

Manajemen Risiko Proyek Perangkat Lunak Menggunakan Pendekatan *Just In Time* Pada Perusahaan Teknologi Informasi

Johan Suryo Prayogo¹, Djoko Budiyo Setyohadi¹

¹Program Studi Magister Teknik Informatika, Universitas Atma Jaya Yogyakarta

Jl. Babarsari 43 Yogyakarta 55281, Telp. (0274) 48758

E-mail: johan_prayogo@yahoo.com, djoko@mail.uajy.ac.id

Masuk: 8 Februari 2017; Diterima: 8 Februari 2017

Abstract. *Software project will always have risks such as technical failure, increasing cost of the project, and overrun schedules. The risks that are not well managed will then transform as a constraint on a software project and will affect the success rate of the software project. PT.Cerise Yogyakarta has not applied software project risk management that it still experienced constraints of technical failure and schedule overruns. The implementation on Kotabagus project in PT.Cerise Yogyakarta used Just In Time approach to determine the value of technical, cost, and schedule. Risk assessment analysis employs The Software Engineering Risk Model (SERIM) to measure the success, and make recommendations for risk reduction. Risk factors affect 50% constraints on the risk elements of technical, 48% constraints on risk elements of cost, and 48% constraints on risk elements of schedule. The software project success value is 0.51, then the risk is still going by 49%.*

Keywords: *software project, risk management, just in time, SERIM*

Abstrak. *Proyek perangkat lunak selalu mempunyai risiko, seperti risiko yang menyebabkan kendala kegagalan teknis, biaya yang bertambah banyak, dan waktu pengerjaan yang semakin lama. Risiko yang tidak dikelola maka akan menjadi kendala pada proyek perangkat lunak dan mempengaruhi keberhasilan dari proyek. PT.Cerise Yogyakarta saat ini belum menerapkan manajemen risiko sehingga kendala ketidaksesuaian teknis dan bertambahnya waktu pengerjaan pada proyek sebelumnya masih terjadi. Penerapan manajemen risiko pada proyek Kotabagus yang dikerjakan PT.Cerise menggunakan pendekatan Just In Time untuk mengetahui nilai elemen risiko technical, cost, dan schedule. Analisis penilaian risiko menggunakan The Software Engineering Risk Model (SERIM) untuk mengukur nilai keberhasilan proyek Kotabagus, dan menyusun rekomendasi untuk pengurangan risiko. Faktor-faktor risiko mempengaruhi 50% kendala pada elemen technical, 48% kendala pada elemen cost, dan 48% kendala pada elemen schedule. Nilai risiko keberhasilan proyek Kotabagus di PT.Cerise adalah 0,51, berarti risiko masih terjadi sebesar 49% pada proyek.*

Kata Kunci: *proyek perangkat lunak, manajemen risiko, just in time, SERIM*

1. Pendahuluan

1.1. Latar Belakang

Proyek pengembangan perangkat lunak selalu mempunyai risiko, dimana risiko tersebut akan mengganggu keberhasilan dari proyek. Risiko yang kemungkinan akan terjadi seperti kegagalan teknis, biaya proyek yang bertambah besar, dan lama pengerjaan proyek yang membutuhkan waktu lebih lama. Proyek pengembangan perangkat lunak itu seperti hutan belantara, dimana risiko yang banyak dan kompleks tersebut menunggu untuk menyabotase proyek anda (Karolak, 1998). Saat ini beberapa perusahaan Teknologi Informasi (TI) belum menjadikan risiko proyek sebagai prioritas untuk dikelola dengan baik.

Manajemen risiko perangkat lunak bertujuan untuk mengenali semua kemungkinan kegagalan dengan melihat dari banyak dan kompleksnya proyek perangkat lunak untuk

memutuskan langkah-langkah solusi yang akan dibuat (Boehm, 1988). Penerapan manajemen risiko membantu untuk mengidentifikasi sumber utama risiko, menganalisis, dan menyelesaikannya (Boehm, 1991). Manajemen risiko proyek perangkat lunak juga membantu manajer proyek sebagai pengambil keputusan untuk mengenali risiko dan membuat strategi untuk mengurangi atau menghindari risiko tersebut. Statistik kegagalan pengembangan proyek perangkat lunak menunjukkan bahwa cukup banyak risiko yang terkait dengan proyek dan pentingnya penerapan manajemen risiko yang kuat (Shukla, 2015).

Salah satu pendekatan manajemen risiko adalah pendekatan *Just In Time* (JIT), yaitu manajemen risiko yang menghasilkan produk yang memiliki biaya yang lebih sedikit dan memiliki kesempatan yang lebih baik dalam memenuhi komitmen jadwal (Karolak, 1996). Pendekatan JIT dapat diterapkan perusahaan pengembang perangkat lunak untuk memperbaiki dan mengurangi risiko atau kegagalan yang mungkin atau sudah terjadi (Suselo, 2007). Pendekatan JIT dikembangkan untuk menganalisis proyek pengembangan perangkat lunak yang bertumpu kepada *technical*, *cost*, dan *schedule*. Pendekatan JIT menggunakan SERIM (*The Software Engineering Risk Model*) yaitu metode yang mengembangkan strategi proaktif, strategi proaktif dimulai sebelum kerja proyek diawali, risiko potensial diidentifikasi, probabilitas dan pengaruh proyek diperkirakan serta diprioritaskan menurut kepentingan, kemudian membangun suatu rencana untuk manajemen risiko. SERIM mengembangkan tindakan yang terfokus dan terencana untuk mengelola risiko sebelum mereka menjadi kenyataan (Stern, 2011).

PT.Cerise Information Technology Yogyakarta adalah perusahaan TI yang berada di kota Yogyakarta, yaitu perusahaan TI yang mengutamakan kebutuhan bisnis klien. Studi kasus pada penelitian ini adalah salah satu proyek perangkat lunak yang dikerjakan oleh PT.Cerise, yaitu proyek perangkat lunak website *e-commerce* Kotabagus yang mempunyai fitur, fungsi, dan fasilitas yang lebih kompleks dari proyek perangkat lunak yang pernah dikerjakan PT.Cerise sebelumnya. PT.Cerise saat ini belum menerapkan manajemen risiko untuk setiap proyek pengembangan perangkat lunak yang dikerjakan. Beberapa risiko yang sering dihadapi dalam pengerjaan proyek perangkat lunak di PT.Cerise adalah risiko proyek seperti teknis, biaya, dan penjadwalan proyek yang terkadang tidak sesuai dengan rencana yang ditetapkan.

