Analisa Risiko Proyek Pengembangan Software Pada CV. XYZ

Nicolas Adriaan Apriatono[1], Adi Wibowo[2], Ibnu Gunawan[3]
Program Studi Teknik Informatika, Fakultas Teknologi Industri, Universitas Kristen Petra
Jln. Siwalankerto 121 – 131 Surabaya 60236
Telp. (031)-2983455, Fax. (031)-8417658
E-mail: nic.apriatono@gmail.com[1], adiw@petra.ac.id[2], ibnu@petra.ac.id[3]

ABSTRAK

CV. XYZ adalah suatu perusahaan berukuran kecil yang bergerak di bidang rekayasa perangkat lunak. Jumlah anggota perusahaan ini tidaklah banyak, berkisar antara 5-8 orang saja. Permasalahan yang tejadi pada perusahaan adalah tidak adanya identifikasi terhadap risiko-risiko yang mungkin terjadi.

Masalah-masalah yang pernah terjadi contohnya adalah klien yang tiba-tiba menambah fitur pada *software* pesanan, tidak ada *milestone* pada proyek, tidak adanya data tentang *system* milik klien dan tidak ada monitoring pada proyek yang sedang berjalan. Hal-hal ini dapat menghambat kinerja perusahaan untuk bekerja. Untuk itu dibutuhkan suatu analisis risiko, yang bertujuan menganalisis faktor-faktor risiko apa saja yang mengganggu pengembangan *software* dan respon terhadap risiko-risiko yang terjadi.

Pada skripsi ini, dilakukan penjelasan cara kerja perusahaan, mencari risiko-risiko yang ada, penilaian terhadap setiap risiko yang ada dan respon terhadap risiko-risiko itu. Proses *risk assessment* dilakukan berdasarkan NIST 800-30 yang menjelaskan tentang sepuluh langkah *risk assesment*, penentuan risiko berdasarkan ISO 29110 tentang cara membuat *software* pada perusahaan yang bergerak dalam rekayasa perangkat lunak yang beranggotakan kurang dari 25 orang, dan *OWASP Risk Rating Methodology* tentang penentuan bobot setiap risiko berdasarkan kriteria tertentu. *OWASP* dipakai sebagai acuan untuk menentukan bobot setiap risiko yang sudah ditemukan sebelumnya dengan memakai ISO 29110. Berdasarkan analisa, metode-metode yang dipakai berguna untuk mencari dan merespon risiko-risiko yang ada. Hasil analisa menunjukkan bahwa terdapat 1 risiko *high,* 2 risiko *medium,* dan 19 risiko *low*. *High risk* yang dihadapi perusahaan adalah tidak adanya identifikasi risiko perusahaan yang menyebabkan perusahaan tidak tahu risiko yang mungkin terjadi. Respon risiko tersebut adalah *avoid* dengan cara melakukan identifikasi risiko.

Kata Kunci: Analisa Risiko Software, ISO 29110, OWASP, NIST 800-30, Proyek Pengembangan Software

ABSTRACT

CV.XYZ is a small size company that works on software engineering.The worker in this company is not many, between 5 – 8 people only. The problem in this company is there are no identification of risks that can happen.

The examples of the problems are the clients suddenly request some features for their software, no milestone on the project, no data about clients' system and no monitoring on the ongoing projects. These things can hold back the company's performance. Thus, risk analysis is needed for analyzing risk faktors that can disturb software development.

In this thesis, identification on how the company works is performed, analysing for any existing risk and response for those risks. The risk assessment process is done based on NIST 800-30 that explains about ten steps of risk assessment, determining risks based on ISO 29110 about how to create software on a company that works in software engineering with less than 25 workers in it, and OWASP Risk Rating Methodology about determining the value of every risk based on certain criterias. OWASP is used as guidelines for determining weight of each risks that has been found using ISO 29110. Based on analysis, those methods used are useful for searching and responding existing risks. Result shows 1 high risk, 2 medium risks and 19 low risks. For high risk there is no risk identification on the company that makes the company does not know what risk can impact them. The response is avoid by doing risk identification.

***Keywords**: Software Risk Analysis, ISO 29110, OWASP, NIST 800-30, Software Engineering Project*

1. PENDAHULUAN

Banyak perusahaan memiliki departemen IT. Departemen IT merupakan salah satu bagian penting dalam perusahaan, karena lewat departemen IT suatu perusahaan bisa meminta dibuatkan sebuah software yang sesuai dengan perusahaan itu. Jika perusahaan tidak memiliki sumberdaya atau departemen IT, suatu perusahaan bisa meminta perusahaan pembuat software atau yang biasa disebut sebagai softwarehouse untuk membuat software sesuai keinginan mereka.

