiasi. Pada tahap ini kebutuhan sistem IT dinyatakan dengan tujuan dan ruang lingkup yang ditetapkan. risiko yang di identifikasi digunakan untuk mendukung syarat pengembangan sistem, termasuk persyaratan keamanan, dan konsep keamanan operasi (strategi).

2. Pengembangan atau akuisisi. Pada tahap ini melakukan perancangan, pembelian atau pengembangan sistem IT. Selain itu juga dilakukan identifikasi/analisis resiko terhadap keamanan sistem IT yang mengganggu proses bisnis suatu perusahaan selama pengembangan sistem.

3. Implementasi. Pada tahap ini melakukan penerapan sistem IT dalam suatu organisasi yang mana fitur keamanan harus diaktifkan dan diuji lagi agar resiko yang timbul semakin tidak ada.

4. Operasi / pemeliharaan. Fase ini berkaitan dengan kegiatan pemeliharaan, dimana perubahan dan modifikasi pada hardware sistem yang digunakan. Kegiatan manajemen risiko dilakukan secara berkala sperti reakreditasi (reauthorization) atau perubahan tampilan yang lebih baru pada sistem IT.

5. Disposal. Dalam tahap ini penghapusan sistem lama dan data dipindahkan ke sistem yang baru. Aktivitas manajemen risiko meliputi pembuangan hardware dan software dilakukan secara aman sehingga data rahasia benar-benar tidak tersimpan lagi pada hardware/software yang lama dan setiap migrasi data dilakukan dengan cara yang aman dan sistematis.

Strategi Pendekatan Manajemen Resiko Dalam Pengembangan Sistem Informasi

1. PENDAHULUAN

Resiko adalah suatu umpan balik negatif yang timbul dari suatu kegiatan dengan tingkat probabilitas berbeda untuk setiap kegiatan[4]. Pada dasarnya resiko dari suatu kegiatan tidak dapat dihilangkan akan tetapi dapat diperkecil dampaknya terhadap hasil suatu kegiatan. Proses menganalisa serta memperkirakan timbulnya suatu resiko dalam suatu kegiatan disebut sebagai manajemen resiko.

Seiring dengan berkembangnya teknologi informasi yang bergerak sangat cepat dewasa ini, pengembangan unit usaha yang berupaya menerapkan sistem informasi dalam organisasinya telah menjadi kebutuhan dasar dan semakin meningkat dari tahun ke tahun. Akan tetapi pola pembangunan sistem informasi yang mengindahkan faktor resiko telah menyebabkan beberapa organisasi mengalami kegagalan menerapkan teknologi informasi tersebut, atau meningkatnya nilai investasi dari plafon yang seharusnya, hal ini juga dapat menghambat proses pencapaian misi organisasi.

Pada dasarnya, faktor resiko dalam suatu perencanaan sistem informasi, dapat diklasifikasikan ke dalam 4 kategori resiko [3], yaitu :

a. Catastrophic (Bencana)

b. Critical (Kritis)

c. Marginal (kecil)

d. Negligible (dapat diabaikan)

Adapun pengaruh atau dampak yang ditimbulkan terhadap suatu proyek sistem informasi dapat berpengaruh kepada a) nilai unjuk kerja dari sistem yang dikembangkan, b) biaya yang dikeluarkan oleh suatu organisasi yang mengembangkan teknologi informasi, c) dukungan pihak manajemen terhadap pengembangan teknologi informasi, dan d) skedul waktu penerapan pengembangan teknologi informasi.[1]

Suatu resiko perlu didefinisikan dalam suatu pendekatan yang sistematis, sehingga pengaruh dari resiko yang timbul atas pengembangan teknologi informasi pada suatu organisasi dapat diantisipasi dan di identifikasi sebelumnya. Mendefinisikan suatu resiko dalam pengembangan teknologi informasi pada suatu organisasi terkait *** dengan Siklus Hidup Pengembangan Sistem (System Development Life Cycle [SDLC]), dimana fase-fase penerapan SDLC dalam pengembangan teknologi informasi di spesifikasikan *** analisa resiko.

2. POLA PENDEKATAN

System Development Life Cycle [SDLC] adalah suatu tahapan proses perancangan suatu sistem yang dimulai dari tahap investigasi; pembangunan; implementasi; operasi/perawatan; serta tahap penyelesaian [4]. Dari dasar tersebut di atas, strategi

penerapan manajemen resiko perlu mempertimbangkan dampak yang mungkin timbul dengan tingkat probabilitas yang berbeda untuk setiap komponen pengembangan sistem informasi.

