ANALISIS RISIKO OPERASIONAL DALAM PEMILIHAN PERANGKAT KERAS (*HARDWARE*) DAN PERANGKAT LUNAK (*SOFTWARE*) PADA INDUSTRI PERBANKAN
(Studi Kasus: Bank X)

Ricky
Program Pascasarjana Magister Manajemen Universitas Katolik Parahyangan
ricky_mikael@yahoo.com

Abstract: *According to Undang-Undang RI Number 10 Year 1998, Bank is an entity that collects funds from the public in the form of deposits, then channeled to the community in the form of credit or other forms in order to improve the standard of living of the people. Therefore, banks need to make some effort to improve productivity, efficiency, and effectiveness in serving customers. One effort that can be done is by making almost all of the banking operations computerized. Since the computerized operations in the bank is growing, then development and software engineering analysis is needed. The development of software engineering is based on forward chaining expert system analysis against the selection of hardware and software to meet the banking needs. However, the development of software engineering is not only based on the user needs (user requirement) and tools that are used in these process analysis methods, but also the analysis of risk management in each process. By applying risk management in software engineering development process, it can assist the management in decision making (decision support system) for mitigating and minimizing the unexpected risks. The documentation of risk is not only done in the beginning of development, but also when the development is on going, so that it can make the development of software engineering becomes easier.*
Keywords: *Bank, Risk Management, Hardware, Software, Decision Support System (DSS), and Risk Documentation*

Abstrak: *Menurut Undang-Undang RI Nomor 10 Tahun 1998, Bank adalah suatu entitas yang mengumpulkan dana dari masyarakat dalam bentuk simpanan, kemudian disalurkan kepada masyarakat dalam bentuk kredit atau bentuk lainnya dalam rangka meningkatkan taraf hidup masyarakat. Oleh karena itu, bank harus membuat beberapa upaya untuk meningkatkan produktivitas, efisiensi, dan efektivitas dalam melayani pelanggan. Salah satu upaya yang dapat dilakukan adalah dengan membuat hampir semua operasi perbankan terkomputerisasi. Sejak operasi komputerisasi di bank tumbuh, maka pengembangan dan rekayasa perangkat lunak analisis dibutuhkan. Pengembangan rekayasa perangkat lunak berdasarkan analisis ahli forward chaining sistem terhadap pemilihan perangkat keras dan perangkat lunak untuk memenuhi kebutuhan perbankan. Namun, pengembangan rekayasa perangkat lunak tidak hanya didasarkan pada kebutuhan pengguna (user requirement) dan alat-alat yang digunakan dalam metode analisis ini, tetapi juga analisis manajemen risiko dalam setiap proses. Dengan menerapkan manajemen risiko dalam proses pengembangan rekayasa perangkat lunak, dapat membantu manajemen dalam pengambilan keputusan (decision support system) untuk mengurangi dan meminimalkan risiko tak terduga. Dokumentasi risiko tidak hanya dilakukan pada awal pembangunan, tetapi juga saat pembangunan berjalan, sehingga dapat membuat pengembangan rekayasa perangkat lunak menjadi lebih mudah*
Kata Kunci*: Bank, Manajemen Risiko, Perangkat Leras, Perangkat Lunak, Decision Support System (DSS), dan Dokumentasi Risiko*

1. Pendahuluan

Bank adalah badan usaha yang kekayaan terutamanya berbentuk aset keuangan (*financial assets*), yang motif utamanya adalah sosial dan *profit* (Hasibuan, Melayu SP. 2005). Bank senantiasa meningkatkan produktivitas, efisiensi, dan efektifitas pelayanannya terhadap nasabah. Salah satu upaya bank adalah dengan membuat hampir seluruh kegiatan operasional utamanya secara terkomputerisasi yang meliputi kegiatan administrasi, akuntansi, transaksi keuangan, ATM, manajemen pemasaran, dan bidang-bidang lainnya yang dapat mendukung kegiatan perbankan. Bank X merupakan sebuah lembaga keuangan

yang telah lama bergerak di bidang perbankan Indonesia. Salah satu departemen di dalam Bank X yang dapat mendukung kegiatan operasional perbankan berjalan secara terkomputerisasi adalah Departemen *Information Technology* (IT). Departemen IT membangun sistem informasi yang disesuaikan dengan kebutuhan penggunanya. Sistem Informasi (SI) adalah suatu aliran di dalam suatu organisasi yang mempertemukan kebutuhan pengolahan transaksi, mendukung operasi, bersifat manajerial, dan kegiatan strategi dari suatu organisasi dan menyediakan pihak luar tertentu dengan laporan-laporan yang diperlukan (Ladjamudin, bin Al-Bahra. 2005). Dengan berkembangnya kegiatan operasional bank menjadi terkomputerisasi tersebut, maka bank perlu melakukan analisis terhadap pembangunan dan pengembangan rekayasa perangkat lunak (*software engineering*).

Dalam pembangunan rekayasa perangkat lunak maupun pengembangannya, dapat dilakukan berdasarkan analisis pemilihan perangkat keras (*hardware*) dan perangkat lunak (*software*) yang sesuai dengan kebutuhan Bank X. Namun Bank X juga perlu memperhatikan kesesuaian antara kapasitas bank (secara material dan non-material) dengan fasilitas yang dibutuhkan, sehingga investasi yang dilakukan dapat menjadi efektif dan memberikan nilai tambah untuk perbankan tersebut dalam jangka panjang. Hal-hal tersebut menjadi penting, karena dalam pembangunan rekayasa perangkat lunak tersebut menyangkut data keuangan dari masyarakat luas, sehingga jika terjadi kesalahan dalam pemilihan dan penerapan perangkat keras (*hardware*) maupun perangkat lunak (*software*) dapat berakibat fatal bagi bank tersebut.

Namun pembangunan rekayasa perangkat lunak tidak hanya didasarkan pada metode analisis terhadap kebutuhan pengguna (*user requirement*) dan alat bantu (*tools*) yang digunakan dalam proses-proses tersebut, namun diperlukan juga analisis manajemen risiko dalam setiap prosesnya. Manajemen risiko memiliki tujuan, yaitu untuk mengenali risiko dalam sebuah proyek dan mengembangkan strategi untuk mengurangi atau bahkan menghindarinya, dilain sisi juga perlu mencari cara untuk memaksimalkan peluang yang ada (Wideman, Max.R. 1992). Proses pengdokumentasian risiko (*risk documentation*) tidak hanya dilakukan pada awal pembangunan proyek rekayasa perangkat lunak, namun perlu dilakukan selama proses pembangunan rekayasa perangkat lunak tersebut, sehingga akan lebih memudahkan dalam pengembangan rekayasa perangkat lunak di masa yang akan datang.

Adapun pendokumentasian risiko-risiko perlu dilakukan dalam proses pembangunan rekayasa perangkat lunak (*software engineering*) pada Bank X, karena selama ini pengembang perangkat lunak (*software engineer*) merasa mengetahui seluruh hal yang terkait dengan pengembangan rekayasa perangkat lunak dan mengetahui seluruh keperluan dan keinginan *user*. Namun para pengembang perangkat lunak tersebut tidak pernah menyadari adanya hal yang tidak terduga sebelumnya yang dapat terjadi, seperti *turn over* karyawan divisi IT, perubahan mendadak *user requirement*, *budget overages*, adanya *error* dan *bug*, dan masalah teknis lainnya sehingga proyek menjadi terhambat. Dengan adanya dokumentasi terhadap risiko-risiko tersebut, maka dapat membantu manajemen Bank X dalam pengambilan keputusan (*decision support system*) untuk langkah mitigasi dan meminimalisir terjadinya risiko lainnya yang tidak terduga sebelumnya. Oleh sebab itu, perlu dilakukan analisis secara mendalam untuk dapat memperkirakan segala kemungkinan yang dapat terjadi dalam proses pembangunan rekayasa perangkat lunak (*software engineering*) pada Bank X.

2. Dasar Teori

Bank

Menurut Undang-Undang RI Nomor 10 Tahun 1998, bank adalah badan usaha yang menghimpun dana dari masyarakat dalam bentuk simpanan dan menyalurkannya kepada masyarakat dalam bentuk kredit dan atau bentuk-bentuk lainnya dalam rangka meningkatkan taraf hidup rakyat banyak. Secara spesifik bank dapat berfungsi sebagai berikut:

1.) *Agent of trust*

Dasar utama kegiatan perbankan adalah kepercayaan (*trust*), baik dalam hal menghimpun dana maupun penyaluran dana (Budisantoso, T dan Sigit. 2006).