Pendekatan JIT dapat mengenali hampir seluruh risiko proyek dan memberikan pengetahuan kepada pengembang perangkat lunak dalam mengelola, mengukur, menilai dan memprediksi risiko (Afrizal & Harjoko, 2009). Berdasarkan masalah yang terjadi pada PT.Cerise maka penelitian ini akan menerapkan manajemen risiko dengan pendekatan JIT pada proyek perangkat lunak Kotabagus untuk mengenali seluruh ruang lingkup perangkat lunak. Manajemen risiko proyek perangkat lunak dengan pendekatan JIT diharapkan dapat membantu manajer proyek perangkat lunak Kotabagus untuk mengukur keberhasilan proyek perangkat lunak dan menyusun strategi rekomendasi untuk pengurangan risiko pada proyek perangkat lunak Kotabagus.

2. Tinjauan Pustaka

2.1. Kajian Pustaka

Suselo (2007) menjelaskan bahwa penerapan manajemen risiko pada proyek perangkat lunak dengan menggunakan pendekatan JIT pada perusahaan pengembang perangkat lunak dapat digunakan untuk memperbaiki dan meminimalkan kendala yang mungkin terjadi atau sudah terjadi. Perangkat lunak JIT dapat mengenali hampir seluruh risiko proyek dan memberikan pengetahuan kepada pengembang perangkat lunak untuk mengelola, mengukur, menilai, dan memprediksi risiko sehingga mencapai keberhasilan dari proyek perangkat lunak (Afrizal & Harjoko, 2009).

Manajemen risiko pendekatan JIT dengan menggunakan SERIM menghasilkan hasil kuantitatif proyek perangkat lunak pada organisasi (Suselo, 2007). Menurut Stern & Arias (2011) SERIM mengembangkan tindakan untuk mengelola risiko secara terfokus dan terencana pada risiko sebelum risiko tersebut benar-benar menjadi kendala dalam proyek perangkat lunak. SERIM memungkinkan untuk melakukan penilaian faktor-faktor risiko pada proyek pengembangan perangkat lunak dari berbagai perspektif, yang berfokus pada perencanaan

tindakan manajemen risiko sebelum risiko terjadi dan dapat memberikan fokus pada area risiko prioritas tertinggi. Kekurangan SERIM adalah kurangnya panduan untuk menggunakan informasi dalam identifikasi risiko.

2.2. Manajemen Risiko Perangkat Lunak

Manajemen risiko perangkat lunak adalah pengelolaan risiko dan meminimalkan kegagalan yang mencakup aspek *technical*, *cost*, dan *schedule* pada pengembang perangkat lunak (Karolak, 1998). Pengembangan perangkat lunak mempunyai tiga area pokok risiko yaitu ketidakjelasan kebutuhan perangkat lunak mengakibatkan ketidaktepatan fungsionalitas yang dikembangkan. Ketidakhahaman estimasi biaya yang digunakan untuk pengembangan perangkat lunak mengakibatkan biaya yang berlebih. Ketidakmampuan mengukur kinerja tim proyek pengembang perangkat lunak dalam menyelesaikan pekerjaan dan besarnya fungsionalitas mengakibatkan semakin lama jadwal pengembangan perangkat lunak.

2.3. Pendekatan *Just in Time* (JIT)

Karolak menyatakan Pendekatan *Just in Time* (JIT) merupakan ide yang populer di industri manufaktur tradisional, kemudian digunakan juga sebagai teknik manajemen risiko perangkat lunak. JIT pada proyek pengembangan perangkat lunak menggunakan filosofi yang bertumpu pada fungsionalitas, biaya, dan jadwal. Langkah awal pendekatan JIT adalah mengidentifikasi satu set kategori risiko tingkat tinggi, kemudian mengaitkan kategori risiko dengan faktor risiko, metrik risiko, dan pertanyaan yang harus ditanyakan kepada manajer proyek. Pertanyaan-pertanyaan tersebut berguna sebagai daftar periksa untuk mengidentifikasi kelas yang berbeda dari risiko (Karolak, 1998). Pendekatan JIT pada proyek pengembangan perangkat lunak adalah pendekatan yang dilakukan pihak manajemen atau manajer proyek yang bersifat *risk-driven*, konsep pendekatan yang dilakukan adalah antisipasi risiko dalam proyek pengembangan perangkat lunak dan menangani risiko sejak dini dalam proyek pengembangan perangkat lunak sehingga mengurangi waktu siklus proses, yang berimbas pada pengurangan biaya, pemenuhan jadwal, serta kesesuaian fungsionalitas.

2.4. Model SERIM (*the Software Engineering Risk Model*)

SERIM (*the Software Engineering Risk Model*) adalah model yang digunakan untuk memberikan manajemen pemecahan alternatif risiko pada suatu proyek perangkat lunak. SERIM digunakan sebagai pendekatan untuk menghitung risiko proyek perangkat lunak, yaitu pendekatan berdasarkan subyek-subyek kemungkinan berdasarkan pengalaman dan analogi kejadian. Model SERIM mengenalkan tiga langkah dalam mengevaluasi risiko (Halloway, 1979), yaitu (1) Langkah untuk menganalisa alternatif, dimana alternatif harus ada ketika memutuskan kegiatan berdasarkan risiko, (2) membuat model yang akan mengevaluasi alternatif, model harus membantu dalam proses pengambilan keputusan dengan menilai alternatif, dan (3) membuat pilihan, jika pilihan tidak dibuat, dengan berlalunya waktu akan menentukan pilihan untuk anda.

Model SERIM menggunakan pendekatan probabilitas subjektif Bayesian untuk penilaian risiko perangkat lunak. Pendekatan ini memberikan probabilitas subjektif berdasarkan pengalaman sebelumnya atau analogi peristiwa masa lalu, yaitu pandangan pribadi mengukur kemungkinan atau kewajaran peristiwa A yang akan terjadi. Ini akan menarik untuk dicatat bahwa jika lebih dari satu orang menilai probabilitas subjektif, kemudian perbedaan hasil dapat diharapkan.

