MENGELOLA RISIKO PROYEK PENGEMBANGAN *SOFTWARE*

(*Managing the Risks of the Software Development Project*)

Endi Putro*, Maria Ariesta

Fakultas Teknik dan Ilmu Komputer Jurusan Sistem Informasi
Universitas Kristen Krida Wacana – Jakarta
***endi.putro@ukrida.ac.id**

Abstrak

Pengembangan *software* disebut sebagai proyek karena terdapat anggaran yang dibatasi dan dengan waktu yang ditentukan dari waktu mulai sampai berakhirnya proyek. Dalam kaitannya dengan pengembangan *software* sebagai suatu proyek, risiko didefinisikan sebagai peristiwa yang tidak ada dalam rencana yang dapat terjadi dan mengakibatkan kerugian proyek. Metodologi yang digunakan untuk mengelola risiko proyek pengembangan *software* adalah memberi pembobotan terhadap matrik yang mempertemukan antara aktivitas pengelolaan proyek dengan pertanyaan-pertanyaan yang disusun berdasarkan faktor-faktor risiko proyek pengembangan *software*. Hasil matrik tersebut digunakan untuk mengantisipasi risiko yang mungkin akan muncul pada saat proyek berjalan.

Kata Kunci: proyek, risiko, mengelola

Abstract

Software Development is considered as a project due to the budget and time restrictions applied to it. In software development project, risk refers to an event that is not in the plan which may occur and result in the project loss. The methodology used to manage the project risks include giving weigth to the matrix that draws together the project management activities and questions generated based on the risk factors of software development projects. The results of these matrixes are used to anticipate the risks that might appear during the project implementation.

***Keywords**: project, risk, managing*

Tanggal Terima Naskah : 19 Juli 2012
Tanggal Persetujuan Naskah : 31 Agustus 2012

1. PENDAHULUAN

Proyek pengembangan *software* adalah proses melakukan siklus dalam menciptakan perangkat lunak untuk sebuah fungsi tertentu [1]. Kegagalan sebuah proyek pengembangan *software* diakibatkan oleh beberapa faktor. Menurut Karolak [2], kurangnya pengetahuan mengenai apa yang terlibat di dalam manajemen *software* merupakan salah satu faktor dari kegagalan proyek pengembangan *software*. Konsep dari manajemen pengembangan dan risiko tidak mudah diajarkan pada sistem pendidikan universitas maupun pada seminar-seminar dan pelatihan. Konsep manajemen yang ditemukan di universitas program bisnis terkadang hanya membahas konsep risiko yang berkaitan dengan keputusan investasi atau asuransi.

Faktor lain dari kegagalan proyek pengembangan *software* adalah kurangnya disiplin dalam menerapkan teknik manajemen yang baik. Seseorang dengan pendidikan manajemen *software* dan risiko yang baik sekalipun bahkan membutuhkan ketelitian dan disiplin yang lebih dalam melakukan identifikasi, perhitungan, penetapan, perencanaan, pengumpulan, dan pelaporan risiko *software*. Hal ini dapat menjadi tugas yang sangat sulit, terutama ketika harus dibatasi oleh tekanan anggaran dan jadwal [3].

Selain itu, *tools* yang diperlukan untuk melakukan manajemen risiko *software* masih sangat kurang. Meskipun *tools* manajemen proyek saat ini sudah banyak tersedia, namun *tools* untuk manajemen risiko *software* tidak mudah ditemukan pada *software* profesional [4]. Terbatasnya pandangan mengenai manajemen risiko *software* dan kegagalan untuk mengintegrasikannya juga menjadi pemicu kegagalan proyek pengembangan *software*.

2. RUMUSAN MASALAH

Berdasarkan uraian yang telah disampaikan di atas, maka dapat dirumuskan masalah sebagai berikut: "Bagaimana mengantisipasi risiko yang mungkin muncul pada saat mengembangkan proyek *software*?"

3. METODOLOGI PENELITIAN

Metodologi penelitian yang digunakan adalah sebagai berikut:

1) Membuat daftar aktivitas risiko yang dilakukan saat mengembangkan *software.*
2) Membuat daftar faktor-faktor risiko yang mungkin muncul saat mengembangkan *software.*
3) Membuat matrik antara aktivitas dan faktor risiko.
4) Pembobotan matrik berdasarkan pertanyaan-pertanyaan pada faktor risiko yang ditentukan.
5) Penghitungan pembobotan untuk menentukan kelayakan proyek menghadapi risiko.