2.) *Agent of development*

Kegiatan perekonomian masyarakat di sektor moneter dan sektor riil tidak dapat dipisahkan, karena kedua sektor tersebut saling mempengaruhi dan saling berinteraksi. Sektor riil tidak dapat berkinerja dengan baik apabila sektor moneter tidak bekerja dengan

baik. Kegiatan bank berupa penghimpunan dan penyaluran dana sangat diperlukan bagi lancarnya kegiatan perekonomian di sektor riil. Kegiatan bank tersebut memungkinkan masyarakat melakukan kegiatan investasi, kegiatan distribusi, serta kegiatan konsumsi barang dan jasa, mengingat bahwa kegiatan investasi-disribusi-konsumsi tidak dapat terlepas dari adanya penggunaan uang (Budisantoso, T dan Sigit. 2006).

3.) Agent of service

Bank disamping melakukan kegiatan penghimpunan dan penyaluran dana, bank juga memberikan penawaran jasa perbankan yang lain kepada masyarakat, yaitu berupa jasa pengiriman uang, penitipan barang berharga, pemberian jaminan bank, dan penyelesaian tagihan (Budisantoso, T dan Sigit 2006).

Risiko *(Risk)*

Risiko merupakan variasi dalam hal-hal yang mungkin terjadi secara alami didalam suatu situasi (Fisk, E.R. 1997). Risiko adalah ancaman terhadap kehidupan, properti atau keuntungan finansial akibat bahaya yang terjadi (Duffield, C & Trigunarsyah, B. 1999). Risiko-risiko yang timbul dalam suatu tujuan, perlu dilakukan skala prioritas terhadap risiko-risiko yang akan memberikan pengaruh terhadap pencapaian suatu tujuan. Risiko-risiko tersebut adalah (Wideman, Max.R. 1992):

1.) External, (tidak dapat diprediksi, tidak dapat dikontrol):
   a. Perubahan peraturan perundang-undangan,
   b. Bencana alam (badai, banjir, gempa bumi),
   c. Akibat kejadian pengrusakan dan sabotase,
   d. Pengaruh lingkungan dan sosial,

2.) External, (dapat diprediksi, tetapi tidak dapat dikontrol):
   a. Risiko pasar,
   b. Risiko operasional,
   c. Pengaruh lingkungan,
   d. Pengaruh sosial,
   e. Perubahan mata uang (*kurs*),
   f. Inflasi,
   g. Pajak

3.) Internal, (non-teknik, tetapi umumnya dapat dikontrol):
   a. Manajemen,
   b. Jadwal yang terlambat,
   c. Pertambahan biaya,
   d. Cash flow,
   e. Potensi kehilangan atas manfaat dan keuntungan

4.) Teknik (dapat dikontrol):
   a. Perubahan teknologi,
   b. Risiko-risiko spesifikasi atas teknologi,
   c. Desain

5.) Hukum, timbulnya kesulitan akibat dari :
   a. Lisensi,
   b. Hak paten,
   c. Gugatan dari luar,
   d. Gugatan dari dalam,
   e. Hal-hal tak terduga.

Manajemen Risiko (*Risk Management*)

Manajemen risiko (*risk management*) merupakan pendekatan yang dilakukan terhadap risiko dengan cara memahami, mengidentifikasi dan mengevaluasi risiko. Kemudian mempertimbangkan apa yang akan dilakukan terhadap dampak yang ditimbulkan dan kemungkinan pengalihan risiko kepada pihak lain atau mengurangi risiko yang terjadi. Manajemen risiko (*risk management*) adalah semua rangkaian kegiatan yang berhubungan dengan risiko yaitu perencanaan (*planning*), penilaian (*assessment*), penanganan (*handling*) dan pemantauan (*monitoring*) risiko (Kerzner, H. 2001).

Tujuan dari manajemen risiko adalah untuk mengenali risiko dalam sebuah proyek dan mengembangkan strategi untuk mengurangi atau bahkan menghindarinya, dilain sisi juga perlu mencari cara untuk memaksimalkan peluang yang ada (Wideman, Max.R. 1992). Manajemen krisis merupakan reaksi terhadap risiko yang sebelumnya tidak teridentifikasi yang muncul menjadi bahaya yang jelas saat ini. Pendekatan yang lebih proaktif adalah dengan mengidentifikasi risiko proyek dan merencanakan bagaimana merespon risiko tersebut ketika muncul. Sangatlah baik untuk mengambil aksi konkret untuk mencegah risiko yang teridentifikasi yang dapat menyebabkan permasalahan dan bukan hanya sekedar

memperbaiki produk dari kesalahan. Manajemen risiko merupakan aplikasi alat bantu dan prosedur yang tepat untuk mengatasi risiko dengan batasan yang dapat diterima. Manajemen risiko (*risk management)* terdiri atas aktivitas-aktivitas berikut ini (Boehm, Barry W. 1989):

1.) Risk assessment (menggambarkan risiko apa yang terjadi dan harus fokus terhadap yang mana)

Merupakan proses untuk menguji proyek dan mengidentifikasi area risiko potensial. Identifikasi risiko dapat difasilitasi dengan bantuan suatu daftar area risiko umum untuk proyek *software*, atau dengan menguji isi *database* organisasi yang berisi risiko serta strategi mitigasi yang teridentifikasi sebelumnya (baik sukses maupun tidak). Analisis risiko melibatkan pengujian bagaimana *outcome* proyek berubah dengan melakukan modifikasi variabel input risiko.

a. Membuat daftar semua potensi bahaya yang akan mempengaruhi proyek.
b. Memperkirakan probabilitas kejadian dan potensi kehilangan dari tiap *item* yang didaftar.
c. Mengurutkan *item-item* tersebut dari yang paling berbahaya sampai kurang berbahaya.

2.) Risk prioritization

Membantu proyek memfokuskan pada risiko yang paling berat dengan memperkirakan *risk exposure*. Prioritas dapat dilakukan dengan cara kuantitatif, dengan estimasi probabilitas (antara 0.1-1.0) dengan kegagalan relatif pada skala 1 sampai 10. Menggabungkan beberapa faktor ini akan menyediakan estimasi *risk exposure* bagi tiap *risk item*, yang dapat terjadi pada kisaran 0.1 sampai 10. Semakin tinggi *exposure*, semakin agresif risiko yang harus ditangani. Lebih mudah untuk mengestimasi probabilitas dan dampaknya sebagai *high, medium* atau *low*. Dengan *item* tersebut, setidaknya terdapat 1 dimensi risiko dengan *rate high* dan perlu diperhatikan terlebih dahulu.

3.) Risk avoidance

Merupakan salah satu cara untuk berhubungan dengan risiko, dimana risiko dihindari dengan tidak melaksanakan proyek tertentu dan hanya melaksanakan proyek yang pasti.

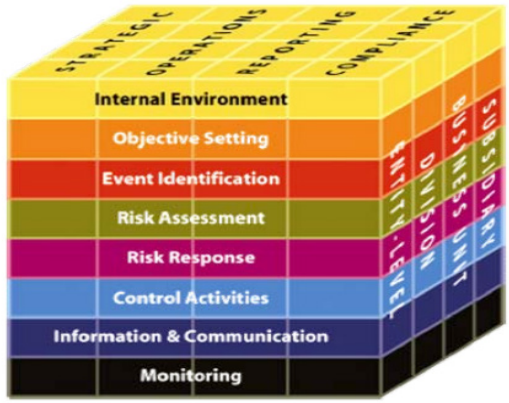
4.) Risk control (berbuat sesuatu terhadap *item-item* tersebut)

Merupakan proses mengatur risiko untuk mencapai *outcome* yang dikehendaki. Merencanakan manajemen risiko akan menghasilkan rencana untuk berhubungan dengan setiap risiko yang signifikan, termasuk mitigasi pendekatan, kepemilikan, dan *timeline*. Resolusi risiko merupakan eksekusi rencana yang berkaitan dengan tiap risiko. Dimana pada akhirnya, *risk monitoring* akan membantu melacak perkembangan pemecahan tiap risiko:

a. Menggunakan teknik dan strategi untuk mengurangi risiko tertinggi.
b. Mengimplementasikan strategi untuk menetapkan faktor risiko tertinggi.
c. Mengawasi efektivitas strategi dan *level* perubahan risiko pada proyek.

Komponen Manajemen Risiko

Manajemen risiko yang diterapkan oleh manajemen dengan kerangka kerja (*framework*)



Gambar 1 *ERM Framework*
Sumber: Boehm, Barry W. (1989)

COSO memiliki delapan komponen yang saling terkait. Komponen-komponen ini dibangun dari tata kelola perusahaan yang diintegrasikan dengan proses manajemen. Delapan komponen pada kerangka kerja (*framework*) COSO tersebut diintegrasikan dengan strategi, operasi, sistem pelaporan, dan kepatuhan serta keberadaan berbagai unit kerja yang terlibat dalam proses manajemen risiko korporat baik di kantor pusat maupun di kantor cabang. Komponen manajemen risiko tersebut adalah (Boehm, Barry W. 1989):

1.) Lingkungan internal

Lingkungan internal yang kondusif, *supportif*, dan positif akan mempengaruhi secara langsung budaya kerja perusahaan dalam melihat dan memitigasi suatu risiko, termasuk di dalamnya filosofi manajemen risiko, toleransi risiko, nilai-nilai integritas dan etika serta lingkungan kerja.

