



Risiko dan Mitigasinya dalam Proyek Pengembangan Perangkat Lunak di Indonesia

Setio Ardy Nuswantoro

17917222

Tesis diajukan sebagai syarat untuk meraih gelar Magister Komputer

Konsentrasi Sistem Inforormasi Enterprise

Program Studi Informatika Program Magister

Fakultas Teknologi Industri

Universitas Islam Indonesia

2020

Lembar Pengesahan Pembimbing

Risiko dan Mitigasinya dalam Proyek Pengembangan Perangkat Lunak di Indonesia

Setio Ardy Nuswantoro

17917222



الجامعة الإسلامية
الابستاد الاندونيستة

Pembimbing

A handwritten signature in blue ink that reads 'Fathul Wahid'. The signature is written in a cursive style.

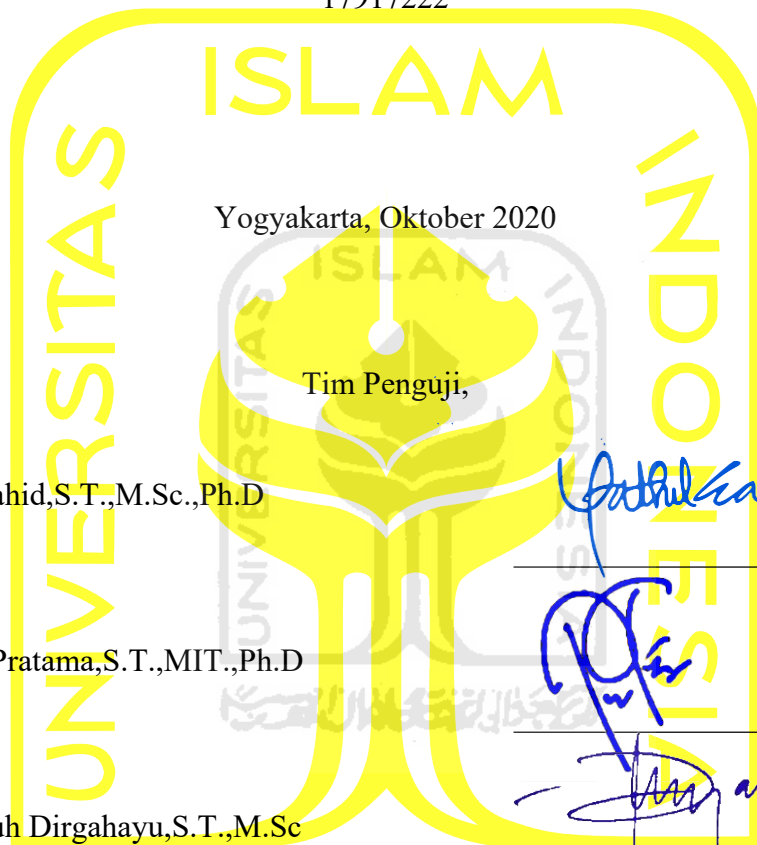
Fathul Wahid, S.T.,M.Sc.,Ph.D

Lembar Pengesahan Penguji

Risiko dan Mitigasinya dalam Proyek Pengembangan Perangkat Lunak di Indonesia

Setio Ardy Nuswantoro

17917222



Yogyakarta, Oktober 2020

Tim Penguji,

Prof. Fathul Wahid, S.T., M.Sc., Ph.D

Ketua

Ahmad Raf'ie Pratama, S.T., MIT., Ph.D

Anggota I

Dr. Raden Teduh Dirgahayu, S.T., M.Sc

Anggota II

الجامعة الإسلامية
الاستاذة الدكتورة
الاستاذة الدكتورة

Mengetahui,

Ketua Program Studi Informatika Program Magister

Universitas Islam Indonesia



Izzati Mulya, S.T., M.Sc., Ph.D.