Model SERIM berhubungan metrik risiko untuk fase *life cycle* perangkat lunak dan aktifitas manajemen risiko perangkat lunak. Dengan demikian, risiko perangkat lunak dapat diidentifikasi dengan fase pengembangan perangkat lunak dan berhubungan dengan setiap pertanyaan metrik yang digunakan dalam metode identifikasi risiko. Contoh menggunakan model SERIM menurut (Karolak, 1998) adalah sebagai berikut: (1) $P(A)$ mewakili total risiko atau kemungkinan keberhasilan proyek perangkat lunak. (2) $P(A_1)$, $P(A_2)$, dan $P(A_3)$ mengidentifikasi kemungkinan sasaran teknis, biaya, dan penjadwalan masa depan berhasil

terpenuhi. (3) P(A4) sampai dengan P(A13) mewakili kemungkinan keberhasilan faktor risiko perangkat lunak yang diidentifikasi menurut metodologi tertentu atau identifikasi risiko. (4) P(B) sampai dengan P(G) mewakili kemungkinan keberhasilan proyek perangkat lunak berdasarkan fase *life cycle* pengembangan perangkat lunak. (5) P(H) sampai dengan P(M) mengidentifikasi kemungkinan terpenuhi kegiatan manajemen risiko perangkat lunak yang diidentifikasi sebelumnya.

Beberapa penelitian mengenai manajemen risiko telah diperkenalkan dan dikembangkan oleh beberapa peneliti. Kumpulan penelitian tersebut tidak dapat dibandingkan antara satu dengan lainnya, disebabkan ruang lingkup penelitian manajemen risiko yang digunakan berbeda-beda. Manajemen risiko proyek perangkat lunak harus dapat dianalisis, dinilai dan dievaluasi dari berbagai ruang lingkup proyek. Ruang lingkup manajemen risiko dengan SERIM terdiri dari: elemen risiko, aktivitas risiko, faktor risiko, matrik risiko dan metodologi *life cycle*.

2.4.1. Elemen Risiko

Penerapan manajemen risiko pada proyek perangkat lunak tidak lepas dari pertimbangan teknologi dan bisnis. Perspektif teknologi menjelaskan alat bantu (*tools*), teknik dan lingkungan, dimana perangkat lunak tersebut diterapkan. Perspektif bisnis menjelaskan sumber daya, jadwal dan dampak bisnis (keberhasilan pembangunan perangkat lunak).

Perangkat lunak JIT mampu untuk mengelola risiko perangkat lunak, baik menurut perspektif teknologi maupun bisnis. Tidak semua risiko dalam perspektif diatas masuk ke dalam risiko perangkat lunak. Hanya terdapat tiga elemen dari risiko yang digunakan dalam perangkat lunak JIT yaitu teknologi, biaya, dan penjadwalan.

Elemen teknologi berhubungan dengan kinerja perangkat lunak, yaitu: kehandalan, kualitas, fungsi, pemeliharaan dan kegunaan kembali. Elemen biaya berhubungan dengan biaya perangkat lunak selama pembangunan perangkat lunak yaitu *variable cost*, *fix cost* dan *budget*. Sedangkan elemen penjadwalan berhubungan dengan jadwal proyek selama pembangunan perangkat lunak, yaitu: jadwal realisasi, jadwal pertemuan dengan pelanggan dan anggota pengembang dan jadwal perubahan waktu proyek.

2.4.2. Aktivitas Risiko

Aktivitas risiko merupakan cara melakukan evaluasi terhadap risiko berdasarkan pandangan dari operasional, strategi, teknologi, bisnis, industri dan para praktisi. Terdapat enam aktivitas yang dilakukan dalam mengevaluasi manajemen risiko perangkat lunak yaitu: (1) Identifikasi risiko yaitu melakukan pengumpulan informasi mengenai proyek perangkat lunak dan mengklasifikasikan informasi tersebut untuk menentukan risiko yang paling potensial dari suatu proyek. Informasi dikumpulkan dengan merujuk data pada proyek perangkat lunak yang pernah dikerjakan. (2) Strategi dan perencanaan risiko yaitu mengembangkan alternatif-alternatif risiko yang akan muncul selama pembangunan perangkat lunak. (3) Penilaian risiko adalah memutuskan dampak risiko yang paling potensial melalui suatu penilaian. (4) Pengurangan atau pemhindaran risiko adalah aktivitas yang dilakukan dalam meminimalkan atau menghindari efek risiko. (5) Membuat laporan digunakan untuk mendokumentasikan pengelolaan risiko dari proyek perangkat lunak, termasuk melakukan perbandingan status risiko dengan risiko proyek yang pernah dikerjakan. (6) Prediksi risiko adalah melakukan prediksi tentang perkembangan risiko dari proyek dengan menggunakan iterasi data dan pengetahuan.

2.4.3. Faktor Risiko

Walaupun secara tidak langsung berpengaruh terhadap perangkat lunak, faktor risiko sangat bermanfaat dalam menjelaskan karakteristik proyek yang dikerjakan pada masa lalu. Penelitian dari McCall & Walter (1977) dan Boehm (1991) menjelaskan terdapat 10 faktor risiko perangkat lunak, dimana faktor risiko tersebut berhubungan dengan kualitas dan kehandalan produk perangkat lunak.

Satu faktor risiko dapat berhubungan lebih dari satu elemen risiko. Berdasarkan pengalaman industri perangkat lunak, setiap faktor risiko diberi pembobotan penilaian berupa tinggi, sedang, dan rendah seperti terlihat pada Tabel 1, dimana bobot tersebut menyatakan derajat pengaruh faktor risiko terhadap elemen risiko.