2.) Penetapan sasaran (target)

Penetapan sasaran dan target bisnis harus dilakukan dengan terlebih dahulu memperhatikan risiko-risiko potensial yang mempengaruhi secara negatif upaya-upaya pencapaian sasaran/target. Manajemen akan selalu menetapkan target bisnis dalam koridor toleransi risiko perusahaan.

3.) Identifikasi risiko

Manajemen akan mengidentifikasikan risiko-risiko internal dan eksternal yang dapat mempengaruhi usaha pencapaian sasaran. Manajemen selalu berupaya memposisikan diri pada suatu *level* sehingga dengan mudah dapat membedakan antara risiko dan peluang. Setiap peluang yang berhasil ditangkap akan dimasukkan ke dalam proses penetapan sasaran perusahaan.

4.) Penilaian risiko

Risiko-risiko dianalisis dan dipertimbangkan probabilitas terjadinya (*likelihood*) dan potensi dampak kerugiannya (*impact*) sebagai acuan mengelolanya. Risiko diukur berdasarkan pendekatan risiko inheren dan risiko residual. Risiko inheren adalah risiko yang melekat pada setiap keputusan sebelum dilakukan perlakuan risiko. Risiko residual adalah risiko yang masih ada setelah dilaksanakan perlakuan risiko.

5.) Tindak lanjut risiko

Manajemen akan menetapkan tindak lanjut dan respon yang efektif terhadap suatu risiko. Spektrum respon menghindari, menerima, mereduksi, atau mentransfer risiko. Pilihan respon akan dipengaruhi oleh toleransi dan hasrat risiko manajemen dan perusahaan.

6.) Pengendalian dan pengawasan risiko

Sejumlah kebijakan dan pedoman dibuat, ditetapkan, dan diterapkan untuk menciptakan suatu sistem pengendalian dan pengawasan yang efektif sehingga memudahkan manajemen memilih respon risiko yang efektif dan efisien.

7.) Sistem pelaporan dan *software* manajemen risiko

Berbagai informasi yang relevan diidentifikasikan, ditangkap, dan dikomunikasikan dalam bentuk yang informatif, terstruktur dengan baik dan tepat waktu agar setiap penanggung jawab organisasi dapat melaksanakan tanggung jawabnya masing-masing di dalam mencapai sasaran perusahaan. Sistem pelaporan ini didukung sistem informasi berbasis komputer dengan menggunakan fasilitas *web*. Manajemen memiliki prioritas yang tinggi untuk mengembangkan dan memiliki kegiatan yang terintegrasi, efektif, dan terhubung secara *online* ke seluruh unit kerja di kantor pusat dan kantor cabang.

8.) Pemantauan

Pemantauan adalah efektivitas yang penting sehingga dapat diketahui modifikasi dan perbaikan yang diperlukan pada sistem manajemen risiko korporat terintegrasi. Pemantauan dilaksanakan melalui aktivitas manajemen yang berkelanjutan, evaluasi khusus, atau keduanya.

Umumnya manajemen risiko (*risk management*) perbankan mengacu kepada Basel II, dimana implementasi Basel II di Indonesia dilakukan dengan cara Basel II menghitung kebutuhan modal yang sesuai dengan profil risiko bank, serta memberikan insentif bagi peningkatan kualitas dalam praktik manajemen risiko di perbankan. Menggunakan berbagai alternatif pendekatan (*approaches*) dalam mengukur risiko kredit (*credit risk*), risiko pasar (*market risk*), dan risiko operasional (*operational risk*), maka hasilnya adalah perhitungan modal bank yang lebih sensitif terhadap risiko (*risk sensitive capital allocation*) (Hasibuan, Melayu SP. 2005):

1.) Risiko kredit *(credit risk)*

Risiko kerugian yang terkait dengan

Tabel 1. Tipe-Tipe Risiko

Tipe Risiko	Keterangan
Generic Risk	Ancaman umum pada semua proyek. Sebagai contoh: perubahan *requirement*, kehilangan anggota tim, kehilangan dana
Product-Specific Risk	Risiko tingkat tinggi yang berhubungan dengan tipe produk yang dikembangkan. Sebagai contoh: ketersediaan sumber daya pengujian
Project Risk	Mempengaruhi jadwal proyek atau sumber daya
Product Risk	Mempengaruhi kualitas atau performansi perangkat lunak
Business Risk	Mempengaruhi kelangsungan hidup perangkat lunak (SDLC)

(Sumber: Kristanti, Tanti 2007)

kemungkinan kegagalan *counterparty* memenuhi kewajibannya; atau risiko bahwa debitur tidak membayar kembali utangnya.

2.) Risiko pasar *(market risk)*

Risiko kerugian baik pada posisi *on-* maupun *off-balance sheet* yang timbul dari pergerakan harga pasar. Istilah risiko pasar digunakan untuk menyebut kelompok risiko yang timbul dari perubahan tingkat suku bunga, *kurs* valuta asing, dan hal-hal lain yang nilai-nya ditentukan pasar, misal ekuitas dan komoditi.

3.) Risiko operasional *(operational risk)*

Risiko yang antara lain disebabkan adanya ketidakcukupan dan atau tidak berfungsinya proses internal, kesalahan manusia, kegagalan sistem (IT), atau adanya *problem* eksternal yang mempengaruhi operasional bank, hukum, dan regulasi.

Menurut Godfrey (1996) dalam melakukan identifikasi risiko, perlu diupayakan terlebih dahulu untuk menentukan sumber risiko itu sendiri secara komprehensif. Risiko dapat bersumber dari politik (*political*), lingkungan (*environmental*), perencanaan (*planning*), pemasaran (*market*), ekonomi (*economic*), keuangan (*financial*), proyek (*project*), teknik (*technical*), manusia (*human*), kriminal (*criminal*), dan keselamatan (*safety*). Risiko-risiko yang teridentifikasi, dibagi berdasarkan tipe-tipe dari risiko tersebut. Beberapa tipe risiko yang dapat muncul selama proyek pengembangan perangkat lunak berjalan dapat dilihat pada Tabel 1.

Risiko dalam Manajemen Proyek

Menghadapi risiko dalam *software engineering* (rekayasa perangkat lunak), harus berdasarkan pendekatan tertentu dan melibatkan penelitian dari pengalaman sejumlah manajer proyek, para penulis, dan peneliti. Terdapat 10 macam risiko terbesar dalam manajemen proyek (Boehm, Barry W. 1989), yaitu:

1.) Personnel shortfall

2.) Penjadwalan dan *budgeting* yang tidak realistik

3.) Membangun fungsi dan properti yang salah

4.) Membangun *user interface* yang salah

5.) Gold-plating

6.) Melanjutkan aliran perubahan *requirement*

7.) Shortfalls pada *furnished component* eksternal

8.) Shortfalls pada *performed task* eksternal

9.) Real-time performance shortfall

10.) Straining computer-science capabilities

Jika ada diantaranya pernah terjadi pada proyek, dibuat sebagai faktor risiko utama sehingga tidak akan kembali terjadi pada proyek di masa mendatang, berdasarkan pengalaman para *software engineer* dan praktisi manajemen, risiko dapat dikontrol (McConnell, Steve 1996). Evaluasi risiko pada suatu proyek tergantung pada probabilitas terjadinya risiko tersebut (frekuensi kejadian) dan dampak dari risiko tersebut bila terjadi (Duffield, C & Trigunarsyah, B. 1999).

Dalam membandingkan pilihan proyek dari berbagai risiko yang terkait sering digunakan "indeks risiko = frekuensi *(likelihood) x* dampak *(concequences)*" (Jones, Capers 1994).

Analisis tingkat nilai risikonya yang merupakan hasil penerimaan risiko *(risk acceptability)* tergantung perkalian antara kemungkinan (*likelihood*) dengan konsekuensi *(consequences)*. Tingkat penerimaan risiko dapat dilihat pada Tabel 2.

Tabel 2. Indikator Penerimaan Risiko

Indikator Penerimaan Risiko	Nilai Risiko
Unacceptable (tidak dapat diterima)	$X \geq 15$
Undesirable (tidak diharapkan)	$8 \leq X < 15$
Acceptable (dapat diterima)	$3 \leq X < 8$
Negligible (dapat diabaikan)	$X < 3$

Sumber: Godfrey, P.S., Sir William Halcrow and Partners Ltd., 1996

Penanganan risiko hanya dilakukan terhadap risiko dominan yakni risiko kategori *unacceptable* dan *undesirable,* dilanjutkan dengan menentukan kepemilikan tanggungjawab risiko *(ownership of risk)* (Godfrey, P.S., Sir William Halcrow and Partners Ltd. 1996). Untuk melakukan analisis risiko secara efektif, harus mempertimbangkan hal-hal berikut (Duffield, C & Trigunarsyah, B. 1999):

1.) Analisis yang dilakukan harus difokuskan pada kerugian finansial langsung daripada gangguan pelayanan atau kematian dan kerugian.
2.) Tingkat ketidakpastian dalam setiap perkiraan *output* harus dapat dinilai.
3.) Akurasi dari analisis harus sesuai dengan akurasi data dan tahapan proyek.
4.) Biaya dan usaha dalam melakukan analisis harus serendah mungkin yang dapat diserap oleh anggaran proyek.