Abstrak

Risiko dan Mitigasinya dalam Proyek Pengembangan Perangkat Lunak di Indonesia

Saat ini di berbagai bidang telah berhasil menerapkan perangkat lunak, akan tetapi tingkat keberhasilannya masih cukup rendah. Dilaporkan tingkat keberhasilan proyek pengembangan perangkat lunak di Asia hanya 22%, sedangkan di Indonesia tingkat keberhasilan implementasi perangkat lunak hanya mencapai 50%. Tujuan dilakukannya penelitian ini untuk mengidentifikasi risiko kegagalan pengembangan perangkat lunak di Indonesia. Penelitian ini dirancang untuk menjawab tiga pertanyaan: (A) Apa risiko dalam proyek pengembangan perangkat lunak, (B) Apakah terdapat perbedaan tahap pengembangan perangkat lunak untuk risiko yang ditemukan dan (C) Apa strategi untuk mitigasi risiko yang ditemukan. Menggunakan metode Delphi dengan serangkaian kuesioner berulang yang bertujuan untuk memperoleh kesepakatan panelis. Penelitian dengan menggunakan metode Delphi yang melibatkan 41 panelis dengan wilayah serta peran yang berbeda dalam pengembangan perangkat lunak dan diperoleh 17 risiko kegagalan yang mencapai kesepakatan panelis. ditemukan bahwa tiap tahap pengembangan (perencanaan, analisis, desain, pengembangan, implementasi dan pemeliharaan) ternyata mempunyai dominan risiko yang berbeda-beda dengan risiko kurangnya analisis kebutuhan atau proses bisnis menjadi risiko yang paling dominan pada tahapan analisis dan desain. Sebagian besar panelis mengangkat analisis, evaluasi dan komunikasi sebagai strategi mitigasi risiko.

Kata kunci

metode delphi, pengembangan perangkat lunak, risiko kegagalan, manajemen risiko ti, strategi mitigasi risiko

Abstract

The Risks and Mitigations Software Development Projects in Indonesia

Currently, in various fields have succeed in software implementation, but the success rate is still quite low. It was reported that the success rate of software development projects in Asia was only 22%, while in Indonesia the success rate of software implementation was only 50%. The purpose of this research is to identify the risk of software development failure in Indonesia. This study is designed to answer three questions: (A) What are the risks in a software development project, (B) is there any difference stages of software development for the risks found and (C) What are the strategies for mitigating the identified risks. Using the Delphi method with a series of repeated questionnaires aimed at obtaining panelists' agreement. Research using the Delphi method involving 41 panelists with different regions and roles in software development and obtained 17 risk of failure to reach panelists agreement. It was found that each development stage (planning, analysis, design, development, implementation and maintenance) turned out to have different dominant risks with the risk of needs analyst less or business process being the most dominant risk at the analysis and design stage. Most of the panelists raised analysis, evaluation and communication as risk mitigation strategies.

Keywords

delphi method, software development, failure risk, it risk management, risk mitigation strategy

Pernyataan Keaslian Tulisan

Dengan ini saya menyatakan bahwa tesis ini merupakan tulisan asli dari penulis, dan tidak berisi material yang telah diterbitkan sebelumnya atau tulisan dari penulis lain terkecuali referensi atas material tersebut telah disebutkan dalam tesis. Apabila ada kontribusi dari penulis lain dalam tesis ini, maka penulis lain tersebut secara eksplisit telah disebutkan dalam tesis ini.

Dengan ini saya juga menyatakan bahwa segala kontribusi dari pihak lain terhadap tesis ini, termasuk bantuan analisis statistik, desain survei, analisis data, prosedur teknis yang bersifat signifikan, dan segala bentuk aktivitas penelitian yang dipergunakan atau dilaporkan dalam tesis ini telah secara eksplisit disebutkan dalam tesis ini.

Segala bentuk hak cipta yang terdapat dalam material dokumen tesis ini berada dalam kepemilikan pemilik hak cipta masing-masing. Apabila dibutuhkan, penulis juga telah mendapatkan izin dari pemilik hak cipta untuk menggunakan ulang materialnya dalam tesis ini.

Yogyakarta, 16 Oktober 2020



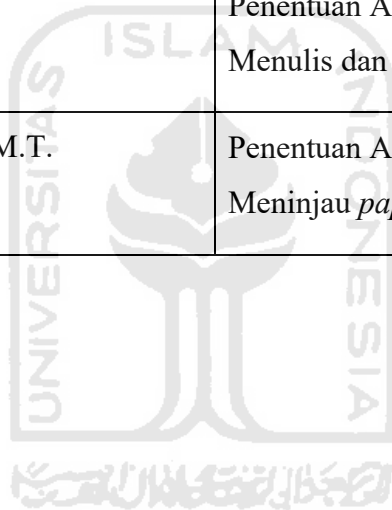
Setio Ardy Nuswantoro S.Kom

Daftar Publikasi

Nuswantoro, S. A., Wahid, F., & Putro, H. P. (2020). *Risiko pengembangan perangkat lunak di indonesia : Temuan studi delphi Software development risk in indonesia: Findings from a delphi study*. 10(1), 43–54.

Sitasi publikasi 1

Kontributor	Jenis Kontribusi
Setio Ardy Nuswantoro	Penentuan Alur Makalah (50%) Menulis <i>paper</i> (70%)
Fathul Wahid, S.T.,M.Sc.,Ph.D	Penentuan Alur Makalah (30%) Menulis dan mengedit <i>paper</i> (30%)
Hanson Prihantoro Putro, S.T., M.T.	Penentuan Alur Makalah (20%) Meninjau <i>paper</i>



Halaman Kontribusi

“Tidak ada kontribusi dari pihak lain”.