4. GEJALA DAN FAKTOR RISIKO

Menurut Karolak [2], aktivitas risiko merupakan cara melakukan evaluasi terhadap risiko. Terdapat enam aktivitas yang dilakukan dalam mengevaluasi manajemen risiko *software*, yaitu:

1) *Risk Identification*, yaitu melakukan identifikasi gejala risiko yang terjadi.
2) *Risk Strategy and Planning*, yaitu merancang suatu tahapan untuk menanggulangi risiko.
3) *Risk Assessment*, yaitu mengukur akibat dari adanya risiko.
4) *Risk Mitigation/Avoidance*, yaitu melakukan mitigasi dari hasil penilaian risiko.
5) *Risk Reporting*, yaitu membuat penulisan terhadap seluruh kegiatan manajemen risiko sehingga dapat digunakan sebagai dasar analisis manajemen risiko berikutnya.
6) *Risk Prediction*, yaitu membuat perkiraan risiko yang akan terjadi berikutnya dalam pengembangan *software*.

Terdapat 10 (sepuluh) faktor risiko *software*, yaitu:

1) *Organization*, yaitu faktor risiko yang berkaitan dengan kedewasaan dari struktur, komunikasi, fungsi, dan kepemimpinan suatu organisasi.
2) *Estimation*, yaitu faktor risiko yang terfokus pada ketidaktepatan estimasi sumber daya, penjadwalan, dan biaya yang diperlukan untuk mengembangkan *software*. Faktor ini berpengaruh tinggi pada elemen risiko biaya dan penjadwalan apabila

teknik estimasi yang digunakan tidak tepat. Faktor ini berpengaruh rendah terhadap risiko teknis *software*, karena tidak mengarahkan bagaimana dan apa yang akan diimplementasikan dalam produk *software*. Estimasi dianggap sangat penting dalam proses pengembangan *software* karena mempengaruhi jumlah sumber daya manusia dan komputer yang akan tersedia untuk mengembangkan *sofware*.

3) *Monitoring*, yaitu faktor risiko yang berhubungan dengan penentuan masalah dengan melakukan pemantauan.
4) *Development Methodology*, yaitu faktor risiko yang mengidentifikasi metode yang digunakan dalam pengembangan *software*.
5) *Tools*, yaitu faktor risiko yang berkaitan dengan *tools* yang dipergunakan pada pengembangan *software*.
6) *Risk Culture*, yaitu faktor risiko yang terkait dengan proses pertimbangan dan pengambilan keputusan.
7) *Usability*, yaitu faktor risiko yang terkait dengan produk *software* sejak diserahkan kepada pengguna *software*.
8) *Correctness*, yaitu faktor risiko yang juga terkait dengan produk *software* sejak diserahkan kepada pengguna *software*.
9) *Reliability*, yaitu faktor risiko yang berhubungan dengan terbebasnya *software* dari kesalahan eksekusi. Faktor ini berpengaruh tinggi pada elemen risiko teknis *software*. Faktor ini berpengaruh rendah terhadap elemen risiko biaya dan penjadwalan karena *software* telah diserahkan kepada pengguna dan biasanya tidak diukur lagi mengenai biaya dan penjadwalan.
10) *Personnel*, yaitu faktor risiko yang berhubungan dengan kemampuan personil dalam menggunakan metode dan *tools*, serta pengetahuan personil mengenai pengembangan *software*.

5. STUDI KASUS

Proyek membangun Sistem Informasi Rumah Sakit "Sehat Sentosa" adalah sebuah proyek pemesanan *software*. Peruntukan perangkat lunak ini diarahkan bagi kebijakan manajemen dalam menjalankan bisnisnya.

Perangkat lunak dipesan pada jasa *Software House* "Cepat Akurat", selanjutnya proyek ini disusun dari perspektif jasa *Software House* sebagai pelaksana pekerjaan membangun sistem informasi. Dalam menjalankan tugasnya, pihak *Software House* "Cepat Akurat" menugaskan lima orang staf dengan susunan satu orang sebagai kepala proyek yang bertugas untuk menentukan kebijakan proyek, termasuk penentuan jadwal dan anggaran, dua orang staf ahli desain yang bertugas sebagai analisis perancangan sistem informasi di rumah sakit, dua orang *programmer* yang bertugas membuat aplikasi program hasil desain dari analis, dan satu orang pembantu umum.