Respon risiko adalah tindakan penanganan yang dilakukan terhadap risiko yang mungkin terjadi. Risiko-risiko penting yang sudah diketahui perlu ditindaklanjuti dengan respon yang dilakukan oleh kontraktor dalam menangani risiko tersebut. Metode yang dipakai dalam menangani risiko (Flanagan, R & Norman, G. 1993):

1.) Menahan risiko *(risk retention)*

    Merupakan bentuk penanganan risiko yang mana akan ditahan atau diambil sendiri oleh suatu pihak. Biasanya cara ini dilakukan apabila risiko yang dihadapi tidak mendatangkan kerugian yang terlalu besar atau kemungkinan terjadinya kerugian itu kecil, atau biaya yang dikeluarkan untuk menanggulangi risiko tersebut tidak terlalu besar dibandingkan dengan manfaat yang akan diperoleh.

2.) Mengurangi risiko *(risk reduction)*

    Tindakan untuk mengurangi risiko yang kemungkinan akan terjadi dengan cara:

    a. Pendidikan dan pelatihan bagi para tenaga kerja dalam menghadapi risiko
    b. Perlindungan terhadap kemungkinan kehilangan
    c. Perlindungan terhadap orang dan properti

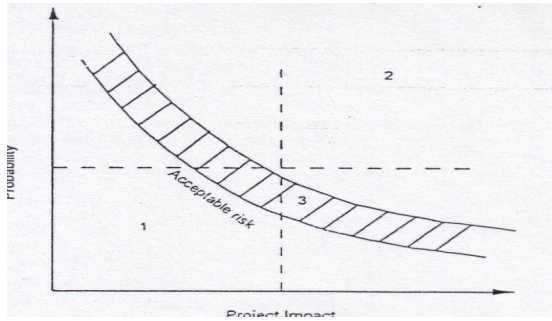
3.) Mengalihkan risiko *(risk transfer)*

    Pengalihan ini dilakukan untuk memindahkan risiko kepada pihak lain. Bentuk pengalihan risiko yang dimaksud adalah asuransi dengan membayar premi.

4.) Menghindari risiko *(risk avoidance)*

    Menghindari risiko sama dengan menolak untuk menerima risiko yang berarti menolak untuk menerima proyek tersebut (Labombang, Mastura. 2011, 39 – 46).

Terdapat dua cara para *software engineer* menangani risiko, yaitu *software engineer* yang reaktif selalu memperbaiki masalah saat masalah tersebut muncul, dan *software engineer* proaktif yang selalu memikirkan kemungkinan risiko yang



Gambar 2. Indeks Risiko
Sumber: Duffield, C & Trigunarsyah, B. (1999)

dapat terjadi pada suatu proyek sebelum risiko-risiko tersebut muncul (Kristanti, Tanti 2007).

Rekayasa Perangkat Lunak (RPL)

Rekayasa perangkat lunak (*software engineering*) merupakan pembangunan dengan menggunakan prinsip atau konsep rekayasa dengan tujuan menghasilkan perangkat lunak yang bernilai ekonomi yang dipercaya dan bekerja secara efisien menggunakan mesin (A. S. Rosa dan Shalahuddin. M. 2011).

Perangkat lunak banyak dibuat dan pada akhirnya sering tidak digunakan karena tidak memenuhi kebutuhan pelanggan atau bahkan karena masalah non-teknis seperti keengganan pemakai perangkat lunak (*user*) untuk mengubah cara kerja dari manual ke otomatis, atau ketidakmampuan *user* menggunakan komputer. Oleh karena itu, rekayasa perangkat lunak dibutuhkan agar perangkat lunak yang dibuat tidak hanya menjadi perangkat lunak yang tidak terpakai (A. S. Rosa dan Shalahuddin. M. 2011).

Rekayasa perangkat lunak lebih fokus kepada praktik pengembangan perangkat lunak dan mengirimkan perangkat lunak yang bermanfaat kepada pelanggan (*customer*). Sedangkan ilmu komputer lebih fokus pada teori dan konsep dasar perangkat komputer. Rekayasa perangkat lunak lebih fokus pada bagaimana membuat perangkat lunak yang memenuhi kriteria tertentu (A. S. Rosa dan Shalahuddin. M. 2011):

1.) Dapat terus dipelihara setelah perangkat lunak selesai dibuat seiring berkembangnya teknologi dan lingkungan (*maintainability*).
2.) Dapat diandalkan dengan proses bisnis yang dijalankan dan perubahan yang terjadi (*dependability* dan *robust*).
3.) Efisien dari segi sumber daya dan penggunaan.
4.) Kemampuan untuk dipakai sesuai dengan kebutuhan (*usability*).

Proses perangkat lunak (*software process*) adalah sekumpulan aktivitas yang memiliki tujuan untuk mengembangkan atau mengubah perangkat lunak. Secara umum proses perangkat lunak terdiri dari (A. S. Rosa dan Shalahuddin. M. 2011):

1.) Pengumpulan spesifikasi (*specification*)
Mengetahui apa saja yang harus dapat dikerjakan sistem perangkat lunak dan batasan pengembangan perangkat lunak.
2.) Pengembangan (*development*)
Pengembangan perangka lunak untuk menghasilkan sistem.
3.) Validasi (*validation*)
Memeriksa apakah perangkat lunak sudah memenuhi keinginan pelanggan (*customer*).
4.) Evolusi (*evolution*)
Mengubah perangkat lunak untuk memenuhi perubahan kebutuhan pelanggan (*customer*).

Sistem Pakar (*Expert System*)

Sistem pakar adalah program komputer yang merepresentasikan dan melakukan penalaran dengan pengetahuan dari seorang pakar dalam bidang tertentu dengan pandangan untuk memecahkan masalah atau memberikan nasihat. Pakar manusia (*human expert*) adalah seseorang yang mempunyai penguasaan terhadap suatu masalah. Berdasarkan pengalamannya, pakar manusia mengembangkan kemampuannya dalam memecahkan masalah secara lebih efisien dan efektif. Sistem pakar juga harus dapat menjelaskan alasan dari setiap langkah dalam mencapai suatu tujuan (*goal*) dan menjawab pertanyaan tentang solusi yang dicapainya, seperti halnya seorang pakar manusia.

Pohon keputusan (*decision tree*) merupakan salah satu alat representasi pengetahuan yang sering digunakan pada sistem pakar. Solusi pada pohon keputusan dihasilkan dari serangkaian solusi yang mungkin melalui serangkaian keputusan atau pertanyaan yang akan mengurangi area pencarian solusi. Masalah yang sesuai menggunakan pohon keputusan adalah masalah yang telah menyediakan jawaban untuk masalah tersebut dari satu set atau beberapa alternatif jawaban yang mungkin.

Pohon keputusan terdiri dari sejumlah simpul (*nodes*) dan cabang (*branch*) yang menghubungkan simpul orang tua (*parents*) ke simpul anak (*child*) dari bagian atas sampai bagian bawah dari pohon keputusan. Simpul paling atas disebut juga sebagai akar (*roots*). Akar tidak memiliki *parents*, sedangkan setiap simpul yang di bawahnya hanya memiliki satu *parent*. Simpul paling bawah yang tidak memiliki anak disebut sebagai simpul daun (*leaf*). Simpul daun merepresentasikan semua solusi (keputusan) yang diturunkan melalui pohon keputusan. Secara umum pohon keputusan menggunakan beberapa kriteria untuk memilih cabang mana yang akan dilalui sehingga nantinya hanya terpilih satu cabang yang menghasilkan keputusan.

Teknik penalaran untuk membangun suatu pohon keputusan dapat diklasifikasikan menjadi dua, yaitu:

1.) Penalaran maju atau runut maju (*forward chaining*)
2.) Penalaran mundur atau runut mundur (*backward chaining*)

Pohon keputusan kemudian dapat dikonversi menjadi serangkaian aturan yang direpresentasikan oleh jalur yang berbeda-beda pada pohon keputusan. Aturan yang dihasilkan dari pohon keputusan selanjutnya digunakan sebagai basis pengetahuan yang diperlukan pada sistem pakar (Tim Penerbit Andi 2003).

Runut Maju *(Forward Chaining)*

Runut maju (*forward chaining*) berarti menggunakan himpunan aturan kondisi-aksi. Dalam metode ini, data digunakan untuk menentukan aturan mana yang akan dijalankan, kemudian aturan tersebut dijalankan. Mungkin proses menambahkan data ke memori kerja. Proses diulang sampai ditemukan suatu hasil (Kuswandi, dan Mutiara, Ema 2004, 79-82).