Pola pendekatan manajemen resiko juga perlu mempertimbangkan faktor-faktor pada System Development Life Cycle (SDLC) yang terintegrasi, yaitu Mengindentifikasikan faktor-faktor resiko yang timbul dan diuraikan disetiap tahap perancangan sistem, yang tersusun sebagai berikut :

Tahap 1. Investigasi

Tahap ini suatu sistem didefinisikan, menyangkut ruang lingkup pengembangan yang akan dibuat, yang semua perencanaan atas pengembangan sistem di dokumentasikan terlebih dahulu. Dukungan yang dibutuhkan dari manajemen resiko pada tahap ini adalah faktor resiko yang mungkin terjadi dari suatu sistem informasi di identifikasikan, termasuk di dalamnya masalah serta konsep pengoperasian keamanan sistem yang semuanya bersifat strategis.

Tahap 2. Pengembangan

Tahap ini merupakan tahap dimana suatu sistem informasi dirancang, pembelian komponen pendukung sistem di laksanakan, aplikasi di susun dalam program tertentu, atau masa dimana konstruksi atas sistem di laksanakan. Pada proses ini, faktor resiko diidentifikasikan selama tahap ini dilalui, dapat berupa analisa atas keamanan sistem sampai dengan kemungkinan yang timbul selama masa konstruksi sistem di laksanakan.

Tahap 3. Implementasi

Tahap ini kebutuhan atas keamanan sistem dikonfigurasikan, aplikasi sistem di uji coba sampai pada verifikasi atas suatu sistem informasi di lakukan. Pada tahap ini faktor resiko di rancang guna mendukung proses pelaksanaan atas implementasi sistem informasi sehingga kebutuhan riil di lapangan serta pengoperasian yang benar dapat dilaksanakan.

Tahap 4. Pengoperasian dan Perawatan

Tahap ini merupakan tahap dimana sistem informasi telah berjalan sebagaimana mestinya, akan tetapi secara secara berkala sistem membutuhkan modifikasi, penambahan peralatan baik perangkat keras maupun perangkat lunak pendukung, perubahan tenaga pendukung operasi, perbaikan kebijakan maupun prosedur dari suatu organisasi. Pada tahap ini manajemen resiko lebih menitik beratkan pada kontrol berkala dari semua faktor yang menentukan berjalannya sistem, seperti perangkat keras, perangkat lunak, analisa sumber daya manusia, analisa basis data, maupun analisa atas jaringan sistem informasi yang ada.

Tahap 5. Penyelesaian/penyebaran

Tahap ini merupakan tahap dimana system informasi yang telah digunakan perlu di lakukan investasi baru karena unjuk kerja atas sistem tersebut telah berkurang, sehingga proses pemusnahan data, penggantian perangkat keras dan perangkat lunak, ataupun berhentinya kegiatan atau kepindahan organisasi ke tempat yang baru. Manajemen resiko yang perlu di perhatikan dalam tahap ini adalah memastikan proses pemusnahan atas komponen-komponen system informasi dapat berjalan dengan baik, terkelola dari segi keamanan.

Setelah pola pendekatan manajemen resiko di definisikan dalam masing-masing tahap SDLC, maka tahap selanjutnya adalah menilai manajemen resiko dalam metodologi tertentu. Upaya memberikan penilaian atas dampak resiko dalam pengembangan sistem informasi, perlu dilakukan karena dapat memberikan gambaran atas besar atau kecilnya dampak ancaman yang mungkin timbul selama proses pengembangan sistem.

3. METODOLOGI PENILAIAN RESIKO

Untuk menentukan kemungkinan resiko yang timbul selama proses pengembangan sistem informasi berlangsung, maka organisasi yang bermaksud mengembangkan sistem informasi perlu menganalisa beberapa kemungkinan yang timbul dari pengembangan sistem informasi tersebut. Adapun metodologi penilaian resiko pengembangan sistem informasi dapat diuraikan dalam 9 langkah[4], yang tersusun sebagai berikut :