Halaman Persembahan

Teruntuk Keluarga, Dosen Pengajar Maagister Informatika, Teman-teman, dan Para Panelis Terimakasih atas segalanya. Semoga Allah SWT membalas dan melipatgandakan semua kebaikan yang telah dilakukan. Amim ya rabbal alamin.



Kata Pengantar

Bismillahirrahmanirrahim...

Assalamualaikum Warahmatullahi Wabarakatuh

Alhamdulillahirobbilalamin puja dan puji syukur kepada Allah SWT yang selalu memberikan pertolongan sehingga penulis dapat menyelesaikan laporan ini dengan judul “Risiko dan mitigasinya dalam proyek pengembangan perangkat lunak di Indonesia”. Ucapan terimakasih juga tidak penulis lupakan teruntuk :

1. Bapak Fathul Wahid, S.T., M.Sc., Ph.D selaku Pembimbing I
2. Bapak Hanson Prihantoro Putro, S.T., M.T. selaku Pembimbing II
3. Ibu Izzati Muhimmah, M.Sc., Ph.D., selaku Ketua Program Studi Teknik Informatika Program Magister, Fakultas Teknologi Industri, Universitas Islam Indonesia.
4. Kedua orang tua yang senantiasa selalu mendukung dan mendoakan.
5. Agus, Coy, Putri dan Virah teman-teman yang selalu menjadi tempat bertukar pikiran
6. Serta semua pihak yang selalu mendukung penulisan tesis ini.

Semoga tulisan ini dapat bermanfaat bagi pembaca dan penulis lain serta semoga Allah SWT selalu senantiasa menolong dan meberikan bimbingannya. Amin ya robbal alamin.

Wassalamualaikum Warahmatullahi Wabarakatuh.

Daftar Isi

Lembar Pengesahan Pembimbing	i
Lembar Pengesahan Penguji.....	ii
Abstrak	iii
Abstract.....	iv
Pernyataan Keaslian Tulisan	v
Daftar Publikasi	vi
Halaman Kontribusi.....	vii
Halaman Persembahan	viii
Kata Pengantar.....	ix
Daftar Isi.....	x
Daftar Tabel.....	xii
Daftar Gambar	xiii
BAB 1 Pendahuluan	1
1.1 Latar Belakang.....	1
1.2 Rumusan Masalah.....	3
1.3 Tujuan Penelitian	3
BAB 2 Tinjauan Pustaka	4
2.1 Tahap Pengembangan Perangkat Lunak.....	4
2.2 Manajemen Risiko	6
2.3 Tabulasi Tinjauan Pustaka	8
2.3.1 Risiko Kegagalan.....	8
2.3.2 Risiko Berdasarkan Tahap Pengembangan.....	9
2.3.3 Referensi Risiko dan Mitigasinya.....	11
BAB 3 Metodologi Penelitian	12
3.1 Metode Pengumpulan Data.....	12

3.2	Metode Delphi.....	12
3.3	Peserta Delphi	13
3.4	Tahap Pengumpulan Data	13
3.5	Pengolahan Data	15
BAB 4 Hasil dan Pembahasan.....		17
4.1	Peran dan Lokasi panelis	17
4.1.1	Tahap Pertama	17
4.1.2	Tahap Kedua.....	18
4.1.3	Tahap Ketiga.....	18
4.1.4	Tahap Keempat.....	19
4.2	Hasil.....	20
4.2.1	Hasil Data Tahap Pertama	20
4.2.2	Hasil Data Tahap Kedua.....	21
4.2.3	Hasil Data Tahap Ketiga.....	25
4.2.4	Hasil Data Tahap Keempat.....	26
4.3	Pembahasan.....	28
4.3.1	Risiko Kegagalan.....	28
4.3.2	Risiko pada Tahap Pengembangan.....	29
4.3.3	Risiko dan Tipe Mitigasinya.....	32
BAB 5.....		37
Kesimpulan dan Implikasi		37
5.1	Kesimpulan	37
5.2	Implikasi Konseptual	38
5.3	Implikasi Praktikal	38
5.4	Saran	38
REFERENSI.....		39