Operasi dari sistem *forward chaining* dimulai dengan memasukkan sekumpulan fakta yang diketahui ke dalam memori kerja (*working memory*), kemudian menurunkan fakta baru berdasarkan aturan yang premisnya cocok dengan fakta yang diketahui. Proses ini dilanjutkan sampai dengan mencapai *goal* atau tidak ada lagi aturan yang premisnya cocok dengan fakta yang diketahui. Pendefinisian struktur pengendalian data aturan ditulis dalam struktur *If – Then* dan diberi nomor aturan untuk membedakan aturan yang satu dengan yang lain. Aturan akan dituliskan pada *file* teks dengan menggunakan *sintaks prolog*.

Sintaks *rule* yang digunakan adalah:

rule <*rule* id>
if [<N>:<kondisi>,…]
then [<aksi>,…]

Keterangan:

a. *rule id*: nomor identifikasi dari aturan tersebut.
b. N: nomor identifikasi untuk kondisi
c. kondisi: premis atau pola untuk dicocokkan dengan memori kerja
d. aksi: konklusi atau aksi yang akan dilakukan.

Decision Support System (DSS)

Decision support system (DSS) merupakan sebuah sistem untuk membantu pengambilan keputusan di dalam sebuah perusahaan atau organisasi. DSS membantu pengambilan keputusan manajemen dengan menggabungkan data, model-model, dan alat-alat analisis yang kompleks, serta perangkat lunak yang akrab dengan tampilan pengguna ke dalam satu sistem yang memiliki kekuatan besar (*powerful*) yang dapat mendukung pengambilan keputusan yang *semi* terstruktur atau tidak terstruktur. DSS menyajikan ke pengguna satu perangkat alat yang fleksibel dan memiliki kemampuan untuk analisis data penting (Turban, Efraim, Mclean Ephraim dan Wetherbe James 2002, 423-478).

Decision support system (DSS) atau sistem pendukung keputusan (SPK) adalah bagian dari sistem informasi berbasis komputer, termasuk sistem berbasis pengetahuan (manajemen pengetahuan) yang dipakai untuk mendukung pengambilan keputusan dalam suatu organisasi atau sebuah perusahaan (Rosadi, Anisa Ayu. 2009).

Tujuan dari *decision support system* (DSS) atau sistem pendukung keputusan (SPK) adalah:

1.) Membantu menyelesaikan masalah semi-terstruktur
2.) Mendukung manajer dalam mengambil keputusan
3.) Meningkatkan efektifitas bukan efisiensi pengambilan keputusan

Gaya pembuatan keputusan menggunakan parameter gaya pembuatan keputusan yang didasarkan pada cara dimana informasi dikumpul-

Tabel 3. Perbedaan antara Keputusan Analitis dan Heuristic

Pembuat Keputusan Analitis	Pembuat Keputusan *Heuristic*
Belajar dengan menganalisis.	Belajar dengan bertindak
Menggunakan prosedur langkah demi langkah.	Menggunakan *trial* and *error*
Menilai informasi dan model secara kuantitatif.	Menilai pengalaman
Membangun algoritma dan model matematis.	Mengandalkan penginderaan
Mengupayakan solusi optimal	Mengupayakan solusi yang memuaskan

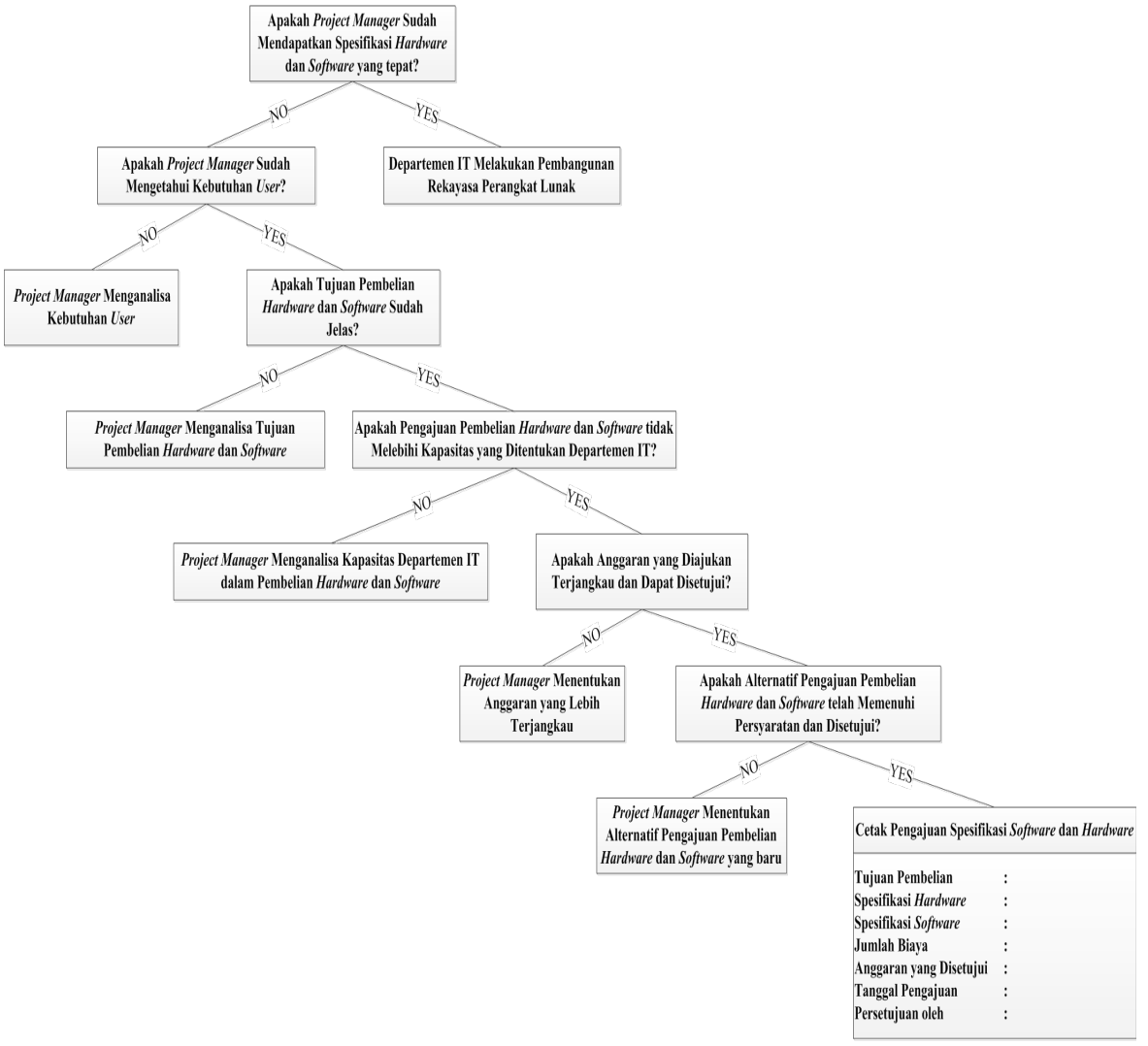
(Sumber: Rosadi, Anisa Ayu., 2009)

kan, diproses, dan digunakan, serta bagaimana informasi dikomunikasikan dan diterapkan. Dalam pembuatan gaya pembuatan keputusan terdapat penggolongan keputusan yaitu keputusan analitis dan *heuristic* seperti yang terlihat pada Tabel 3.

3. Hasil dan Pembahasan

Dalam pembangunan rekayasa perangkat lunak maupun pengembangannya, perlu dilakukan sistem pengambilan keputusan dari proses penentuan alternatif perangkat keras (*hardware*) dan perangkat lunak (*software*). Proses penentuan alternatif tersebut menggunakan metode *expert system*/sistem pakar, yaitu dengan cara membuat suatu pohon keputusan (*decision tree*) yang memetakan cara berpikir para pakar (Kuswandi, dan Mutiara, Ema 2004 79-82). Dalam kasus penentuan alternatif perangkat keras (*hardware*) dan perangkat lunak (*software*) pada Bank X ini, dapat menggunakan metode penyusunan pohon keputusan (*decision tree*) sistem pakar *forward chaining*.

Penyusunan pohon keputusan (*decision tree*) sistem pakar *forward chaining* ini, disesuaikan dengan prioritas dan aturan pembangunan rekayasa perangkat lunak (*software engineer*) yang terdapat di Departemen *Information Technology* Bank X. Pohon keputusan (*decision tree*) merupakan susunan yang berupa pertanyaan-pertanyaan yang diperoleh berdasarkan proses pengumpulan data yang telah dilakukan melalui metode wawancara pada 2 orang staf ahli (*expert*) bagian IT dan manajemen risiko IT pada Bank X. Pertanyaan yang diajukan disesuaikan dengan



Gambar 3. *Decision Tree* Sistem Pakar *Forward Chaining*

prioritas dan aturan dalam setiap prosesnya. Aturan dari pembuatan *IF… THEN…* dari kasus penentuan alternatif perangkat keras (*hardware*) dan perangkat lunak (*software*) ini adalah seperti pada Gambar 3.