Instalasi – instalasi rumah sakit yang digunakan sebagai subjek implementasi sistem informasi adalah:

- Instalasi Pendaftaran Pasien
- Instalasi Gawat Darurat
- Instalasi Rawat Inap
- Instalasi Rawat Jalan
- Instalasi Rawat Intensif
- Instalasi Farmasi
- Instalasi Operasi
- Instalasi Gizi
- Instalasi Laboratorium
- Pembayaran Pasien.

Paradigma yang digunakan untuk merancang perangkat lunak menggunakan teknik "*Water Fall*". Asumsi-asumsi dalam perancangan proyek adalah:

- *Software House* "Cepat Akurat" sebagai pelaksana proyek hanya mengerjakan pembangunan perangkat lunak.
- Perangkat keras akan disediakan oleh pihak Rumah Sakit "Sehat Sentosa" seberapapun kebutuhannya.

Berikut hasil perencanaan proyek pengembangan *software* yang disusun menggunakan aplikasi Gann *Chart* berdasarkan waktu penyelesaian dan biaya yang dianggarkan.

	Task Name	Duration	Start	ID	Finish	Predecessors	Resource Names
1	Persiapan	7 days?	Tue 8/14/12	1	Wed 8/22/12		
2	Membuat Proposal Pengajuan Proyek	2 days	Tue 8/14/12	2	Wed 8/15/12		Kepala Proyek
3	Membuat MOU	1 day?	Thu 8/16/12	3	Thu 8/16/12	2	Kepala Proyek
4	Negosiasi MOU	3 days	Fri 8/17/12	4	Tue 8/21/12	3	Kepala Proyek
5	Persetujuan	1 day?	Wed 8/22/12	5	Wed 8/22/12	4	Kepala Proyek
6	Desain	7 days	Thu 8/23/12	6	Fri 8/31/12		
7	Investigasi keperluan SI Rumah Sakit	3 days	Thu 8/23/12	7	Mon 8/27/12	5	Designer[300%],Kepala Proyek
8	Pemodelan hasil investigasi	5 days	Mon 8/27/12	8	Fri 8/31/12	7FS-1 day	Designer
9	Implementasi	43 days?	Mon 9/3/12	9	Wed 10/31/12		
10	Pemrograman Instalasi Pendaftaran Pasien	7 days	Mon 9/3/12	10	Tue 9/11/12	8	Pemrogram
11	Pemrograman Instalasi Gawat Darurat	7 days?	Mon 9/3/12	11	Tue 9/11/12	10SS	Pemrogram
12	PemrogrramanInstalasi Rawat Inap	7 days?	Wed 9/12/12	12	Thu 9/20/12	11	Pemrogram
13	Pemrograman Instalasi Rawat Jalan	7 days?	Wed 9/12/12	13	Thu 9/20/12	12SS	Pemrogram
14	Pemrograman Instalasi Rawat Intensif	7 days	Fri 9/21/12	14	Mon 10/1/12	13	Pemrogram
15	Pemrograman Instalasi Farmasi	7 days	Fri 9/21/12	15	Mon 10/1/12	14SS	Pemrogram
16	Pemrograman Instalasi Operasi	7 days	Tue 10/2/12	16	Wed 10/10/12	15	Pemrogram
17	Pemrograman Instalasi Gizi	7 days	Tue 10/2/12	17	Wed 10/10/12	16SS	Pemrogram[300%]
18	Pemrograman Instalasi Laboratorium	7 days	Tue 10/2/12	18	Wed 10/10/12		Pemrogram
19	Pemrograman Pembayaran Pasien	7 days	Tue 10/2/12	19	Wed 10/10/12	18SS	Pemrogram
20	Entry Data	7 days	Thu 10/11/12	20	Fri 10/19/12	19	Pembantu
21	Pengujian program	7 days	Mon 10/22/12	21	Tue 10/30/12	20	Designer,Pemrogram
22	Install program	1 day	Wed 10/31/12	22	Wed 10/31/12	21	Pemrogram
23	Perawatan	82 days	Mon 10/22/12	23	Tue 2/12/13		
24	Inventarisasi kesalahan	7 days	Mon 10/22/12	24	Tue 10/30/12	21SS	Designer[200%],Pemrogram
25	Perbaikan program	14 days	Tue 10/23/12	25	Fri 11/9/12	24FS-6 days	Pemrogram
26	Install perbaikan program	7 days	Mon 11/12/12	26	Tue 11/20/12	25	Pemrogram
27	Garansi	60 days	Wed 11/21/12	27	Tue 2/12/13	26	Kepala Proyek