Dalam setiap proses pembuatan software, pasti akan selalu ada risiko yang dapat terjadi. Risiko itu bisa dihindari, ditanggulangi atau diterima. Risiko – risiko itu juga memiliki dampak yang berbeda satu dengan lainnya. CV. XYZ merupakan perusahaan yang bergerak dalam bidang pembuatan software atau yang biasa disebut sebagai softwarehouse. Perusahaan ini juga tidak luput dari risiko – risiko ini.

Dari permasalahan diatas akan dibuat sebuah analisa risiko yang berguna untuk menganalisa risiko – risiko yang mungkin terjadi dan mencari cara untuk menanggulanginya.

2. DASAR TEORI

2.1. Software Engineering

Ada beberapa contoh *software engineering* yang ada:

1. *Waterfall* : Diciptakan oleh William Royce. Dalam kenyataannya, model siklus ini sangat sulit untuk diterapkan, karena dibutuhkan sebuah koordinasi yang baik dari tim perangkat lunak, serta kerjasama yang toleratif antara pihak pengembang dengan pihak pengguna [6].
2. Spiral : Diawali dari perencanaan dari perangkat lunak itu sendiri, yang didalamnya termasuk waktu pengerjaan,

sumber daya yang dibutuhkan dan informasi menyangkut pengerjaan proyek. Tahap berikutnya adalah analisa risiko. Dilanjutkan dengan proses pembuatan software. Saat proses pembuatan software dianggap selesai, maka masuk tahap construction and release [6].

3. *Rapid Aplication Development* : Suatu proses pembuatan software yang menitik beratkan pada waktu pembuatan yang sangat singkat
4. *Prototyping* : Prototyping dimulai dengan pengumpulan kebutuhan klien lalu dievaluasi dan setelah itu dijadikan sebagai dasar pembuatan *software.*

2.2. Metodologi Analisa Risiko

- Metodologi analisa kuantitatif, yaitu metodologi yang berkisar antara proses mengoleksi, menganalisa, menginterpretasi dan menulis hasil dari sebuah penelitian yang menitikberatkan pada kuantitas (angka). Secara garis besar, data yang dihasilkan adalah data dalam bentuk angka [3]. Satu hal yang mencolok dalam analisa kuantitatif adalah penggunaan *survey* yang menyediakan data numerik terhadap *trend*, kebiasaan dari sebuah populasi. Untuk koleksi data memakai kuesioner atau *interview.*

- Metodologi analisa kualitatif, yaitu metodologi yang menitik beratkan pada koleksi data, analisa interpretasi dan laporan yang berbeda dengan metode kuantitatif. Memakai sampel, koleksi dari data yang *open-ended*, analisa teks atau gambar, represenasi informasi yang berbentuk figur dan tabel merupakan ciri khas dari metode kualitatif. Secara garis besar, data yang dihasilkan dari analisa kualitatif adalah berbentuk teks [3]. Salah satu bentuk metode kualitatif yang dipakai adalah *case study*, dimana peneliti melakukan analisa terhadap suatu kasus tertentu.

- Metodologi campuran, merupakan metodologi yang menggabungkan metodologi kuantitatif dan kualitatif. Metode ini menggabungkan *survey* yang dilakukan pada kuantitatif untuk menentukan strata atau tingkatan yang didapat dari metode kualitatif [3]. Dalam metode ini, data yang dikumpulkan berbentuk angka dan teks atau gambar. Berikut contoh model inti dalam metodologi campuran menurut [2]:

  - *Convergent parallel mixed method*: Metodologi campuran dimana peneliti menggabungkan data kuantitatif dan kualitatif untuk menyediakan analisa pada permasalahan.

  - *Explanatory sequential mixed method*: Metodologi dimana peneliti melakukan metodologi kuantitatif, menganalisa hasilnya dan kemudian membuat hasil penelitian untuk menjelaskan hasil lebih detail denan metode kualitatif. Disebut sequential karena pada fase kuantitatif diikuti oleh fase kualitatif

  - *Exploratory sequential mixed method*: Kebalikan dari *explanatory sequential method*, dimana peneliti memulai penelitian dengan metode kualitatif. Data yang didapat lalu dianalisa untuk dipakai dalam fase berikutnya yaitu metode kuantitatif.