a. Menentukan karakteristik dari suatu sistem

b. Mengidentifikasikan ancaman-ancaman

c. Mengidentifikasikan kelemahan sistem

d. Menganalisa pengawasan

e. Menentukan beberapa kemungkinan pemecahan masalah

f. Menganalisa pengaruh resiko terhadap pengembangan sistem

g. Menentukan resiko

h. Merekomendasikan cara-cara pengendalian resiko

i. Mendokumentasikan hasil keputusan

Sumber *Posted on 7 November 2015 by wahyutias88*

Mitigasi Risiko Mitigasi risiko adalah salah satu respon pada manajemen risko untuk menangani risiko dengan cara mengambil tindakan dini untuk mengurangi probabilitas dan atau dampak risiko. Mitigasi risiko hanya menyentuh permukaan risiko. Hal tersebut merupakan perluasan kesadaran risiko. Orang-

orang yang memulai rencana mitigasi tahu lebih banyak tentang risiko daripada mereka yang berhenti di analisis (Pandian, 2006). Mitigasi risiko termasuk dalam proses manajemen risiko.

Manajemen risiko adalah bagian utama dari setiap manajemen strategis organisasi. Manajemen risiko adalah proses dimana organisasi mengatasi risiko yang melekat pada kegiatan mereka dengan tujuan mencapai keuntungan yang berkelanjutan dalam setiap aktivitas dan seluruh kegiatan mereka (The Institute of Risk Management, 2002). Manajemen risiko memiliki 4 tahapan yang disebut IAMT, yaitu identifikasi (identification), analisis(analysis), mitigasi (mitigation), dan pelacakan (tracking). Siklus IAMT menunjukkan bahwa manajemen risiko adalah proses yang berkesinambungan dan tak berujung (Pandian, 2006). Manajemen risiko pada perusahaan IT biasanya terintegrasi pada Software Development Life Cycle (SDLC) (Unuakhalu, Sigdel, & Garikapati, 2014). Metode Agile adalah jenis pengembangan sistem jangka pendek yang fleksibel. Sehingga metode ini dapat beradaptasi terhadap perubahan requirement yang cepat. Beberapa metode yang termasuk Agile, antara lain Extreme Progrmming (XP), Scrum, Kanban, Feature Driven Development (FDD), Dynamic System Development Methodology (DSDM) dan Lean Software Development.

SDLC Software Development Life Cycle (SDLC) adalah proses yang menggambarkan metode dan strategi untuk mengembangkan desain dan memelihara proyek perangkat 13 lunak untuk memastikan bahwa semua tujuan, sasaran, fungsional, dan kebutuhan pengguna terpenuhi (Arora & Arora, 2016). SDLC memiliki tahap-tahap analisis kebutuhan, desain, pengkodean (Coding), uji coba dan pengelolaan (Kumar, Zadgaonkar, & Shukla, 2013). Beberapa contoh model SDLC, antara lain Waterfall, VModel, Agile, dll

Contoh studi kasus:

http://ariesulistio.blogspot.com/2008/06/manajemen-risiko-dalam-pengamanan.html

MANAJEMEN RISIKO DALAM PENGAMANAN SISTEM INFORMASI PERBANKAN

MANAJEMEN RISIKO DALAM

PENGAMANAN SISTEM INFORMASI PERBANKAN

Tanggal 19 Mei 2003, Bank Indonesia mengeluarkan peraturan nomor 5/8/PBI/2003 tentang “Penerapan Manajemen Risiko Bagi Bank Umum”. Tujuan dikeluarkannya peraturan ini adalah agar Bank umum di Indonesia menerapkan prinsip-prinsip manajemen risiko yang sejalan dengan rekomendasi yang dikeluarkan oleh Bank for International Settlement (BIS) yang dikenal dengan Basel II.

Dalam Basel II, perhitungan kecukupan modal tidak hanya didasari pada risiko kredit tetapi ditambah dengan perhitungan risiko lainnya, yaitu risiko pasar dan risiko operasional.

Dalam penjelasan peraturan Bank Indonesia, risiko operasional adalah risiko yang antara lain disebabkan oleh adanya ketidakcukupan dan atau tidak berfungsinya proses internal, kesalahan manusia, kegagalan sistem, atau adanya problem eksternal yang mempengaruhi operasional Bank. Sehingga jelas risiko yang disebabkan oleh kegagalan pengamanan sistem informasi termasuk dalam risiko operasional.