Berdasarkan Gambar 3, dapat jelaskan bahwa alternatif pemilihan perangkat keras (*hardware*) dan perangkat lunak (*software*) dapat disusun berdasarkan algoritma berikut:

Tahap 1

IF *Project manager* sudah mengetahui kebutuhan *user*

THEN Analisis tujuan pembelian perangkat keras (*hardware*) dan perangkat lunak (*software*) dilakukan

Tahap 2

IF *Project manager* sudah mengetahui kebutuhan *user AND* tujuan pembelian perangkat keras (*hardware*) dan perangkat lunak (*software*) sudah jelas

THEN Pengecekan kapasitas yang ditentukan departemen IT

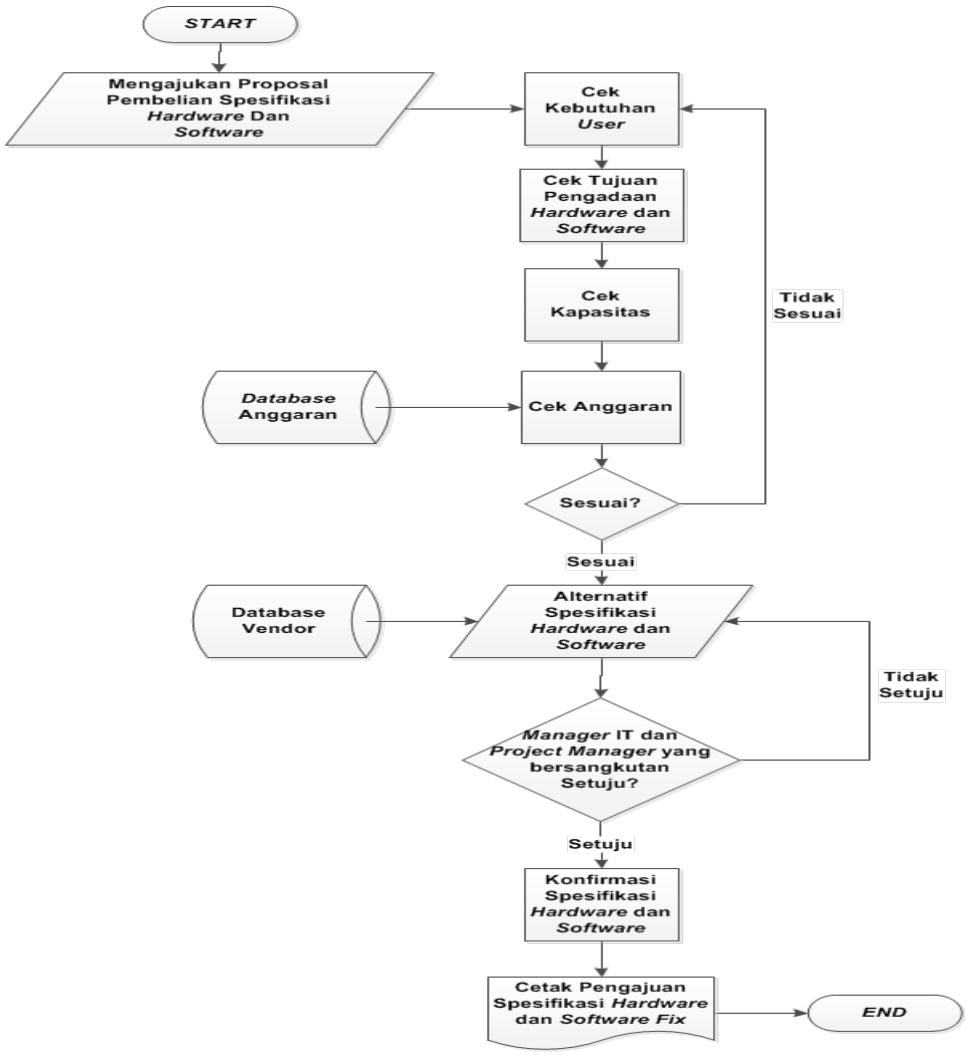
Tahap 3

IF *Project manager* sudah mengetahui kebutuhan *user AND* tujuan pembelian perangkat keras (*hardware*) dan perangkat lunak (*software*) sudah jelas *AND* kapasitas telah sesuai dengan yang ditentukan departemen IT

THEN Pengecekan biaya yang dibutuhkan dengan anggaran

Tahap 4

IF *Project manager* sudah mengetahui kebutuhan *user AND* tujuan pembelian perangkat keras (*hardware*) dan perangkat lunak (*software*) sudah jelas



Gambar 4. Flowchart Sistem Pengajuan Spesifikasi Hardware dan Software Pembangunan Rekayasa Perangkat Lunak

AND kapasitas telah sesuai dengan yang ditentukan departemen IT *AND* biaya yang dibutuhkan sesuai dengan anggaran yang ditentukan

THEN Memberi alternatif dan mencetak pengajuan spesifikasi *software* dan *hardware.*

ELSE

Alternatif perangkat keras (*hardware*) dan perangkat lunak (*software*) tidak dapat disusun.

Algoritma tersebut disusun dengan menggunakan aturan *forward chaining,* yaitu dengan melakukan pelacakan ke depan secara berurutan agar mendapat suatu kesimpulan. Dalam kasus pemilihan perangkat keras (*hardware*) dan perangkat lunak (*software*) tersebut, hal yang pertama kali dilakukan dimulai pada proses pertama yaitu pengecekan kebutuhan dan tujuan dari pembangunan rekayasa perangkat lunak (*software engineering*) yaitu pembelian perangkat keras (*hardware*) dan perangkat lunak (*software*). Jika kebutuhan *user* dan tujuan sudah jelas maka dapat mengajukan pemilihan perangkat keras (*hardware*) dan perangkat lunak (*software*) sesuai dengan kebutuhan dan kapasitas yang ditentukan departemen *information technology* pada Bank X tersebut, dimana dalam proses pemilihan perangkat keras (*hardware*) dan perangkat lunak (*software*) harus sesuai dengan anggaran dan telah disetujui oleh *project manager* IT.

Pada Gambar 4 terlihat bahwa jika seluruh batasan pada setiap proses tersebut terpenuhi maka alternatif pemilihan perangkat keras (*hardware*) dan perangkat lunak (*software*) pun akan dihasilkan oleh sistem. Berdasarkan alternatif pemilihan perangkat keras (*hardware*) dan perangkat lunak (*software*) tersebut, maka administrator akan mencari *vendor* yang kompeten dalam pengadaan perangkat keras (*hardware*) dan perangkat lunak (*software*). Apabila tidak memenuhi batasan pada setiap proses tersebut, maka alternatif pemilihan perangkat keras (*hardware*) dan perangkat lunak (*software*) tidak dapat disusun.

Dalam setiap proses tersebut, perlu dilakukan proses mengidentifikasi risiko proyek yang kemudian dilakukan proses pendokumentasian risiko, guna memudahkan dalam proses identifikasi risiko dan status risiko pada setiap proses selama proyek berjalan, sehingga dapat membantu dalam proses pengambilan keputusan untuk proses mitigasi risiko. Daftar risiko tersebut disertakan sebagai bagian dokumentasi dari rencana proyek atau menjadi dokumen yang berdiri sendiri. *Form* untuk mendokumentasikan risiko ditunjukkan pada Tabel 4.

Kolom C, L dan E pada Tabel 4 menyediakan tempat untuk menangkap probabilitas materialisasi risiko menjadi masalah (*consequences*/C), kegagalan yang dapat terjadi akibat risiko (*likelihood*/L), dan semua *risk exposure* (C dikali L). Skala penilaian untuk mengukur kecenderungan (*likelihood*/L) terjadinya kejadian yang merugikan akibat risiko yang teridentifikasi yakni: 5 (sangat sering), 4 (sering), 3 (kadang-

Tabel 4. Contoh *Risk Documentation Form*

ID	*Risk Description*	*C*	*L*	*E*	*First Indicator*	*Risk Mitigation Approach*	*Who*	*Due*
	List each major risk facing the project. Describe each risk in the form "condition – consequence". Example: "Subcontractor's staff does not have sufficient technical expertise, so their work is delayed for training and slowed by learning curve."	*C	*L	*E	*For each risk, describe the earliest indicator or trigger condition that might indicate that the risk is turning into a problem*	• *For each risk, state one or more approaches to control, avoid, minimize, or otherwise mitigate the risk* • *Risk mitigation approaches should yield demonstrable results, so you can measure whether the risk exposure is changing.*	*Assign each risk mitigation to an individual*	*State a date by which the mitigation approach is to be implemented.*

(Sumber: Kristanti, Tanti., 2007)

kadang), 2 (jarang), dan 1 (sangat jarang). Sedangkan skala yang digunakan untuk mengukur konsekuensi (*consequences*/C) risiko yakni: 5 (sangat besar), 4 (besar), 3 (sedang), 2 (kecil), dan 1 (sangat kecil).