2.3. Risk Assessment

Risk assesment adalah sebuah cara untuk menentukan risiko – risiko yang mungkin terjadi dalam manajemen risiko. Menurut NIST ada 10 langkah dalam risk assesment yang harus dijalani:

1. *System Characterization :* Menentukan batasan – batasan dari sistem IT yang sedang dipakai. Metode dalam mengumpulkan data – data karakteristik adalah dengan kuesioner dan interview.
2. *Threat Identification :* Mengidentifikasi sumber ancaman yang mungkin terjadi, dimana ancaman dari manusia merupakan ancaman terbesar yang mungkin terjadi.
3. *Vulnerability Identification :* Merupakan identifikasi kelemahan dalam sebuah sistem, misalnya pada bagian desain, implementasi.
4. *Control Analysis* : Tujuan dari langkah ini adalah menganalisa kontrol yang sudah diimplementasi untuk meminimalisasi kemungkinan dari ancaman yang akan mengeksploitasi kelemahan yang ada.
5. *Threat Source / Vulnerability Pairs* : Menentukan sumber ancaman beserta kelemahannya yang menjadi perhatian terbesar. Kelemahan tanpa sumber ancaman bukan merupakan risiko, begitu juga sebaliknya.
6. *Likelihood Determination* : Menetukan kemungkinan dari kelemahan – kelemahan yang ada untuk dieksploitasi.
7. *Impact Analysis :* Menentukan dampak yang berasal dari latihan ancaman dari setiap pasangan sumber ancaman/kelemahan yang menjadi perhatian.
8. *Risk Determination* : untuk menentukan risiko yang ada dengan menggunakan *risk-level matrix.* Tabel 1 menggambarkan bentuk *risk-level matrix*
9. *Control Recommendations :* Selesaikan *risk assesment* ini dengan menentukan kontrol apa yang harus dilakukan untuk meminimalisasi risiko dari pasangan ancaman/kelemahan yang menjadi perhatian utama.
10. *Result Documentation* : Laporan yang dibuat dari *risk assesment* harus memiliki detail yang cukup sehingga memungkinkan manajemen dari perusahaan untuk menetukan langkah yang harus diambil dari risiko – risiko yang sudah diidentifikasi.

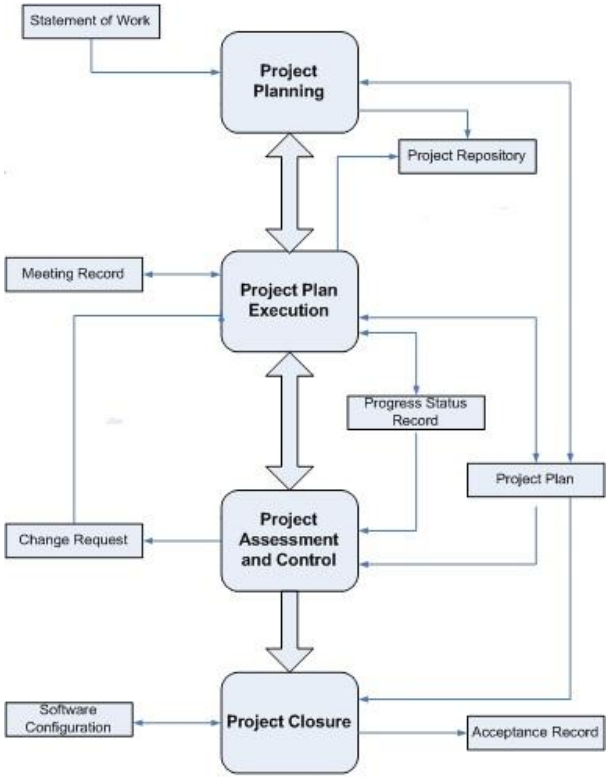
Tabel 1. *Risk-level matrix*

	Impact		
Threat Likelihood	*Low (10)*	*Medium (50)*	*High (100)*
High (1.0)	*Low* 10 x 1.0 = 10	*Medium* 50 x 1.0 = 50	*High* 100 x 0.1 = 100
Medium (0.5)	*Low* 10 x 0.5 = 5	*Medium* 50 x 0.5 = 25	*Medium* 100 x 0.5 = 50
Low (0.1)	*Low* 10 x 0.1 = 1	*Low* 50 x 0.1 = 5	*Low* 100 x 0.1 = 10

2.4. ISO/IEC 29110

ISO / IEC 29110 adalah sebuah standar internasional dari ISO (*International Organization for Standardization*) yang berfokus terhadap rekayasa sistem dan *software – lifecycle profiles* pada VSE (*Very Small Entities*) dimana VSE yang dimaksud adalah sebuah perusahaan, organisasi, departemen atau proyek yang beranggotakan hingga 25 orang saja [4]. Ada 3 peran dasar yaitu PM (*Project Manager*), WT (*Work Team*), dan CS (*Customer*).