Gambar 1. Rencana waktu penyelesaian proyek pengembangan *software*

	Task Name	Fixed Cost	Fixed Cost Accrual	Total Cost	Baseline	Variance	Actual	Remaining
1	Persiapan	Rp0.00	End	Rp6,500,000.00	Rp0.00	Rp6,500,000.00	Rp0.00	Rp6,500,000.00
2	Membuat Proposal Pengajuan Proyek	Rp5,000,000.00	End	Rp5,000,000.00	Rp0.00	Rp5,000,000.00	Rp0.00	Rp5,000,000.00
3	Membuat MOU	Rp500,000.00	End	Rp500,000.00	Rp0.00	Rp500,000.00	Rp0.00	Rp500,000.00
4	Negosiasi MOU	Rp500,000.00	End	Rp500,000.00	Rp0.00	Rp500,000.00	Rp0.00	Rp500,000.00
5	Persetujuan	Rp500,000.00	End	Rp500,000.00	Rp0.00	Rp500,000.00	Rp0.00	Rp500,000.00
6	Desain	Rp0.00	End	Rp15,000,000.00	Rp0.00	Rp15,000,000.00	Rp0.00	Rp15,000,000.00
7	Investigasi keperluan SI Rumah Sakit	Rp5,000,000.00	End	Rp5,000,000.00	Rp0.00	Rp5,000,000.00	Rp0.00	Rp5,000,000.00
8	Pemodelan hasil investigasi	Rp10,000,000.00	End	Rp10,000,000.00	Rp0.00	Rp10,000,000.00	Rp0.00	Rp10,000,000.00
9	Implementasi	Rp0.00	End	Rp225,000,000.00	Rp0.00	Rp225,000,000.00	Rp0.00	Rp225,000,000.00
10	Pemrograman Instalasi Pendaftaran Pasie	Rp20,000,000.00	End	Rp20,000,000.00	Rp0.00	Rp20,000,000.00	Rp0.00	Rp20,000,000.00
11	Pemrograman Instalasi Gawat Darurat	Rp20,000,000.00	End	Rp20,000,000.00	Rp0.00	Rp20,000,000.00	Rp0.00	Rp20,000,000.00
12	PemrogrramanInstalasi Rawat Inap	Rp20,000,000.00	End	Rp20,000,000.00	Rp0.00	Rp20,000,000.00	Rp0.00	Rp20,000,000.00
13	Pemrograman Instalasi Rawat Jalan	Rp20,000,000.00	End	Rp20,000,000.00	Rp0.00	Rp20,000,000.00	Rp0.00	Rp20,000,000.00
14	Pemrograman Instalasi Rawat Intensif	Rp20,000,000.00	End	Rp20,000,000.00	Rp0.00	Rp20,000,000.00	Rp0.00	Rp20,000,000.00
15	Pemrograman Instalasi Farmasi	Rp20,000,000.00	End	Rp20,000,000.00	Rp0.00	Rp20,000,000.00	Rp0.00	Rp20,000,000.00
16	Pemrograman Instalasi Operasi	Rp20,000,000.00	End	Rp20,000,000.00	Rp0.00	Rp20,000,000.00	Rp0.00	Rp20,000,000.00
17	Pemrograman Instalasi Gizi	Rp20,000,000.00	End	Rp40,000,000.00	Rp0.00	Rp40,000,000.00	Rp0.00	Rp40,000,000.00
18	Pemrograman Instalasi Laboratorium	Rp20,000,000.00	End	Rp20,000,000.00	Rp0.00	Rp20,000,000.00	Rp0.00	Rp20,000,000.00
19	Pemrograman Pembayaran Pasien	Rp20,000,000.00	End	Rp20,000,000.00	Rp0.00	Rp20,000,000.00	Rp0.00	Rp20,000,000.00
20	Entry Data	Rp10,000,000.00	End	Rp10,000,000.00	Rp0.00	Rp10,000,000.00	Rp0.00	Rp10,000,000.00
21	Pengujian program	Rp10,000,000.00	End	Rp10,000,000.00	Rp0.00	Rp10,000,000.00	Rp0.00	Rp10,000,000.00
22	Install program	Rp5,000,000.00	End	Rp5,000,000.00	Rp0.00	Rp5,000,000.00	Rp0.00	Rp5,000,000.00
23	Perawatan	Rp0.00	End	Rp60,000,000.00	Rp0.00	Rp60,000,000.00	Rp0.00	Rp60,000,000.00
24	Inventarisasi kesalahan	Rp10,000,000.00	End	Rp10,000,000.00	Rp0.00	Rp10,000,000.00	Rp0.00	Rp10,000,000.00
25	Perbaikan program	Rp20,000,000.00	End	Rp20,000,000.00	Rp0.00	Rp20,000,000.00	Rp0.00	Rp20,000,000.00
26	Install perbaikan program	Rp10,000,000.00	End	Rp10,000,000.00	Rp0.00	Rp10,000,000.00	Rp0.00	Rp10,000,000.00
27	Garansi	Rp20,000,000.00	End	Rp20,000,000.00	Rp0.00	Rp20,000,000.00	Rp0.00	Rp20,000,000.00