Risiko diurutkan secara *descending*, sehingga risiko dengan prioritas utama berada pada daftar yang teratas. Mekanisme prioritas risiko digunakan untuk dapat menentukan dimana posisi bank harus memfokuskan dalam proses kontrol dan mitigasi risiko. Proses kontrol dan mitigasi risiko untuk beberapa risiko, dapat difokuskan pada pengurangan probabilitas atau mengurangi dampak risiko tersebut.

Pada kolom berikutnya dalam *form* merupakan cara untuk mendokumentasikan hasil observasi yang dapat mendasari adanya indikator pertama bahwa suatu faktor risiko akan menjadi masalah untuk saat ini. Kolom *risk mitigation approach* untuk mengidentifikasi langkah-langkah untuk menjaga agar risiko tetap terkendali. Pendokumentasian pernyataan risiko tersebut, dilakukan dengan menggunakan format *condition-consequence* yang menyatakan situasi/kondisi risiko yang diperhatikan, diikuti dengan setidaknya satu *outcome* yang kurang potensial (*consequence*) jika risiko tersebut berubah menjadi masalah. Adapun proses dokumentasi risiko yang telah dilakukan pada penelitian ini, yaitu:

Tabel 5. ***Risk Documentation Form (Hardware)***

ID	*Risk Description*	*C*	*L*	*E*	*First Indicator*	*Risk Mitigation Approach*	*Who*	*Due*
1	Kesalahan pemilihan *vendor hardware*	5	5	25	• *Database vendor hardware* bank belum memiliki catatan *track record vendor*. • *Turnover k*aryawan divisi IT.	• Melakukan dokumentasi, *track record* tingkah laku *vendor*, dan harga yang ditawarkan • Pemberian *training* pada karyawan baru	• Divisi pembelian *hardware* • *Manager* Departemen IT	Setiap melakukan proses pembelian *hardware,* langsung mencatat seluruh data *vendor* ke dalam *database vendor*.
2	Pencurian perangkat keras (*hardware*)	5	4	20	• Aset *hardware* bank hilang • Jumlah aset yang ada dengan pencatatan tidak sama	• Memasang CCTV • Memberikan kode pada setiap aset *hardware* yang dimiliki • Lapor polisi dan memberikan sanksi hukum	• Divisi pembelian *hardware* • *Security* • *Manager* Departemen IT	Setiap melakukan proses pembelian *hardware,* langsung mencatat seluruh data *hardware* ke dalam *database* aset *hardware*.
3	Kualitas perangkat keras (*hardware*) rendah/buruk	4	5	20	• *Database vendor hardware* bank belum memiliki catatan *track record vendor*. • *Turnover* karyawan divisi IT.	• Melakukan dokumentasi, *track record* tingkah laku *vendor*, dan harga yang ditawarkan. • Meminta garansi yang paling lama. • Pemberian *training* pada karyawan baru	• Divisi pembelian *hardware* • *Manager* Departemen IT	Setiap melakukan proses pembelian *hardware,* langsung dilakukan pengecekan kualitas *hardware* dan mencatat seluruh data *vendor* ke dalam *database vendor*.

ID	*Risk Description*	*C*	*L*	*E*	*First Indicator*	*Risk Mitigation Approach*	*Who*	*Due*
4	*Hardware* tidak *compatible*	5	4	20	• *User requirement* yang berubah-ubah • Membeli barang dari *vendor* yang baru/bukan langganan • *Turnover* karyawan divisi IT.	• Meminta garansi yang paling lama kepada *vendor hardware*. • Melakukan retur jika memungkinkan, jika tidak tukar barang. • Pemberian *training* pada karyawan baru	• Divisi pembelian *hardware* • *Vendor hardware* • *Manager* Departemen IT	Setiap melakukan proses pembelian *hardware,* langsung dilakukan pengecekan kompabilitas dari *hardware*.
5	Kerusakan *hardware* karena *human error*	3	5	15	• Baru merekrut karyawan baru • Terjadi bencana (kebakaran, banjir, dan gempa)	• Dilakukan tahap *training* terhadap penggunaan *hardware.* • Meminta garansi yang paling lama kepada *vendor*. • Meminta tanggung jawab pada yang bersangkutan.	• Divisi pembelian *hardware* • *Vendor hardware* • *Manager* Departemen IT	Setiap akhir bulan melakukan proses pengecekan kondisi, penanggung jawab, dan jangka waktu penggunaan *hardware* dan mencatat seluruh data *hardware* ke dalam *database hardware*

Tabel 6. ***Risk Documentation Faorm (Software)***

ID	*Risk Description*	*C*	*L*	*E*	*First Indicator*	*Risk Mitigation Approach*	*Who*	*Due*
1	Terdapat aplikasi yang *error/bug*	5	5	25	• *Error*/*Bug* yang timbul karena sistem yang *crash.* • *Error*/*Bug* yang timbul karena kesalahan *coding.* • *Turnover* karyawan divisi IT.	• Pengecekan dan uji coba sistem aplikasi yang dibangun sebelum diimplementasikan. • Perbaikan aplikasi yang terdapat *error/bug* • Pemberian *training* terhadap tim pengembang perangkat lunak di Departemen IT bank	• *User* • Pengembang rekayasa perangkat lunak • *Manager* Departemen IT	Setiap akhir bulan saat pelaporan *progress* perkembangan pembangunan atau pengembangan rekayasa perangkat lunak.

ID	*Risk Description*	*C*	*L*	*E*	*First Indicator*	*Risk Mitigation Approach*	*Who*	*Due*
2	Kesalahan pemilihan *vendor software*	5	5	25	• *Database vendor software* bank belum memiliki catatan *track record vendor*. • Perkembangan teknologi. • Kebijakan pemerintah dalam bidang IT perbankan • *Turnover* karyawan divisi IT.	• Membuat perjanjian dalam setiap tahap pembangunan atau pengembangan rekayasa perangkat lunak • Melakukan dokumentasi, penggambaran model, dan persetujuan yang ditandatangani kedua belah pihak pada setiap diadakannya *client meeting* dan wawancara	• *User* • Pengembang rekayasa perangkat lunak • *Manager* Departemen IT	Setiap melakukan proses pembelian *software,* langsung mencatat seluruh data *vendor* ke dalam *database vendor*.
3	*User* lamban dan tidak lengkap memberitahukan kebutuhannya (*user requirement*)	4	5	20	• Perkembangan teknologi. • Kebijakan pemerintah dalam bidang IT perbankan • *Turnover* karyawan divisi IT.	• Membagi pembangunan rekayasa perangkat lunak dalam beberapa tahap utama (*batch*) • Membuat perjanjian waktu (*deadline*) dalam setiap tahap pembangunan atau pengembangan rekayasa perangkat lunak • Melakukan dokumentasi, penggambaran model, dan persetujuan yang ditandatangani kedua belah pihak pada setiap diadakannya *client meeting* dan wawancara	• *User* • Pengembang rekayasa perangkat lunak • *Manager* Departemen IT	Dilakukan *meeting* setiap akhir bulan saat pelaporan *progress* perkembangan pembangunan atau pengembangan rekayasa perangkat lunak untuk menyamakan *user requirement*.

ID	*Risk Description*	*C*	*L*	*E*	*First Indicator*	*Risk Mitigation Approach*	*Who*	*Due*
4	*Security data*	4	5	20	• *Error*/*Bug* yang timbul karena sistem yang *crash* • Kapasitas dan fasilitas bank belum memenuhi standar. • Perkembangan teknologi. • Kebijakan pemerintah dalam bidang IT perbankan • *Turnover* karyawan divisi IT.	• Pemberian *password* bagi *user* yang diberikan fasilitas untuk mengakses data nasabah. • Seluruh data-data nasabah disimpan dalam *database* yang dienkripsi. • Seluruh data-data nasabah diberikan proteksi keamanan dari serangan virus, *hacking, spyware,* dan lain sebagiannya dengan cara memberikan *antivirus*, *firewall*, dan lain sebagainya. • Bank memiliki *server* sendiri untuk menyimpan *back-up* data nasabah, sehingga tidak seharusnya bank menyewa pihak eksternal untuk menyimpan data nasabah. • Untuk mengantisipasi *server down* karena serangan *hacker*, bank membuat sistem *back-up* data nasabah setiap jangka waktu yang dekat (misalnya: setiap 15 detik sekali, sistem otomatis melakukan *back-up*). • Setiap komputer yang diberikan fasilitas untuk mengakses data nasabah, tidak diberikan celah untuk meng-*copy* data (misalnya: komputer tidak diberikan akses untuk internet, USB, *disc*, disket, kabel data, dan lain sebagainya). • Setiap komputer yang diberikan fasilitas untuk mengakses data nasabah, harus dapat dimonitor setiap kegiatannya sehingga dapat diketahui siapa saja yang akan mencuri data nasabah	• *User* • Pengembang rekayasa perangkat lunak • *Manager* Departemen IT	Setiap saat (24 jam / 7 hari) dilakukan proses *monitoring* dan kontrol, terhadap seluruh data bank.