Gambar 1 menunjukkan proses kerja bagian pertama yaitu *Project Management.*



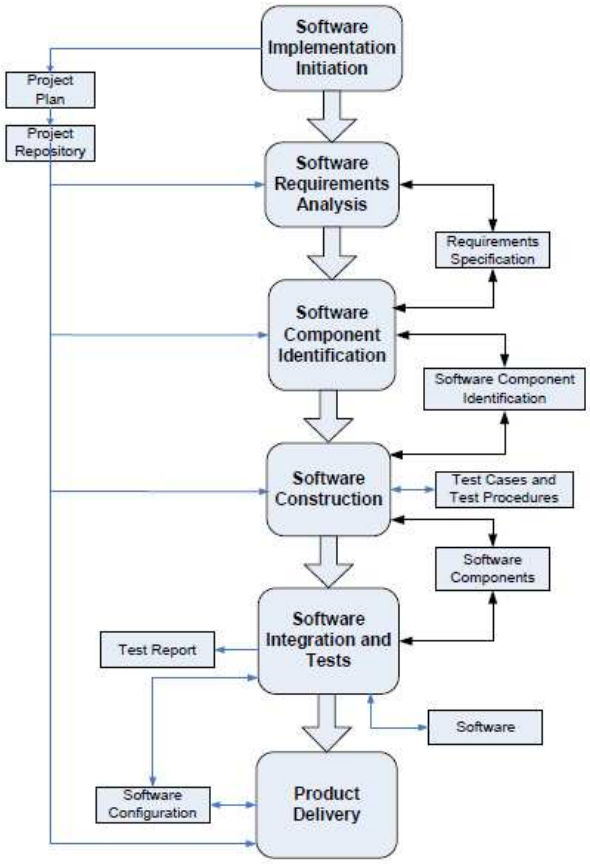
Gambar 1 **Project Management**

Dimulai dari *project planning*, dimana berisi tentang dokumentasi perencanaan secara detail untuk mengatur proyek yang sedang berjalan. Ada 2 hal yang menarik yaitu *statement of work* yang berisi tentang surat kerja dari klien kepada perusahaan dan *project repository* yang berisi data tentang proyek *software*. Setelah itu dilanjutkan dengan *project plan execution* yang berisi tentang aktivitas implementasi *project plan.* Pada proses ini ada *meeting record* dan *progress status record*.

Hal berikutnya adalah *project assessment and control* yang berisi tentang evaluasi dari performa proyek. Jika klien meminta perubahan maka di fase inilah tempatnya, karena di fase ini hal-hal yang berkaitan dengan *project plan* dilakukan. Fase terakhir dari *project management* adalah *project closure*, dimana pada fase ini berisi tentang dokumentasi dari proyek serta produk (dalam hal ini software) yang sesuai dengan *statement of work.*

Setelah melalui proses *project management*, maka ada proses selanjutnya yaitu *software implementation* yaitu proses performa sistematik dari analisis, identifikasi komponen software, konstruksi, integrasi, tes, dan aktivitas pemberian produk yang sesuai dengan kebutuhan yang sudah ditentukan. Gambar 2 menunjukkan proses dari *software implementation* yang terdiri dari 6 fase.

Proses ini dimulai dengan *Software Implementation initiation,* yang berisi aktivitas yang menjamin *project plan* dalam perencanaan proyek dijalankan oleh tim kerja. Proses berikutnya adalah *Software Requirement analysis* yaitu berisi aktivitas yang menganalisa permintaan customer dan melaksanakan kebutuhan software proyek yang sudah divalidasi.



Gambar 2 ***Software Implementation***

Semua data dari *project plan* dan *project repository* dimasukkan dalam fase ini. Fase berikutnya adalah *Software Component Identification* yaitu berisi tentang aktivitas yang merubah kebutuhan software ke komponen arsitektur software. Sesuai dengan namanya, pada fase ini komponen *software* diidentifikasi. Setelah itu fase berikutnya adalah *Software Construction* yaitu berisi tentang aktivitas pembuatan kode software dan data dari *software component identification.*.

Pada fase ini dibuat pula *test cases* dan *test procedures*. Setelah itu dilanjutkan dengan fase *Software Integration and Test* yang berisi tentang aktivitas yang memastikan bahwa komponen software yang terintegrasi sesaui dengan kebutuhan *software*. Pada fase ini dilakukan integrasi pada *software* dan juga *test* yang didapat dari *test cases* dan *test procedures* dari fase sebelumnya yang lalu menghasilkan *test report*.

Fase terakhir dari *Software Implementation* yaitu *Product Delivery* yang berisi tentang produk yaitu software kepada *Project Manager* dan dukungan yang diberikan misalnya saja garansi terhadap *software* yang sudah diterima *customer* dan adanya training dari perusahaan untuk *customer*.