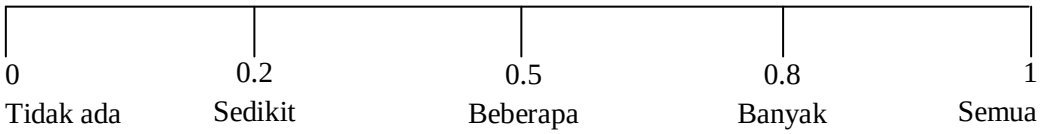
Gambar 2. Rencana anggaran proyek pengembangan *software*

Hasil laporan proyek pengembangan *software* sebagai berikut:

- Total anggaran: Rp 286.500.000
- Waktu yang diperlukan untuk menyelesaikan proyek: 6 bulan
  Mulai: 12 Agustus 2012
  Selesai: 12 Februari 2013

6. ANALISIS

Analisis dilakukan dengan menghitung risiko berdasarkan sepuluh faktor risiko yang telah disebutkan. Analisis ini hanya melakukan perhitungan atas dua faktor risiko, yaitu *estimation* dan *realibility* [5]. Faktor risiko proyek diukur melalui daftar pertanyaan dan dijawab dengan bobot angka yang sesuai dengan proyek bersangkutan. Berikut adalah skala yang digunakan untuk menentukan bobot dari setiap pertanyaan pada masing-masing faktor risiko:



Gambar 3. Skala pembobotan faktor risiko

Beberapa pertanyaan berikut digunakan untuk mengukur faktor risiko *estimation* pada suatu proyek pengembangan *software*:

E1) Metode estimasi apa yang digunakan?
   a) Menebak
   b) Analogi atau perbandingan
   c) *Price to win*
   d) Ditentukan oleh keadaan
   e) *Top-down*
   f) *Bottom-up*
   g) Lainnya.

   Berdasarkan metode yang dipilih, bobot dari masing-masing metode adalah: a) 0.2, b) 0.6, c) 0.3, d) 0.3, e) 0.5, f) 0.7, g) 0.4.

E2) Apakah menggunakan sebuah model biaya *software*?
   Bobot 0 menunjukkan tidak ada model biaya *software* yang digunakan. Bobot 0.5 menunjukkan model biaya digunakan tetapi mungkin belum digunakan dengan benar. Bobot 1 menunjukkan proyek menggunakan sebuah model biaya yang secara akurat mencerminkan proyek.

E3) Apakah estimasi dilakukan berdasarkan matriks produktivitas proyek pengembangan *software* sebelumnya?
   Bobot 0 menunjukkan estimasi tidak dilakukan berdasarkan matriks produktivitas proyek pengembangan *software* sebelumnya. Bobot 0.5 menunjukkan estimasi dilakukan berdasarkan matriks produktivitas proyek *software* sebelumnya, tetapi tidak sama dengan proyek pengembangan *software* yang diestimasi saat ini. Bobot 1 menunjukkan estimasi dilakukan berdasarkan matriks produktivitas yang sama dengan proyek *software* sebelumnya.

E4) Apakah jadwal diestimasi berdasarkan matriks produktivitas proyek pengembangan *software* sebelumnya?
   Bobot 0 menunjukkan jadwal diestimasi berdasarkan matriks produktivitas proyek pengembangan *software* sebelumnya. Bobot 0.5 menunjukkan jadwal

diestimasi berdasarkan matriks produktivitas proyek *software* sebelumnya, tetapi tidak sama dengan jadwal proyek pengembangan *software* yang diestimasi saat ini. Bobot 1 menunjukkan jadwal diestimasi berdasarkan matriks produktivitas yang sama dengan proyek *software* sebelumnya.