Tabel 1. Derajat Pengaruh Faktor Risiko Terhadap Elemen Risiko

Faktor Risiko	Elemen Risiko Perangkat Lunak		
	Teknologi	Biaya	Penjadwalan
Organization	Rendah	Tinggi	Tinggi
Estimation	Rendah	Tinggi	Tinggi
Monitoring	Sedang	Tinggi	Tinggi
Development Methodology	Sedang	Tinggi	Tinggi
Tools	Sedang	Sedang	Sedang
Risk Culture	Tinggi	Sedang	Sedang
Usability	Tinggi	Rendah	Rendah
Correctness	Tinggi	Rendah	Rendah
Reability	Tinggi	Rendah	Rendah
Personnel	Tinggi	Tinggi	Tinggi

2.4.4. Matriks Risiko

Matriks Risiko digunakan untuk menilai faktor risiko dalam perangkat lunak. konsep ini ditemukan pertama kali oleh McCall & Walter (1977) dan Boehm (1991) yang berfungsi untuk mendapatkan perangkat lunak yang berkualitas dan handal. Matrik risiko perangkat lunak merupakan kumpulan pertanyaan (kuisisioner) dengan jawaban yang diberi bobot nilai sesuai dengan pendapat responden dalam manajemen risiko proyek perangkat lunak.

3. Metode Penelitian

3.1. Studi Kasus

Penelitian ini mencoba untuk menerapkan manajemen risiko dengan menggunakan pendekatan JIT pada proyek perangkat lunak, dimana penelitian dilakukan di PT.Cerise Information Technology Yogyakarta. Pada tahun 2016, PT.Cerise mengerjakan proyek perangkat lunak website *e-commerce* Kotabagus. Kotabagus adalah website *e-commerce* yang mempunyai fitur, fungsi, dan fasilitas yang lebih kompleks dari proyek perangkat lunak yang pernah dikerjakan PT.Cerise sebelumnya.

Untuk menangani risiko sejak dini pada proyek pengembangan perangkat lunak, peneliti merencanakan untuk melakukan manajemen risiko dengan pendekatan JIT pada proyek pembangunan perangkat lunak Kotabagus di PT.Cerise Yogyakarta. Tujuan yang ingin dicapai dalam penelitian ini adalah: (1) Menerapkan manajemen risiko pada proyek Kotabagus di PT.Cerise dengan menggunakan pendekatan JIT untuk mengetahui nilai risiko elemen risiko *technical*, elemen risiko *cost*, dan elemen risiko *schedule*. (2) Melakukan analisis risiko untuk mengukur nilai keberhasilan proyek Kotabagus di PT.Cerise Yogyakarta. (3) Menyusun rekomendasi pengurangan risiko.

3.2. Pengumpulan Data

Penelitian ini menggunakan kuisisioner metrik risiko yang terdiri dari 81 pertanyaan yang melakukan evaluasi ruang lingkup proyek perangkat lunak dengan pendekatan JIT. Responden yang dilibatkan dalam penelitian ini adalah empat responden, yaitu manajer proyek dan anggota tim proyek pengembangan perangkat lunak Kotabagus. Beberapa contoh pertanyaan yang digunakan dalam penelitian ini adalah sebagai berikut: (1) Apakah perusahaan berencana atau sudah menggunakan manajer proyek perangkat lunak yang berpengalaman? (2) Apakah perusahaan telah membuat proyek perangkat lunak yang mirip sebelumnya? (3) Apakah perusahaan sudah merencanakan atau mempunyai dokumentasi struktur organisasi? (4) Apakah perusahaan mempunyai struktur organisasi tetap? (5) Apakah tingkat kepercayaan diri tim proyek tinggi? (6) Apakah ada komunikasi yang baik dengan perusahaan yang berbeda untuk mendukung proyek pengembangan perangkat lunak? (7) Apakah perusahaan melakukan

manajemen konfigurasi perangkat lunak? (8) Apakah perusahaan melakukan manajemen kualitas perangkat lunak?

Jawaban kuisioner secara umum diberikan nilai probabilitas antara 0 sampai dengan 1 seperti yang terlihat pada Tabel 2. Berdasarkan pada Tabel 2, peneliti dapat memberikan nilai untuk setiap jawaban metrik risiko dari responden dengan menggunakan bobot nilai. Bobot nilai yang diberikan untuk jawaban tidak pernah terjadi atau tidak pernah dilakukan adalah nilai 0, nilai yang diberikan untuk jawaban pernah terjadi atau pernah dilakukan tapi tidak sering adalah nilai 0,3, nilai yang diberikan untuk jawaban sering terjadi atau sering dilakukan adalah nilai 0,6, dan nilai yang diberikan untuk jawaban selalu terjadi adalah nilai 1.

Tabel 2. Bobot nilai setiap jawaban kuisioner metrik risiko

Nilai	Keterangan
0	“Tidak Pernah” (Tidak pernah terjadi atau tidak pernah dilakukan)
0,3	“Pernah” (Pernah terjadi atau pernah dilakukan tapi tidak sering)
0,6	“Sering” (Sering terjadi atau sering dilakukan)
1	“Selalu” (Selalu terjadi atau selalu dilakukan)

Tabel 3 adalah jawaban dari responden yang diberikan bobot nilai dalam menjawab 81 pertanyaan metrik risiko. Pada variabel Q_n , dimana Q adalah pertanyaan dan n adalah nomor. Bobot nilai diberikan berdasarkan jawaban responden dan diberikan nilai berdasarkan pada Tabel 2.