2.5. OWASP Risk Rating Methodology

Open World Application Security Object (OWASP) adalah suatu organisasi nirlaba yang memiliki misi yaitu meningkatan keamanan dari suatu *software* [5]. Berikut adalah metode penilaian risiko yang dibuat OWASP :

- Mengidentifikasi risiko,
- Menentukan faktor-faktor untuk estimasi *likelihood*,

- Menentukan faktor-faktor yang berpengaruh terhadap *impact*,
- Menghitung risk severity,
- Memutuskan risiko mana saja yang harus diprioritaskan berdasarkan Risk Severity nya.

Langkah pertama yaitu mengidentifikasi risiko. Dalam mengidentifikasi risiko, perlu adanya informasi terkait jenis risiko apa saja yang mungkin terjadi, bentuk dan proses penyerangan risiko yang dapat terlaksana..

Langkah kedua adalah menentukan faktor *likelihood*. Secara sederhana perhitungan *likelihood* dapat dilakukan dengan langsung membagi risiko ke dalam beberapa kategori yakni high, medium, low. Ada beberapa faktor yang dapat membantu penentuan *likelihood*, yang pertama adalah *threat agent*.:

- *Skill Level*
- *Motive*
- *Opportunity*
- *Size*

Faktor berikutnya adalah vulnerability faktor, dimana faktor ini dipakai untuk mengestimasi kemungkinan vulnerability ditemukan dan dipergunakan. Vulnerablity faktors juga dibagi ke dalam beberapa kriteria yakni sebagai berikut:

- *Ease of Discovery*
- *Ease of Exploit*
- *Awareness*
- *Intrusion Detection*

Langkah berikutnya adalah menghitung *impact* dari risiko yang ditemukan. Ada 2 jenis *impact* faktors yaitu *technical* dan *business impact* factor. Berikut adalah beberapa faktor dalam *technical impact* factor:

- *Loss of Confidentiality*
- *Loss of Integrity*
- *Loss of Avalaibility*
- *Loss of Accountability*

Berikut adalah beberapa faktor dalam *business impact* factor:

- *Financial Damage*
- *Reputation Damage*
- *Non-Compliance*
- *Privacy Violation*

Tahap berikutnya adalah menentukan *severity* dari setiap risiko yang ditemukan dengan cara mencari rata-rata dari faktor setiap risiko.Setelah itu ditentukan levelnya melalui *likelihood and impact levels*. Setiap risiko mempunyai bobot *likelihood* dan *impact* yang berbeda, mulai dari *low,* lalu *medium*, dan yang paling tinggi adalah *high.* Gambar 3 menunjukkan *likelihood and impact level*.

Likelihood and Impact Levels	
0 to <3	LOW
3 to <6	MEDIUM
6 to 9	HIGH

Gambar 3 *Likelihood and Impact Levels*

3. MODEL PERUSAHAAN

3.1 Cara Kerja Perusahaan

Berikut diuraikan cara kerja perusahaan secara singkat. Proses ini dimulai dengan pertemuan antara *General Manager* dengan klien yang bersangkutan. Klien kemudian memberitahu kebutuhannya (memberi gambaran umum). Setelah itu *general manager* berbicara dengan internal (melakukan *meeting)* perusahaan untuk membahas kebutuhan klien.

Perusahaan lalu membuat proposal yang ditujukan kepada klien. *General manager* dan klien lalu melakukan tawar menawar hingga tercapai *deal* berupa kontrak kerja. Setelah terciptanya kontrak kerja, perusahaan memulai dengan menganalisa kebutuhan klien lebih dalam, misalnya saja teknologi yang bisa dipakai dan masalah yang dihadapi klien.

Perusahaan lalu mulai membuat desain system dan *interface* sederhana.. Jika tidak ada revisi, maka perusahaan mulai membuat *software* itu. Setelah pembuatan selesai, diadakan *trial* selama sebulan dalam *development server* yang dilangsungkan secara *online* karena *software - software* yang dibuat berbasiskan *web.* Jika dalam masa *trial* ditemukan *bug,* maka akan diperbaiki.