E5) Apakah estimasi direvisi setiap bulan atau lebih sering?
Bobot 0 menunjukkan estimasi tidak direvisi. Bobot 0.5 menunjukkan estimasi direvisi, tetapi tidak sampai setiap bulan. Bobot 1 menunjukkan estimasi direvisi setiap bulan atau lebih sering lagi.

E6) Bagaimana keakuratan estimasi biaya pada proyek *software* sebelumnya dibandingkan dengan biaya yang sebenarnya?
Bobot 0 menunjukkan biaya yang diestimasi sekitar 100% atau lebih dari biaya yang sebenarnya. Bobot 0.5 menunjukkan biaya yang diestimasi sekitar 50% dari biaya yang sebenarnya. Bobot 1 menunjukkan biaya yang diestimasi sekitar 5% dari biaya yang sebenarnya.

E7) Bagaimana keakuratan estimasi jadwal pada proyek *software* sebelumnya dibandingkan dengan jadwal yang sebenarnya?
Bobot 0 menunjukkan jadwal yang diestimasi sekitar 100% atau lebih dari jadwal yang sebenarnya. Bobot 0.5 menunjukkan jadwal yang diestimasi sekitar 50% dari jadwal yang sebenarnya. Bobot 1 menunjukkan jadwal yang diestimasi sekitar 5% dari jadwal yang sebenarnya.

Tabel 1 menunjukkan enam akivitas risiko dari manajemen risiko *software* yang diaplikasikan dalam suatu matriks pertanyaan yang mengukur estimasi.

Tabel 1. Matriks faktor risiko *estimation* terhadap aktivitas risiko

Risk Activities	***Metrics***						
	E1	**E2**	**E3**	**E4**	**E5**	**E6**	**E7**
Identification	X	X	X	X	X	X	X
Strategy and Planning	X	X	X	X	X	X	X
Assessment		X	X	X	X	X	X
Mitigation/Avoidance		X	X	X	X		
Reporting					X		
Prediction	X	X	X	X	X	X	X

Beberapa pertanyaan berikut digunakan untuk mengukur faktor risiko *reliability* pada suatu proyek pengembangan *software*:

R1) Apakah ada penanganan kondisi *error* dalam kode dan desain *software*?
Bobot 0 menunjukkan tidak ada penanganan kondisi *error* dalam kode dan desain *software*. Bobot 0.5 menunjukkan ada penanganan kondisi *error* tetapi tidak di setiap kemungkinan terjadinya *error*. Bobot 1 menunjukkan adanya penanganan kondisi *error* dalam kode dan desain *software* untuk setiap kemungkinan terjadinya *error*.

R2) Ketika kondisi *error* terdeteksi, apakah proses tetap berlanjut?
Bobot 0 menunjukkan bahwa ketika kondisi *error* terdeteksi, proses dihentikan atau tidak berlanjut. Bobot 0.5 menunjukkan bahwa pada beberapa kondisi *error* terdeteksi, proses tetap berlanjut. Bobot 1 menunjukkan bahwa pada setiap kondisi *error* terdeteksi, proses tetap berlanjut.

R3) Apakah ada toleransi kesalahan untuk data *input* dan data *output*?
Bobot 0 menunjukkan tidak adanya toleransi kesalahan untuk data *input* dan data *output*. Bobot 0.5 menunjukkan adanya toleransi kesalahan namun hanya untuk beberapa data *input* dan data *output*. Bobot 1 menunjukkan adanya toleransi kesalahan untuk semua data *input* dan data *output*.

R4) Apakah ada pengecekan validitas *input* terlebih dahulu sebelum melakukan suatu proses?
Bobot 0 menunjukkan tidak ada pengecekan validitas *input* terlebih dahulu sebelum melakukan suatu proses. Bobot 0.5 menunjukkan pengecekan validitas dilakukan namun hanya terhadap beberapa *input* sebelum melakukan suatu proses. Bobot 1 menunjukkan ada pengecekan validitas pada semua *input* terlebih dahulu sebelum melakukan suatu proses.