Tabel 3. Hasil nilai jawaban pertanyaan metrik risiko

Q1 = 0,80	Q11 = 0,52	Q21 = 0,62	Q31 = 0,45	Q41 = 0,45	Q51 = 0,37	Q61 = 0,52	Q71 = 0,15	Q81 = 0,52
Q2 = 0,45	Q12 = 0,70	Q22 = 0,80	Q32 = 0,45	Q42 = 0,45	Q52 = 0,45	Q62 = 0,62	Q72 = 0,22	
Q3 = 0,45	Q13 = 0,45	Q23 = 0,52	Q33 = 0,37	Q43 = 0,30	Q53 = 0,70	Q63 = 0,62	Q73 = 0,37	
Q4 = 0,52	Q14 = 0,37	Q24 = 0,52	Q34 = 0,45	Q44 = 0,30	Q54 = 0,47	Q64 = 0,90	Q74 = 0,45	
Q5 = 0,60	Q15 = 0,45	Q25 = 0,52	Q35 = 0,70	Q45 = 0,60	Q55 = 0,62	Q65 = 0,80	Q75 = 0,37	
Q6 = 0,52	Q16 = 0,60	Q26 = 0,52	Q36 = 0,45	Q46 = 0,45	Q56 = 0,52	Q66 = 0,22	Q76 = 0,15	
Q7 = 0,45	Q17 = 0,47	Q27 = 0,60	Q37 = 0,52	Q47 = 0,37	Q57 = 0,45	Q67 = 0,62	Q77 = 0,62	
Q8 = 0,52	Q18 = 0,55	Q28 = 0,62	Q38 = 0,70	Q48 = 0,37	Q58 = 0,30	Q68 = 0,70	Q78 = 0,52	
Q9 = 0,52	Q19 = 0,47	Q29 = 0,70	Q39 = 0,45	Q49 = 0,30	Q59 = 0,37	Q69 = 0,30	Q79 = 0,52	
Q10 = 0,35	Q20 = 0,70	Q30 = 0,70	Q40 = 0,45	Q50 = 0,37	Q60 = 0,52	Q70 = 0,37	Q80 = 0,52	

3.3. Analisis Risiko

Analisis risiko proyek perangkat lunak dengan SERIM diperlukan struktur pohon kemungkinan yang berguna sebagai penghubung perbedaan-perbedaan ruang lingkup yang ada pada perangkat lunak JIT. Beberapa persamaan digunakan untuk menyelesaikan pohon kemungkinan, beberapa kemungkinan dikelompokkan berdasarkan aktivitas manajemen risiko, tahapan fase *life cycle*, dan faktor risiko. Faktor risiko selanjutnya dikelompokkan dan dipadukan berdasarkan elemen-elemen risiko untuk menghasilkan nilai kesuksesan proyek perangkat lunak. Pelaksanaan model SERIM menggunakan beberapa parameter dan persamaan yang harus diidentifikasi dan dipertimbangkan.

Beberapa persamaan digunakan untuk menyelesaikan pohon kemungkinan adalah sebagai berikut. Persamaan (1) untuk menghitung nilai keberhasilan proyek perangkat lunak, diasumsikan jika nilai bobot dari ketiga elemen risiko berbeda maka $P(A) = w_1P(A_1) + w_2P(A_2) + w_3P(A_3)$ dimana w_i adalah angka positif dan $w_1 + w_2 + w_3 = 1$. Persamaan (2) adalah probabilitas elemen risiko *technical*, persamaan (3) adalah probabilitas elemen risiko *cost*, dan persamaan (4) adalah probabilitas elemen risiko *schedule*. Nilai w_n adalah nilai bobot faktor risiko yang dipengaruhi terhadap elemen risiko, dimana $w_4 + w_5 + w_6 + w_7 + w_8 + w_9 + w_{10} + w_{11} + w_{12} + w_{13} = 1$. Persamaan (5) menghitung faktor risiko *organizations*, persamaan (6) menghitung faktor risiko *estimation*, persamaan (7) menghitung faktor risiko *monitoring*, persamaan (8) menghitung faktor risiko *development methodology*, persamaan (9) menghitung faktor risiko *tools*, persamaan (10) menghitung faktor risiko *risk culture*, persamaan (11) menghitung faktor risiko *usability*, persamaan (12) menghitung faktor risiko *correctness*, persamaan (13) menghitung faktor risiko *reliability*, dan persamaan (14) menghitung faktor risiko *personnel*.

$$P(A) = \sum_{n=1}^3 w_n P(A_n) \tag{1}$$

$$P(A_1) = \sum_{n=4}^{13} w_n P(A_n) \tag{2}$$

$$P(A_2) = \sum_{n=4}^{13} w_n P(A_n) \tag{3}$$

$$P(A_3) = \sum_{n=4}^{13} w_n P(A_n) \tag{4}$$

$$P(A_4) = \sum_{n=1}^8 P(Q_n) / 8 \tag{5}$$

$$P(A_5) = \sum_{n=9}^{15} P(Q_n) / 7 \tag{6}$$

$$P(A_6) = \sum_{n=16}^{22} P(Q_n) / 7 \tag{7}$$

$$P(A_7) = \sum_{n=23}^{29} P(Q_n) / 7 \tag{8}$$

$$P(A_8) = \sum_{n=30}^{38} P(Q_n) / 9 \tag{9}$$

$$P(A_9) = \sum_{n=39}^{49} P(Q_n) / 11 \tag{10}$$

$$P(A_{10}) = \sum_{n=50}^{55} P(Q_n) / 6 \tag{11}$$

$$P(A_{11}) = \sum_{n=56}^{64} P(Q_n) / 9 \tag{12}$$

$$P(A_{12}) = \sum_{n=65}^{76} P(Q_n) / 12 \tag{13}$$

$$P(A_{13}) = \sum_{n=77}^{81} P(Q_n) / 5 \tag{14}$$

Persamaan (15) menghitung kemungkinan keberhasilan fase *pre-requirement*, persamaan (16) menghitung kemungkinan keberhasilan fase *requirement*, persamaan (17) menghitung kemungkinan keberhasilan fase *design*, persamaan (18) menghitung kemungkinan keberhasilan fase *code*, persamaan (19) menghitung kemungkinan keberhasilan fase *testing*, dan persamaan (20) menghitung keberhasilan fase *maintenance*.

$$P(B) = \sum (Q1, Q2, Q3, Q4, Q5, Q9, Q10, Q11, Q12, Q14, Q15, Q16, Q17, Q18, Q19, Q21, Q22, Q23, Q24, Q28, Q30, Q35, Q38, Q39, Q40, Q41, Q42, Q43, Q44, Q45, Q46, Q47, Q48, Q49, Q60, Q77, Q78, Q79, Q80, Q81) / 40 \tag{15}$$

$$P(C) = \frac{(Q1, Q2, Q3, Q4, Q5, Q7, Q8, Q13, Q14, Q15, Q18, Q19, Q20, Q21, Q22, Q24, Q25, Q26, Q28, Q30, Q35, Q38, Q39, Q40, Q41, Q42, Q43, Q44, Q51, Q52, Q54, Q56, Q60, Q76)/34}{34} \quad (16)$$