Selesai *trial,* maka *software* dimasukkan pada *production server* dan memasuki masa *live* dimana artinya program siap dipakai secara penuh oleh klien. Selama *live* perusahaan akan memberikan training dan *software* memiliki masa garansi yang bervariasi antara 1 bulan hingga 3 bulan tergantung nilai dari kontrak

3.2 Software Yang Dibuat

1.*Software* berbasis web tentang bengkel dan *showroom.* Fitur fiturnya antara lain adalah :

- *View Work Order*
- *Entry Master Product*
- *View Company Profile*
- *View Work Order*
- *View Master Product*

2.*Software* tentang *water traffic* berbasis *web* yang berguna untuk memonitor distribusi air di perumahan. Fitur-fiturnya antara lain:

- *Home Loggers View*
- *Home Loggers Comparison*
- *Home Dashboard*
- *Home Dashboard* Chart
- *Home Consumption Chart*

3.*Software* tentang kontraktor berbasis web yang berguna untuk mendata pekerja, material bangunan dan vendor penjual bahan bangunan. Fitur-fiturnya antara lain :

- *Employees View*
- *Materials View*
- *Vendors View*
- *Employees Add*
- *Materials Add*

4.*Software* berbasis Open ERP yang dipakai untuk laporan produksi perusahaan di kawasan berikat. Fitur-fiturnya antara lain:

- Konsumen *View*
- *Supplier View*
- Pengirim *View*
- Produk *View*
- Bahan Baku *View*

4. IDENTIFIKASI RISIKO

Pada bagian ini akan diberikan contoh identifikasi risiko yang dilakukan pada perusahaan. Pada tabel sudah diberikan pertanyaan, jawaban dari perusahaan, penyebab masalah (jika ada), risiko (jika ada), *control perusahaan* (jika ada) dan *threat* (jika ada). Setiap pertanyaan mengacu pada ISO 29110 sebagai dasar pembuatan pertanyaan. Untuk melihat contoh identifikasi risiko bisa dilihat pada Tabel 2

Tabel 2 Contoh Identifikasi Risiko pada Perusahaan

PM	Pertanyaan	*Software* 1	*Software* 2	*Software* 3	*Software* 4
1.2	Apakah dulu perusahaan pernah mengerjakan *software* yang serupa / mirip dengan *software* ini?	Ya	Tidak	Tidak	Ya
Penyebab masalah adalah *software* yang dibuat merupakan *software* yang berbeda dari *software – software* yang sudah pernah dibuat dan juga karena ada yang memakai bahasa pemrograman yang berbeda. Hal ini mempunyai risiko yaitu waktu pengerjaan yang lebih lama karena tidak ada contoh dari *software – software* yang sudah ada dan *programmer* harus belajar lagi karena *software* yang dibuat memakai bahasa pemrograman yang berbeda. *Control* perusahaan : Belajar bahasa pemrograman dan cara membuat *software* dengan cepat Threat : Klien meminta *software* yang dibuat sesuai dengan fitur yang diinginkan klien dan tidak melebihi batas waktu yang ditentukan.					

5. ANALISA DAN RESPON RISIKO

5.1 Penentuan Faktor dan Pembobotan Aspek Likelihood dan Impact

Kriteria di OWASP tidak dipakai karena kriteria tersebut lebih cocok untuk aspek *security*, sehingga untuk penilaian risiko pengembangan *software* dipakai kritera-kriteria dibawah [1]:

Likelihood dibagi menjadi 3 yaitu

- *Skill Level*
- *Teamwork*
- *Awareness*

Impact dibagi menjadi 5 yaitu

- *Financial Damage*
- *Reputation Damage*
- Kehilangan Integritas
- *Availability*
- *Accountability*

5.2 Risk Severity

Setelah didapatkan *likelihood* dan *impact*, ditentukanlah *severity* dari setiap risiko-risiko yang telah ditemukan sebelumnya, apakah risiko itu termasuk dalam tingkat *severity high, medium, atau low*. Yang dipakai adalah penilaian dari NIST 800-30. *Risk Severity* dihitung berdasarkan tabel *risk matrix* milik NIST 800-30. Tabel *risk level matrix* bisa dilihat pada Tabel 3. Dari 22 faktor risiko terdapat 1 risiko *high,* 2 risiko *medium,* dan 19 risiko *low.* Tabel 4 menampilkan 10 risiko dengan *severity* tertinggi.

Tabel 3 *Risk Level Matrix*

	Impact		
Threat Likelihood	*Low (10)*	*Medium (50)*	*High (100)*
High (1.0)	*Low* 10 x 1.0 = 10	*Medium* 50 x 1.0 = 50	*High* 100 x 0.1 = 100
Medium (0.5)	*Low* 10 x 0.5 = 5	*Medium* 50 x 0.5 = 25	*Medium* 100 x 0.5 = 50
Low (0.1)	*Low* 10 x 0.1 = 1	*Low* 50 x 0.1 = 5	*Low* 100 x 0.1 = 10