R5) Apakah kesalahan *hardware* (perangkat keras) dapat terdeteksi dan diproses dalam *software*?
Bobot 0 menunjukkan kesalahan *hardware* tidak dapat terdeteksi dan diproses dalam *software*. Bobot 0.5 menunjukkan beberapa kesalahan *hardware* dapat dideteksi dan diproses dalam *software*. Bobot 1 menunjukkan setiap kesalahan *hardware* dapat dideteksi dan diproses dalam *software*.

R6) Apakah penggunaan tipe data global diminimalisir?
Bobot 0 menunjukkan tipe data global sangat banyak digunakan didalam *software*. Bobot 0.5 menunjukkan adanya campuran antara tipe data global dan tipe data lokal. Bobot 1 menunjukkan hanya ada sedikit atau bahkan tidak ada tipe data global yang digunakan didalam *software*.

R7) Apakah kekurangan data dikumpulkan sepanjang integrasi *software*?
Bobot 0 menunjukkan kekurangan data tidak dikumpulkan sepanjang integrasi *software*. Bobot 0.5 menunjukkan beberapa kekurangan data dikumpulkan sepanjang integrasi *software*. Bobot 1 menunjukkan semua kekurangan data dikumpulkan sepanjang integrasi *software*.

R8) Apakah data rusak yang *login* akan ditutup sebelum dikirimkan kepada pengguna?
Bobot 0 menunjukkan tidak ada data rusak yang *login* akan ditutup sebelum dikirimkan kepada pengguna. Bobot 0.5 menunjukkan beberapa data rusak yang *login* akan ditutup sebelum dikirimkan kepada pengguna. Bobot 1 menunjukkan semua data rusak yang *login* akan ditutup sebelum dikirimkan kepada pengguna.

R9) Apakah model kehandalan *software* digunakan untuk memprediksi kehandalan?
Bobot 0 menunjukkan tidak ada model kehandalan *software* yang digunakan untuk memprediksi kehandalan. Bobot 0.5 menunjukkan beberapa model kehandalan *software* digunakan untuk memprediksi kehandalan. Bobot 1 menunjukkan model kehandalan *software* digunakan untuk memprediksi kehandalan.

R10) Apakah pengujian dilakukan dengan perencanaan?
Bobot 0 pengujian dilakukan tanpa adanya perencanaan. Bobot 0.5 menunjukkan beberapa pengujian dilakukan dengan perencanaan. Bobot 1 menunjukkan semua pengujian dilakukan dengan perencanaan.

R11) Apakah pengujian kepentingan telah dilakukan?
Bobot 0 menunjukkan pengujian kepentingan tidak dilakukan. Bobot 0.5 menunjukkan beberapa pengujian kepentingan dilakukan. Bobot 1 menunjukkan pengujian kepentingan dilakukan pada semua bagian *software*.

R12) Apakah pengujian dilakukan oleh kelompok yang terpisah dari kelompok pengembang *software*?
Bobot 0 menunjukkan pengujian dilakukan oleh kelompok pengembang *software*. Bobot 0.5 menunjukkan sebagian dari *software* diuji oleh kelompok yang terpisah dan bagian lainnya diuji oleh kelompok pengembang *software*. Bobot 1 menunjukkan semua bagian *software* diuji oleh kelompok yang terpisah dari kelompok pengembang *software*.

Tabel 2 menunjukkan enam akivitas risiko dari manajemen risiko *software* yang diaplikasikan dalam suatu matriks pertanyaan yang mengukur kehandalan.

Tabel 2. Matriks faktor risiko *reability* terhadap aktivitas risiko

Risk Activities	***Metrics***											
	R1	**R2**	**R3**	**R4**	**R5**	**R6**	**R7**	**R8**	**R9**	**R10**	**R11**	**R12**
Identification	X	X	X	X	X	X	X	X	X	X	X	X
Strategy and Planning												
Assessment	X		X	X	X	X	X	X	X	X	X	X
Mitigation/ Avoidance								X	X	X	X	X
Reporting												
Prediction	X	X	X	X	X	X	X	X	X	X	X	X

7. HASIL DAN PEMBAHASAN

Pada penelitian ini, diterapkan penghitungan manajemen risiko dari proyek pengembangan *software* sistem informasi rumah sakit. Sistem tersebut dirancang untuk memudahkan manajemen data pasien rumah sakit dan meningkatkan efisiensi kinerja rumah sakit dalam melayani pasien. *Software* tersebut direncanakan akan dibuat dalam jangka waktu 6 (enam) bulan dengan memperkerjakan sepuluh sumber daya manusia. Nilai dari proyek tersebut sebesar Rp 285.500.000,-.Untuk menghindari kegagalan dalam proyek tersebut, dilakukan analisis risiko pada tahap awal proyek pengembangan *software*.