$$P(D) = \frac{(Q1, Q3, Q4, Q5, Q6, Q7, Q8, Q13, Q14, Q15, Q18, Q19, Q20, Q21, Q22, Q24, Q25, Q26, Q28, Q30, Q31, Q35, Q38, Q39, Q40, Q41, Q42, Q43, Q44, Q51, Q53, Q55, Q57, Q60, Q65, Q66, Q67, Q69)/38}{38} \quad (17)$$

$$P(E) = \frac{(Q1, Q3, Q4, Q5, Q6, Q7, Q8, Q13, Q14, Q15, Q18, Q19, Q20, Q21, Q22, Q24, Q25, Q26, Q28, Q30, Q35, Q37, Q38, Q39, Q40, Q41, Q42, Q43, Q44, Q45, Q51, Q58, Q60, Q61, Q65, Q66, Q67, Q68, Q69, Q70)/39}{39} \quad (18)$$

$$P(F) = \frac{(Q1, Q3, Q4, Q5, Q6, Q7, Q8, Q13, Q14, Q15, Q18, Q19, Q20, Q21, Q22, Q24, Q25, Q27, Q28, Q29, Q30, Q32, Q34, Q35, Q36, Q38, Q39, Q40, Q41, Q42, Q43, Q44, Q51, Q59, Q60, Q62, Q64, Q71, Q73, Q74, Q75, Q76)/42}{42} \quad (19)$$

$$P(G) = \frac{(Q1, Q3, Q4, Q5, Q6, Q7, Q8, Q13, Q14, Q15, Q18, Q19, Q20, Q21, Q22, Q24, Q25, Q28, Q30, Q35, Q36, Q38, Q39, Q40, Q41, Q42, Q43, Q50, Q51, Q60, Q63, Q72)/33}{33} \quad (20)$$

Persamaan (21) menghitung kemungkinan terpenuhinya kegiatan identifikasi, persamaan (22) adalah kemungkinan terpenuhinya kegiatan strategi dan perencanaan, persamaan (23) adalah kemungkinan terpenuhinya kegiatan penilaian, persamaan (24) adalah kemungkinan terpenuhinya kegiatan pengurangan dan penghindaran, persamaan (25) adalah kemungkinan terpenuhinya kegiatan laporan, dan persamaan (26) adalah kemungkinan terpenuhinya kegiatan prediksi.

$$P(H) = \sum_{n=1}^{81} (Q_n)/81 \quad (21)$$

$$P(I) = \frac{(Q3, Q9, Q10, Q11, Q12, Q14, Q15, Q16, Q23, Q49, Q77)/11}{11} \quad (22)$$

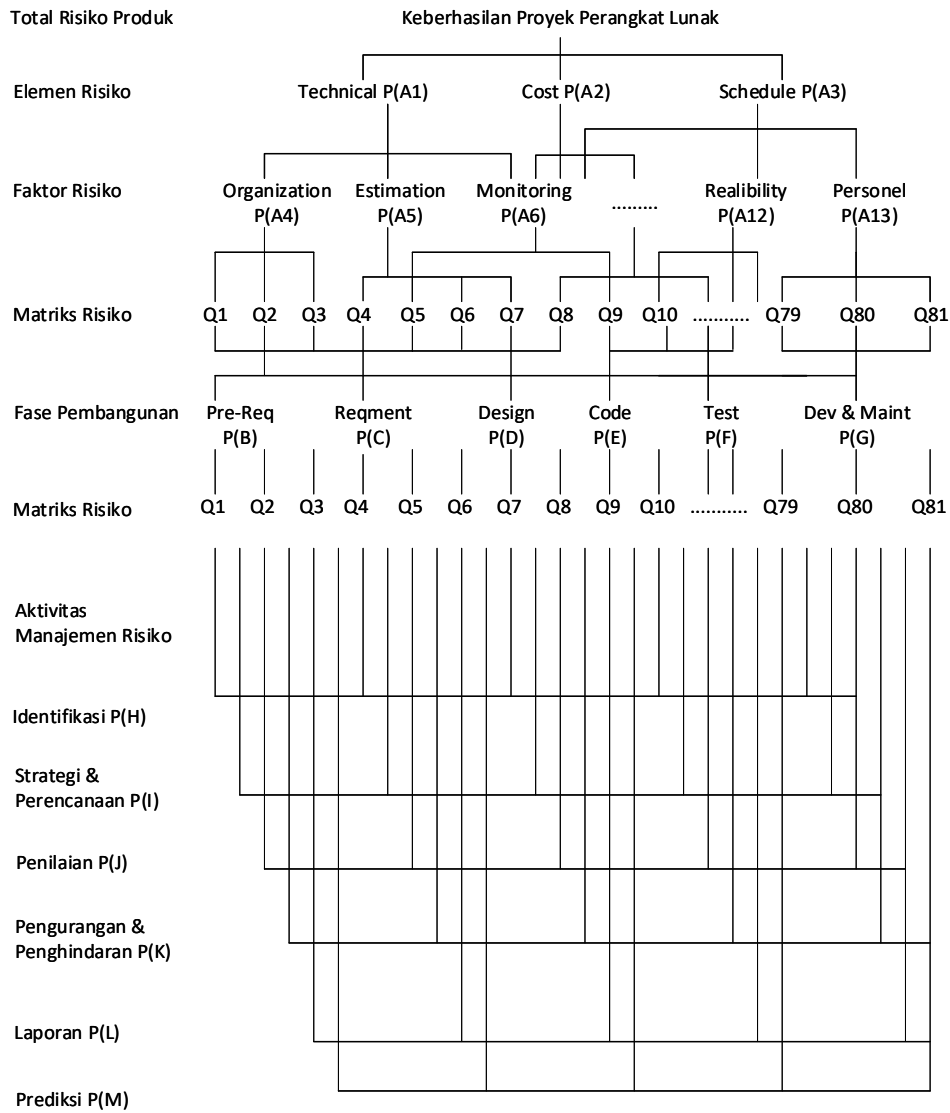
$$P(J) = \frac{(Q1, Q2, Q3, Q7, Q8, Q10, Q11, Q12, Q13, Q14, Q15, Q20, Q21, Q22, Q25, Q29, Q31, Q32, Q33, Q34, Q35, Q36, Q52, Q53, Q55, Q56, Q57, Q58, Q59, Q60, Q61, Q62, Q63, Q64, Q65, Q67, Q68, Q69, Q70, Q71, Q72, Q73, Q74, Q75, Q76, Q77)/46}{46} \quad (23)$$