Tabel 4 Penentuan *Risk Severity*

No. Risiko	Risiko	*Likelihood Category*	*Impact Category*	*Risk Severity*
9	Tidak ada identifikasi risiko yang dilakukan oleh perusahaan karena tidak ada waktu, sehingga perusahaan tidak tahu risiko-risiko apa saja yang mungkin bisa terjadi pada saat pembuatan *software*.	*High*	*High*	*High*
10	Project plan yang tidak lengkap dari setiap *software* sehingga semakin banyak risiko yang mungkin terjadi.	*Medium*	*Medium*	*Medium*
18	Tidak adanya data yang mencatat system yang dipakai oleh klien sekarang sehingga ada kemungkinan terjadi ketidakcocokan dengan program yang dibuat.	*High*	*Medium*	*Medium*
1	Tidak ada contoh software yang bisa dijadikan contoh dan juga karena *programmer* yang terlibat dalam pembuatan software harus belajar bahasa pemrograman yang baru sehingga waktu pembuatan software menjadi lebih lama.	*Low*	*Low*	*Low*

Sambungan Tabel 4 Penentuan *Risk Severity*

No. Risiko	Risiko	*Likelihood Category*	*Impact Category*	*Risk Severity*
3	Penentuan estimasi waktu pembuatan *software* yang memakai perkiraan, dimana ada kemungkinan proyek menjadi lebih lama dari perkiraan awal.	*Low*	*Low*	*Low*
4	Tidak adanya dokumentasi yang memuat pekerja, peralatan dan *software* yang dipakai sehingga tidak jelas siapa saja yang terlibat, daftar perlatan dan *software* yang dipakai dalam pembuatan *software*.	*High*	*Low*	*Low*
5	Pembuatan program semakin lama karena *programmer* harus belajar bahasa pemrograman yang baru dan juga belajar untuk mnghubungkan *software* dengan alat yang diperuntukkan untuk *software*.	*Low*	*Low*	*Low*
6	Tidak ada estimasi waktu secara detail pada pembuatan *software* sehingga tidak ada control waktu terhadap progress dari software.	*Low*	*Low*	*Low*
7	Tidak ada milestone pada proyek yang sedang dikerjakan sehingga tidak ada patokan terhadap hal-hal apa saja yang sudah harus diselesaikan dalam suatu waktu sehingga proyek sulit dikontrol dari segi waktu	*Low*	*Low*	*Low*
8	Penentuan estimasi biaya dengan perkiraan, sehingga ada kemungkinan biaya *software* yang membengkak dari perkiraan awal.	*Low*	*Low*	*Low*

5.3 Risk Response

Untuk tiap-tiap risiko tersebut diberikan respon. Respon terhadap risiko dapat berupa *accept, avoid, mitigate*, dan *transfer*. *Risk Response* dijabarkan dalam Tabel 5 dengan memakai 10 risiko tertinggi.

Tabel 5 *Risk Response*

No. Risiko	Risiko	*Risk Severity*	*Risk Response*
9	Tidak ada identifikasi risiko yang dilakukan oleh perusahaan karena tidak ada waktu, sehingga perusahaan tidak tahu risiko-risiko apa saja yang mungkin bisa terjadi pada saat pembuatan *software*.	*High*	*Avoid,* karena identifikasi risiko merupakan hal yang penting dalam pembuatan *software*. Cara identifikasi yang bisa dipakai misalnya saja NIST 800-30, OWASP atau memakai ISO 29110 sebagai *guideline* dalam pembuatan *software*
10	Project plan yang tidak lengkap dari setiap *software* sehingga semakin banyak risiko yang mungkin terjadi.	*Medium*	*Avoid,* karena *project plan* berkaitan erat dengan proyek yang sedang dijalankan meskipun klien tidak pernah meminta *project plan* yang lengkap.
18	Tidak adanya data yang mencatat system yang dipakai oleh klien sekarang sehingga ada kemungkinan terjadi ketidakcocokan dengan program yang dibuat.	*Medium*	*Avoid,* karena mencatat system biayanya lebih murah daripada harus menyesuaikan *software* kembali yang tentunya memakan waktu dimana lalu merembet kedalam memakan biaya pengembangan *software* lagi.
1	Tidak ada contoh software yang bisa dijadikan contoh dan juga karena *programmer* yang terlibat dalam pembuatan software harus belajar bahasa pemrograman yang baru sehingga waktu pembuatan software menjadi lebih lama.	*Low*	*Accept,* karena perusahaan hanya mengikuti kemauan dari klien dan *programmer* harus belajar tekun untuk menguasai bahasa pemrograman yang baru agar tidak keluar dari waktu yang ditentukan
3	Penentuan estimasi waktu pembuatan *software* yang memakai perkiraan, dimana ada kemungkinan proyek menjadi lebih lama dari perkiraan awal.	*Low*	*Mitigate,* dengan cara dihitung secermat mungkin kerumitan *software* yang akan dibuat dan dengan memakai *buffer* sehingga jika terjadi molor, maka *buffer* yang akan dipakai.
4	Tidak adanya dokumentasi yang memuat pekerja, peralatan dan *software* yang dipakai sehingga tidak jelas siapa saja yang terlibat, daftar peralatan dan *software* yang dipakai dalam pembuatan *software*.	*Low*	*Avoid,* karena biaya pendokumentasian tidak mahal , sedangkan efek dari sisi biaya jika tidak didokumentasikan cukup mengganggu.