Pegamanan sistem informasi disini harus diartikan secara luas tidak hanya menyangkut misalnya pengamanan terhadap akses oleh orang yang tidak berwewenang. Menurut *Control Objective for Information and reated Technology* (COBIT) 4.1 kriteria pengamanan sistem informasi diantaranya adalah untuk memastikan ketersediaan (*availability*), keaslian (*integrity*), kerahasiaan (*confidentiality*), kompli dengan peraturan (*complieance*) dan *reliability* dari sistem informasi dalam menunjang kegiatan perusahaan.

Hal itu memastikan pengembangan sistem sudah sesuai dengan kebutuhan bisnis perbankan dengan melakukan terlebih dahulu studi kelayakan, pengawasan terhadap proses pemilihan sistem, dan pengujian sistem termasuk dalam obyektif pengamanan sistem informasi, yang merupakan tahapan-tahapan dalam SDLC (*System Development Life Cycle*).

Penerapan peraturan BI merupakan tantangan tersendiri bagi Bank Umum di Indonesia terutama dalam kaitannya dengan manajemen pengamanan sistem informasi. Pertama karena, belum banyak Bank yang melakukan analisis risiko dalam pengadaan kontrol sistem keamanan informasi. Kedua, belum banyak Manajemen Senior yang terlibat dalam tugas pengamanan sistem informasi. Ketiga, belum siapnya sistem pengawasan intern (*internal audit*) dalam melakukan pengawasan terhadap teknologi informasi secara umum maupun kontrol sistem pengamanan secara khusus.

Analisis Risiko

Analisis risiko seharusnya merupakan tahapan awal dalam manajemen pengamanan sistem informasi. Beberapa alasan yang digunakan Bank untuk tidak melakukan analisis risiko adalah kesulitan dalam mengkuantifikasikan risiko sistem informasi, selain juga memerlukan waktu yang cukup lama untuk melakukannya. Mengkuantifikasikan risiko operasional memang bukan hal yang mudah, tapi ini seharusnya bukan menjadi alasan untuk tidak melakukan analisis risiko.

Bank dapat menggunakan analisis kualitatif sebagai alternatifnya. Untuk tahap awal, adalah mengidentifikasikan aset, potensi kerentanannya (*vulnerabilities*), potensi ancamannya (*threats*), melakukan pengukuran risiko secara kualititatif (misalnya tinggi, sedang, rendah), baru dilanjutkan dengan menentukan kontrol.

Dengan demikian pemilihan dan pengadaan kontrol pengamanan mempunyai landasan yang kuat, misalnya apakah berdasarkan tingkat risikonya atau apakah kontrol tersebut dapat mengurangi beberapa risiko secara sekaligus.

Peran Manajemen Senior

Karena Manajemen Senior, termasuk di dalamnya jajaran Direksi, mempunyai peran dan tanggung jawab untuk mengembangkan strategi perusahaan. Mereka juga diharapkan secara eksplisit, dan terdokumentasi memberikan arahan, kebijakaan, serta menentukan akuntabilitas dalam penanganan risiko yang ditimbulkan oleh sistem informasi. Selain itu, dalam melakukan review dan memberi persetujuan terhadap kontrol pengamanan yang penting. Karena pentingnya peran Manajemen Senior, termasuk dalam salah satu pilar prinsip-prinsip manajemen risiko untuk Electronic Banking.

Sistem Pengawasan Intern

Adanya peraturan BI nomor 5/8/PBI/2003, akan menjadi pekerjaan yang cukup besar bagi Grup Audit Intern dalam menyiapkan sumber daya yang mengerti sistem informasi dan teknologi, sekaligus menguasai metodologi dan teknik-teknik audit. Menyiapkan staf internal auditor yang ada, yang sebelumnya tidak memiliki pengetahuan dan pengalaman tentang sistem informasi dan teknologi, untuk dapat melakukan audit sistem informasi akan memerlukan waktu yang cukup lama. Sedangkan merekrut staf baru yang memiliki pengetahuan sistem informasi dan audit sekaligus, juga bukan hal yang mudah.

Alternatif yang dilakukan adalah dengan menarik staf dari Grup Teknologi Informasi untuk dijadikan staf audit. Diperlukan masa transisi untuk *transfer knowledge*, sekaligus mempelajari metodologi dan teknik-teknik audit. Selanjutnya tim audit yang menangani sistem informasi dibentuk dengan melibatkan staf yang ditarik dari Grup Teknologi Informasi tadi ditambah staf Audit Intern yang ada.