$$P(K) = \frac{(Q1, Q3, Q4, Q6, Q7, Q8, Q10, Q11, Q12, Q13, Q16, Q17, Q18, Q19, Q20, Q21, Q22, Q23, Q24, Q26, Q27, Q28, Q29, Q30, Q48, Q49, Q52, Q54, Q56, Q57, Q58, Q59, Q61, Q62, Q72, Q73, Q74, Q75, Q76, Q77, Q78, Q79, Q80)/42}{42} \quad (24)$$

$$P(L) = \frac{(Q13, Q17, Q18, Q19, Q20, Q21, Q22)/7}{7} \quad (25)$$

$$P(M) = \sum_{n=1}^{81} (Q_n)/81 \quad (26)$$

Gambar 1 memperlihatkan P(A) yang menunjukkan total kemungkinan kesuksesan proyek perangkat lunak. P(A₁), P(A₂), P(A₃) adalah elemen risiko berupa *technical*, *cost*, dan *schedule*. P(A₄) sampai dengan P(A₁₃) adalah 10 faktor risiko. P(B) sampai dengan P(G) menunjukkan tahapan fase *life cycle*, dan P(H) sampai dengan P(M) adalah aktivitas dalam manajemen risiko.



Gambar 1. Model Manajemen Risiko (Karolak, 1996)

4. Pembahasan

Beberapa persamaan yang digunakan untuk menyelesaikan pohon kemungkinan. Tabel 4 adalah hasil penilaian risiko proyek perangkat lunak dengan beberapa persamaan. Hasil penilaian yang dapat dilihat pada Tabel 4 menjelaskan bahwa nilai keberhasilan proyek perangkat lunak P(A) adalah 0,51, yaitu risiko sebesar 49% dalam proyek perangkat lunak masih terjadi. Nilai elemen risiko *technical* P(A1) adalah 0,50 menjelaskan risiko yang menyebabkan kegagalan teknis sebesar 50%. Nilai elemen risiko *cost* P(A2) adalah 0,52 menjelaskan bahwa risiko yang menyebabkan bertambahnya biaya sebesar 48%. Nilai elemen risiko *schedule* adalah 0,52 dimana menjelaskan bahwa risiko yang menyebabkan bertambahnya waktu pengerjaan proyek sebesar 48%.

Nilai faktor risiko *organizational* P(A4) adalah 0,54, yaitu menjelaskan bahwa faktor risiko *organizational* menyebabkan risiko sebesar 46% pada proyek pengembangan perangkat lunak. Nilai faktor risiko *estimation* P(A5) adalah 0,48, yaitu menjelaskan bahwa faktor risiko *estimation* menyebabkan risiko sebesar 52% pada proyek pengembangan proyek perangkat lunak. Nilai faktor risiko *monitoring* P(A6) adalah 0,60, yaitu menjelaskan bahwa faktor risiko *monitoring* menyebabkan risiko sebesar 40% pada proyek pengembangan perangkat lunak. Nilai faktor risiko *development & methodology* P(A7) adalah 0,57, yaitu menjelaskan bahwa faktor risiko *development & methodology* menyebabkan risiko sebesar 43% pada proyek pengembangan perangkat lunak. Nilai faktor risiko *tools* P(A8) adalah 0,53, yaitu menjelaskan bahwa faktor risiko *tools* menyebabkan risiko sebesar 47% pada proyek pengembangan perangkat lunak. Nilai faktor risiko *risk culture* P(A9) adalah 0,40, yaitu menjelaskan bahwa faktor risiko *risk culture* menyebabkan risiko sebesar 60% pada proyek pengembangan perangkat lunak. Nilai faktor risiko *usability* P(A10) adalah 0,50, yaitu menjelaskan bahwa faktor risiko *usability* menyebabkan risiko sebesar 50% pada proyek pengembangan perangkat lunak. Nilai faktor risiko *correctness* P(A11) adalah 0,53, yaitu menjelaskan bahwa faktor risiko *correctness* menyebabkan risiko sebesar 47% pada proyek pengembangan perangkat lunak. Nilai faktor risiko *reliability* P(A12) adalah 0,39, yaitu menjelaskan bahwa faktor risiko *reliability* menyebabkan risiko sebesar 61% pada proyek pengembangan perangkat lunak. Nilai faktor risiko *personnel* P(A13) adalah 0,54, yaitu menjelaskan bahwa faktor risiko *personnel* menyebabkan risiko sebesar 46% pada proyek pengembangan perangkat lunak.

Tabel 4. Tabel penilaian perangkat lunak dengan SERIM

<i>Probabilitas Risiko</i>	<i>Nilai</i>	<i>Probabilitas Risiko</i>	<i>Nilai</i>
P(A)	0,51	P(B)	0,52
P(A ₁)	0,50	P(C)	0,52
P(A ₂)	0,52	P(D)	0,51
P(A ₃)	0,52	P(E)	0,53
P(A ₄)	0,54	P(F)	0,50
P(A ₅)	0,48	P(G)	0,51
P(A ₆)	0,60	P(H)	0,50
P(A ₇)	0,57	P(I)	0,49
P(A ₈)	0,53	P(J)	0,51
P(A ₉)	0,40	P(K)	0,52
P(A ₁₀)	0,50	P(L)	0,58
P(A ₁₁)	0,53	P(M)	0,50
P(A ₁₂)	0,39		
P(A ₁₃)	0,54		

Nilai fase *pre-requirement* P(B) adalah 0,52, menjelaskan bahwa fase *pre-requirement* menyebabkan risiko sebesar 48% pada proyek pengembangan perangkat lunak. Nilai fase *requirement* P(C) adalah 0,52, menjelaskan bahwa fase *requirement* menyebabkan risiko sebesar 48% pada proyek pengembangan perangkat lunak. Nilai fase *design* P(D) adalah 0,51, menjelaskan bahwa fase *design* menyebabkan risiko sebesar 49% pada proyek pengembangan perangkat lunak. Nilai fase *code* P(E) adalah 0,53, menjelaskan bahwa fase *code* menyebabkan risiko sebesar 47% pada proyek pengembangan perangkat lunak. Nilai fase *testing* P(F) adalah 0,50, menjelaskan bahwa fase *testing* menyebabkan risiko sebesar 50% pada proyek pengembangan perangkat lunak. Nilai fase *maintenance* P(G) adalah 0,51, menjelaskan bahwa fase *maintenance* menyebabkan risiko sebesar 49% pada proyek pengembangan perangkat lunak.