Sambungan Tabel 5 *Risk Response*

No. Risiko	Risiko	*Risk Severity*	*Risk Response*
5	Pembuatan program semakin lama karena *programmer* harus belajar bahasa pemrograman yang baru dan juga belajar untuk menghubungkan *software* dengan alat yang diperuntukkan untuk *software*.	*Low*	*Accept,* karena hal ini berkaitan dengan klien, sehingga mau tidak mau *programmer* harus belajar bahasa pemrograman yang sesuai dengan proyek yang dijalankan.
6	Tidak ada estimasi waktu secara detail pada pembuatan *software* sehingga tidak ada control waktu terhadap progress dari software.		*Mitigate,* jika tidak bisa membuat secara detail maka coba lakukan dengan pemakaian *buffer* (hari ditambahkan dari perkiraan) sehingga jika terjadi molor, maka *buffer* yang akan dipakai.
7	Tidak ada milestone pada proyek yang sedang dikerjakan sehingga tidak ada patokan terhadap hal-hal apa saja yang sudah harus diselesaikan dalam suatu waktu sehingga proyek sulit dikontrol dari segi waktu	*Low*	*Avoid,* karena dengan membuat *milestone* maka proyek lebih mudah dikontrol karena dapat dikethaui dengan jelas sampai mana *progress* dari *software* yang sedang dibuat, apakah molor, lebih cepat atau tepat waktu.
8	Penentuan estimasi biaya dengan perkiraan, sehingga ada kemungkinan biaya *software* yang membengkak dari perkiraan awal.	*Low*	*Mitigate,* dengan cara dibandingkan dengan jumlah hari yang diperkirakan akan dipakai.

6. KESIMPULAN DAN SARAN

Dari proses analisa risiko yang dilakukan dapat disimpulkan beberapa hal:

- Semua faktor risiko yang ada diakibatkan oleh manusia dan prosedur yang dipakai, dalam hal ini bisa dikatakan sebagai vulnerabilities, dimana orang-orang di bagian perusahaan lebih banyak menjadi penyebab dimana salah satunya karena adanya sifat tidak konsisten di dalam prosedur pengembangan *software*, misalnya ada project plan yang lebih lengkap dari suatu *software* dibanding *software* lainnya, tidak ada *milestone*, dan tidak ada proses *monitoring*. Hal lainnya datang dari klien yang mengakibatkan terjadinya risiko pada sisi perusahaan, misalnya pembuatan *software* dengan bahasa pemrograman yang baru sehingga dalam melakukan estimasi waktu dan biaya harus memakai perkiraan.
- Total ada 22 risiko yang ada yang dibagi dalam 3 tingkatan (*rating)*, yang bisa dilihat pada Tabel 6.
- Saran yang diberikan adalah perusahaan menerapkan manajemen risiko secara penuh sehingga kemungkinan muncul risiko bisa dikurangi. Selain hal tersebut, perlu diadakan peninjauan ulang terhadap kebijakan-kebijakan yang terdapat di perusahaan untuk mengurangi atau bahkan menghapus beberapa risiko yang mungkin terjadi.

Tabel 6 Perbandingan Risiko

Risk Rating	Jumlah Risiko
High	1
Medium	2
Low	19

7. REFERENSI

[1] Chrisdiyanto, I. 2013. *IT Risk Assesment Di Perpustakaan Universitas Kristen Petra*. Surabaya : Universitas Kristen Petra

[2] Creswell, J. 2014. *Research Design Fourth Edition*. USA : SAGE Publications, Inc.

[3] Garbarino, S. dan Holland, J. 2009. *Quantitative and Qualitative Methods in Impact Evaluation and Measuring Result*. UK : GSDRC

[4] ISO. 2012. *Software Engineering Lifecycle Profiles for Very Small Entities (VSE).* Switzerland : ISO

[5] OWASP Foundation. 2014. *OWASP Risk Rating Methodology*.URI: https://www.owasp.org/index.php/OWASP_Risk_Rating_Methodology

[6] Rizky, S. 2011. *Konsep Dasar Rekayasa Perangkat Lunak*. Jakarta : Prestasi Pustaka.