Tabel 3. Matriks faktor risiko *estimation* terhadap aktivitas risiko

Risk Activities	***Metrics***						
	E1	**E2**	**E3**	**E4**	**E5**	**E6**	**E7**
Identification	0.6	0.5	0.9	0.8	0.8	0.5	0.6
Strategy and Planning	0.6	0.5	0.9	0.8	0.8	0.5	0.6
Assessment		0.5	0.9	0.8	0.8	0.5	0.6
Mitigation/Avoidance		0.5	0.9	0.8	0.8		
Reporting					0.8		
Prediction	0.6	0.5	0.9	0.8	0.8	0.5	0.6

Tabel 4. Matriks faktor risiko *reability* terhadap aktivitas risiko

Risk Activities	***Metrics***											
	R1	**R2**	**R3**	**R4**	**R5**	**R6**	**R7**	**R8**	**R9**	**R10**	**R11**	**R12**
Identification	0.7	0.5	0.4	0.7	0.3	0.8	0.6	0.5	0.5	0.4	0.5	0.5
Strategy and Planning												
Assessment	0.7		0.4	0.7	0.3	0.8	0.6	0.5	0.5	0.4	0.5	0.5
Mitigation/ Avoidance								0.5	0.5	0.4	0.5	0.5
Reporting												
Prediction	0.7	0.5	0.4	0.7	0.3	0.8	0.6	0.5	0.5	0.4	0.5	0.5

Berdasarkan kedua matriks di atas, penghitungan risiko dapat dilakukan dengan menggunakan persamaan berikut:

1) $P(A_5) = [\sum_{n=1}^{7} P(En)] / 7$

   Dimana En adalah nilai matriks untuk nomor pertanyaan mengenai faktor risiko *estimation* yang berjumlah 7 pertanyaan.

2) $P(A_{12}) = [\sum_{n=1}^{12} P(Rn)] / 12$

   Dimana Rn adalah nilai matriks untuk nomor pertanyaan mengenai faktor risiko *reliability* yang berjumlah 12 pertanyaan.

Untuk melihat kesuksesan suatu proyek, nilai P(A) terletak antara nilai kemungkinan 0 (paling buruk) hingga nilai kemungkinan 1 (sukses). Berdasarkan perhitungan dari persamaan dan nilai bobot matriks yang telah ditentukan sebelumnya, maka dapat diketahui nilai kemungkinan *estimation* $P(A_5)$, yaitu 0.67 dan *reliability* $P(A_{12})$, yaitu 0.55.Nilai kemungkinan 0.67 pada faktor risiko *estimation* menyatakan bahwa proyek memiliki tingkat kesuksesan atau keberhasilan yang cukup baik. Nilai kemungkinan 0.55 pada faktor risiko *reliability* menyatakan bahwa proyek memiliki tingkat kesuksesan menengah.

8. KESIMPULAN

Manajemen risiko membantu manajer proyek untuk mengenali setiap risiko proyek dan memberikan pengetahuan dalam mengelola, mengukur, menilai, dan memprediksi risiko dengan menggunakan metodologi proses dan alat bantu berupa matriks. Kemungkinan risiko yang akan menurunkan tingkat keberhasilan suatu proyek pengembangan *software* harus diantisipasi.Antisipasi dapat dilakukan dengan memenuhi hal-hal yang diperlukan pada masing-masing faktor risiko. Proyek yang baik akan memberikan nilai bobot dari masing-masing pertanyaan dalam faktor risiko dengan angka 1 atau mendekati angka 1.

REFERENSI

[1]. Gray,C.F. dan Larson,E.W, "*Project Management: The Management Process"*, Irwin McGraw- Hill, Boston, SI, 2000.

[2]. Karolak, Dale Walter, "*Software Engineering Risk Management*", IEEE Computer Society Press, California, 1999.

[3]. Charette, Robert N., "*Applications Strategies for Risk Analysis"*, Mc. Graw-Hill, New York, 1989.

[4]. Crockford, Neil, "*An Introduction to Risk Management"*, Woodhead-Faulkner, Cambridge, 1980.

[5]. Davis, Alan M., "*Software Requirements Analysis & Specification"*. Englewood Cliffs, Prentice-Hall, 1990.