ID	*Risk Description*	*C*	*L*	*E*	*First Indicator*	*Risk Mitigation Approach*	*Who*	*Due*
5	*Data breach* (kebocoran data)	5	2	10	• *Error*/*bug* yang timbul karena sistem yang *crash* • Perkembangan teknologi. • *Turnover* *k*aryawan divisi IT.	• Pemberian *password* bagi *user* yang diberikan fasilitas untuk mengakses data nasabah. • Seluruh data-data nasabah disimpan dalam *database* yang dienkripsi. • Seluruh data-data nasabah diberikan proteksi keamanan dari serangan virus, *hacking, spyware,* dan lain sebagiannya dengan cara memberikan *antivirus*, *firewall*, dan lain sebagainya. • Bank memiliki *server* sendiri untuk menyimpan *back-up* data nasabah, sehingga tidak seharusnya bank menyewa pihak eksternal untuk menyimpan data nasabah. • Untuk mengantisipasi *server down* karena serangan *hacker*, bank membuat sistem *back-up* data nasabah setiap jangka waktu yang dekat (misalnya: setiap 15 detik sekali, sistem otomatis melakukan *back-up*). • Setiap komputer yang diberikan fasilitas untuk mengakses data nasabah, tidak diberikan celah untuk meng-*copy* data (misalnya: komputer tidak diberikan akses untuk internet, USB, *disc*, disket, kabel data, dan lain sebagainya). • Setiap komputer yang diberikan fasilitas untuk mengakses data nasabah, harus dapat dimonitor setiap kegiatannya sehingga dapat diketahui siapa saja yang akan mencuri data nasabah.	• *User* • Pengembang rekayasa perangkat lunak • *Manager* Departemen IT	Setiap saat (24 jam / 7 hari) dilakukan proses *monitoring* dan kontrol, terhadap seluruh data bank.

4. Kesimpulan

Berdasarkan pada pembahasan pada bagian sebelumnya, dapat ditarik beberapa kesimpulan.

Pertama, diperlukan batasan utama yang harus dipenuhi dalam pembangunan rekayasa perangkat lunak berdasarkan pemilihan perangkat keras (*hardware*) dan perangkat lunak (*software*) yang dikembangkan, yaitu kebutuhan pengguna, tujuan pembangunan rekayasa perangkat lunak, kapasitas bank, anggaran yang disediakan, pengetahuan, dan kemampuan pengguna

Kedua, sistem pemilihan alternatif perangkat keras dan perangkat lunak menggunakan metode pohon keputusan (*decision tree*) sistem pakar *forward chaining,* dimana metode tersebut diterapkan melalui urutan batasan dan *rules* yang diperoleh melalui metode wawancara beberapa pakar (*expert*), sebagai berikut:

1.) Pengecekan kebutuhan pengguna dan tujuan dari pembangunan rekayasa perangkat lunak (*software engineer*),
2.) Mengajukan pemilihan perangkat keras dan perangkat lunak sesuai dengan kapasitas yang ditentukan Departemen *Information Technology* pada Bank X,
3.) Mengecek pengajuan berdasarkan dengan anggaran, pengetahuan pengguna, dan kemampuan pengguna dalam menggunakan sistem yang akan dibangun,

Ketiga, manajemen risiko merupakan faktor yang sangat penting untuk pembangunan atau pengembangan rekayasa perangkat lunak, yaitu untuk mengurangi *surprise factor*.

Keempat, manajemen risiko berfungsi untuk mengidentifikasi, memberikan arah, dan memitigasi potensi risiko sebelum risiko tersebut dapat menyebabkan proyek pembangunan atau pengembangan rekayasa perangkat lunak menjadi gagal.

Kelima, mengidentifikasi risiko tidak bisa dilakukan secara sederhana karena risiko-risiko tersebut perlu didokumentasikan (*risk documentation*) agar dapat memudahkan komunikasi diantara komunitas *project stakeholder* selama proyek berjalan dan menjadi dokumen untuk pembangunan atau pengembangan rekayasa perangkat lunak perbankan di masa yang akan datang.

Keenam, dengan menerapkan manajemen risiko dalam pembangunan rekayasa perangkat lunak perbankan bukan berarti menghindari proyek yang memiliki risiko tinggi, karena manajemen risiko bertujuan untuk membuka mata para pengembang rekayasa perangkat lunak agar dapat mengetahui risiko mana saja yang dapat berakibat fatal dan melakukan yang terbaik untuk memastikan risiko tersebut tidak akan mencegah kesuksesan proyek.

Ketujuh, berdasarkan hasil analisis risiko pada Bank X, diperoleh beberapa prioritas risiko yang harus diwaspadai dan yang perlu segera dilakukan langkah mitigasi sebagaimana dicantumkan pada Tabel 7 berikut ini.

Tabel 7. Faktor Risiko yang Harus Diwaspadai dan Dimitigasi

Sektor Proyek	*Faktor Risiko*	*Persentase Proyek dalam Risiko*	*Implikasi/ Dampak*
Perangkat Keras *(Hardware)*	Kesalahan pemilihan *vendor hardware*	25%	Tinggi
	Pencurian perangkat keras (*hardware*)	20%	Tinggi
	Kualitas perangkat keras (*hardware*) rendah/buruk	20%	Tinggi
	Hardware tidak *compatible*	20%	Tinggi
	Kerusakan *hardware* karena *human error*	15%	Tinggi
	Total	**100%**	
Perangkat Lunak *(Software)*	Terdapat aplikasi yang *error/bug*	25%	Tinggi
	Kesalahan pemilihan *vendor software*	25%	Tinggi
	User lamban dan tidak lengkap memberitahukan kebutuhannya (*user requirement*)	20%	Tinggi
	Security data	20%	Tinggi
	Data breach (kebocoran data)	10%	Biasa
	Total	**100%**	

Referensi

Boehm, Barry W. (1989). *Software Risk Management, Los Alamitos*, Calif.: IEEE Computer Society Press.

Budisantoso, T dan Sigit. (2006). *Bank dan Lembaga Keuangan Lain*. Edisi 2. Jakarta: Salemba Empat.

Duffield, C & Trigunarsyah, B. (1999). *Project Management-Conception to Completion*. Australia: Engineering Education Australia. (EEA).

Fisk, E.R. (1997). *Construction Project Administration.* Fifth Edition. New Jersey: Prentice Hall.

Flanagan, R & Norman, G. (1993). *Risk Management and Construction.* London: Blackwell Science.

Godfrey, P.S., Sir William Halcrow and Partners Ltd. (1996). *Control of Risk a Guide to Systematic Management of Risk from Construction*. Wesminster London: Construction Industry Research and Information Association (CIRIA).

Hasibuan, Melayu SP. (2005). *Dasar-Dasar Perbankan*. Jakarta: PT. Bumi Aksara.

Jones, Capers. (1994). *Assessment and Control of Software Risks*. Englewood Cliffs, N.J.: PTR Prentice-Hall.

Kerzner, H. (2001). *Project Management.* Seventh Edition. New York: John Wiley & Sons, Inc.

Kristanti, Tanti. (2007). *Manajemen Resiko Software Engineering*.

Kuswandi, dan Mutiara, Ema (2004). *Delapan Langkah dan Tujuh Alat Statistik untuk Peningkatan Mutu Berbasis Komputer*. Jakarta: PT Elex Media Komputindo. (Bab 8 Diagram Tulang Ikan (Fishbone Diagram) hal 79-82.

Labombang, Mastura. (2011). Manajemen Risiko dalam Proyek Konstruksi. *Jurnal SMARTek*, Vol. 9 No. 1. Pebruari 2011: 39 – 46.

Ladjamudin, bin Al-Bahra. (2005). *Analisis dan Desain Sistem Informasi.* Edisi Pertama. Yogyakarta: Graha Ilmu.

McConnell, Steve. (1996). *Rapid Development: Taming Wild Software Schedules*. Redmond, Wash.: Microsoft Press.

Rosa, A. S. dan Shalahuddin. M. (2011). *Modul Pembelajaran Rekayasa Perangkat Lunak (Terstruktur dan Berorientasi Objek)*. Bandung: Modula.

Rosadi, Anisa Ayu. (2009). *Sistem Pendukung Keputusan Teknik.* Bandung: Universitas Islam Bandung.

Tim Penerbit Andi. (2003). *Pengembangan Sistem Pakar Menggunakan Visual Basic.* Edisi Pertama. Yogyakarta: ANDI.

Turban, Efraim, Mclean Ephraim dan Wetherbe James. (2002). *Information Technology for Management.* Edisi Ketiga. John Wiley & Sons, Inc. Hal 423-478.

Wideman, Max.R. (1992). *Project and Program Risk Management: A Guide to Managing Project Risk Opportunities*. Amerika: Project Management Institute.