Nilai aktivitas identifikasi P(H) adalah 0,50, menjelaskan bahwa 50% risiko bertumpu pada aktivitas identifikasi. Nilai aktivitas strategi dan perencanaan P(I) adalah 0,49, menjelaskan bahwa 51% risiko bertumpu pada aktivitas strategi dan perencanaan. Nilai aktivitas penilaian P(J) adalah 0,51, menjelaskan bahwa 49% risiko bertumpu pada aktivitas penilaian. Nilai aktivitas pengurangan atau penghindaran P(K) adalah 0,52, menjelaskan bahwa 48% risiko bertumpu pada aktivitas pengurangan atau penghindaran. Nilai aktivitas laporan P(L)

adalah 0,58, menjelaskan bahwa 42% risiko bertumpu pada aktivitas identifikasi. Nilai aktivitas prediksi P(M) adalah 0,50, menjelaskan bahwa 50% risiko bertumpu pada aktivitas prediksi.

Tabel 4 menunjukkan tiga nilai kemungkinan faktor risiko terendah, yaitu *reliability* P(A12) adalah 0,39, *risk culture* P(A9) adalah 0,40, dan *estimation* P(A5) adalah 0,48. Nilai faktor risiko *reliability*, *risk culture*, dan *estimation* memungkinkan manajer proyek untuk menyusun strategi dalam mengurangi atau menghindari risiko. Beberapa strategi yang dapat digunakan untuk mengurangi atau menghindari risiko adalah menerapkan metode standar untuk prosedur estimasi, melakukan dokumentasi hasil estimasi, menerapkan model keandalan perangkat lunak dalam proyek, dan menerapkan manajemen risiko untuk proyek pengembangan perangkat lunak. Nilai faktor risiko tertinggi *monitoring* P(A6) adalah 0,60, menjelaskan bahwa tim proyek pengembangan perangkat lunak Kotabagus sudah melakukan monitoring dengan baik dan mempunyai komunikasi koordinasi anggota tim yang baik.

5. Kesimpulan

Hasil kesimpulan dari pembahasan adalah nilai keberhasilan proyek pengembangan perangkat lunak Kotabagus adalah 0,51, dimana menjelaskan bahwa risiko sebesar 49% masih terjadi pada proyek Kotabagus. Faktor risiko mempengaruhi 50% untuk risiko yang memicu kegagalan teknis produk perangkat lunak (technical), 48% untuk risiko yang memicu bertambah besarnya biaya proyek (*cost*), dan 48% untuk risiko yang memicu lamanya waktu pengerjaan proyek perangkat lunak Kotabagus (*schedule*). Nilai faktor risiko terendah adalah *risk culture* dan *reliability*. Faktor risiko *risk culture* menyebabkan risiko sebesar 60% pada proyek pengembangan perangkat lunak Kotabagus, dan faktor risiko *reliability* menyebabkan risiko sebesar 61% pada proyek pengembangan perangkat lunak Kotabagus. Nilai faktor risiko tertinggi adalah *monitoring*, menjelaskan bahwa tim proyek pengembangan perangkat lunak Kotabagus monitoring yang baik dan mempunyai komunikasi koordinasi anggota tim yang baik.

Nilai faktor risiko seperti *reliability*, *risk culture*, dan *estimation* menjadi perhatian manajer proyek untuk menyusun strategi pengurangan atau penghindaran risiko. Beberapa strategi yang digunakan untuk menghindari atau mengurangi risiko adalah menerapkan model keandalan perangkat lunak untuk memprediksi keandalan perangkat lunak, menerapkan model estimasi dan melakukan dokumentasi untuk estimasi teknis, biaya dan jadwal, serta melakukan dokumentasi dan pelaksanaan prosedur manajemen risiko di PT.Cerise Yogyakarta.

Referensi

- Afrizal, Y., & Harjoko, A. 2009. Perangkat Lunak JIT (Just in Time) untuk Memprediksi Risiko Proyek Perangkat Lunak. *Jurnal Sistem Informasi*, 4 (1): 61-74.
- Boehm, B.W. 1988. A Spiral Model Software Development and Enhancement. *Computer*, 21 (5):61-72.
- Boehm, B.W. 1991. Software Risk Management: Principles and Practices. *IEEE Software*, 8 (1): 32-41.
- Halloway, C.A. 1979. *Decision Making Under Uncertainty: Models and Choices*. Englewood Cliffs: Prentice-Hall.
- Karolak, D.W. 1998. *Software Engineering Risk Management: Finding Your Path Through The Jungle*. Prentice-Hall.
- Karolak, D.W., Karolak, N. 1996. *Software Engineering Risk Management: A Just-In-Time Approach*. Los Alamitos, CA, USA: IEEE Computer Society Press.
- McCall, Richard, J.A.P.K. & Walter, G.F. 1977. *Factors in Software Quality*. Report 77DIS02, Sunnyvale: General Electric Command and Information System Tech.
- Pressman, R.S. 2001. *Software Engineering: A Practitioner's Approach*. Pressman Inc.
- Shukla, S., & Husain, M. 2015. Study of Software Risk Analysis Models on Distributed Systems. *International Journal of Research and Development in Applied Science and Engineering (IJRDASE)*, 7 (1).
- Stern, R., Arias, J. C. 2011. Review of Risk Management Methods. *Business Intelligence Journal*, 4 (1): 59-78.

Suselo, T. 2007. Analisis Manajemen Risiko Perangkat Lunak dengan Pendekatan Just-in-Time: Studi Kasus Optimasi Organisasi dan Dokumentasi pada Organisasi Pengembang Perangkat Lunak. *Jurnal Teknologi Industri*, XI (2): 57-61.