Daftar Tabel

Tabel 2.1. Tabulasi Risiko Kegagalan Perangkat Lunak.....	8
Tabel 2.2. Risiko Tahap Analisis Kebutuhan dan Perencanaan	9
Tabel 2.3. Risiko pada Tahap Desain	10
Tabel 2.4. Risiko pada Tahap <i>development</i>	10
Tabel 2.5. Risiko Teknis pada implementasi dan Pemeliharaan	10
Tabel 2.6. Referensi Risiko dan Mitigasi risiko	11
Tabel 3.1. Studi Empiris Metode Delphi	13
Tabel 4.1. Hasil Risiko Tahap Pertama	21
Tabel 4.2. Penilaian Konsensus <i>Development</i>	22
Tabel 4.3. Penilaian Konsensus <i>Product Owner</i>	22
Tabel 4.4. Penilaian Konsensus <i>Project Manager</i>	23
Tabel 4.5. Penilaian Konsensus Kelompok.....	24
Tabel 4.6. Risiko berdasarkan tahap pengembangan.....	25
Tabel 4.7. Hasil Data Tahap Keempat.....	26
Tabel 4.8. Risiko Kegagalan yang Mecipai Nilai Konsensus Kelompok.....	29
Tabel 4.9. Risiko dan Mitigasinya.....	32
Tabel 4.10. Keterangan Tipe Mitigas	35

Daftar Gambar

Gambar 2.1. Tahap Pengembangan.....	4
Gambar 3.1. Tahapan Pengumpulan Data	14
Gambar 4.1. Lokasi Panelis Tahap Pertama.....	17
Gambar 4.2. Peran Panelis Tahap Pertama.....	17
Gambar 4.3. Lokasi Panelis Tahap Kedua	18
Gambar 4.4. Peran Panelis Tahap Kedua	18
Gambar 4.5. Lokasi Panelis Tahap Ketiga	19
Gambar 4.6. Peran Panelis Tahap Ketiga	19
Gambar 4.7. Lokasi Panelis Tahap Keempat.....	20
Gambar 4.8. Peran Panelis Tahap Keempat	20
Gambar 4.9 Visualisasi Risiko pada Tahap Pengembangan	30



BAB 1

Pendahuluan

1.1 Latar Belakang

Pengembangan perangkat lunak memiliki proses yang kompleks dengan melibatkan departemen, perusahaan dan individu dengan pengetahuan yang luas serta spesialisasi individu yang beragam (Akgün, 2020; de Vasconcelos et al., 2017; Ghobadi, 2015; Gupta et al., 2019). Saat ini pengembangan perangkat lunak menjadi semakin kompleks karena keberagaman permintaan pelanggan dan bahkan sering berubah-ubah (Kussunga & Ribeiro, 2019). Dalam berbagai bidang saat ini telah berhasil menerapkan perangkat lunak, akan tetapi tingkat keberhasilannya masih cukup rendah (Savolainen, Ahonen, & Richardson, 2012). Dilaporkan tingkat keberhasilan proyek pengembangan perangkat lunak di Asia hanya 22%, 58% bermasalah dan 20% berakhir dengan kegagalan (The Standish Group International, 2015). Di Indonesia tingkat kegagalan implementasi perangkat lunak cukup tinggi yaitu 50%, hal ini dikarenakan dalam prosesnya ketergantungan akan peran *Project Management Office* (PMO) masih sangat tinggi (Raharjo et al., 2018). Risiko dalam konteks pengembangan perangkat lunak merujuk pada properti atau aspek tertentu berdasarkan proses lingkungan atau tugas dari proyek pengembangan perangkat lunak yang apabila diabaikan maka kemungkinan kegagalan proyek akan meningkat (Verner et al., 2014). Dalam proyek pengembangan perangkat lunak ditemukan lebih dari satu risiko kegagalan seperti persyaratan yang tidak jelas, estimasi biaya tidak tepat, estimasi waktu tidak sesuai, penggunaan teknologi baru, manajemen proyek yang buruk dan lain-lain (Antony & Gupta, 2019; Gupta et al., 2019; Montequin et al., 2016; Song & Jiang, 2016; Taherdoost & Keshavarzsaleh, 2016; Wong et al., 2017).

Pada proyek pengembangan perangkat lunak terdapat sebuah integrasi yang kompleks dimana satu kegagalan dapat memicu kegagalan dalam skala besar yang berakibat pada kehancuran proyek (Sriwindono & Yahya, 2012). Salah satu kegagalan proyek diakibatkan karena kurangnya kesiapan perusahaan dalam menerapkan dan mengintegrasikan proses manajemen risiko (Tavares et al., 2019). Pemahaman manajemen risiko diperlukan agar dapat menambah nilai suatu proyek (Willumsen et al., 2019). Apabila risiko proyek tidak dapat ditangani dengan tepat maka akan berdampak pada

kegagalan proyek untuk itu diperlukan strategi manajemen risiko guna mengatasi risiko-risiko yang ditemukan (Wan Husin et al., 2019). Selama proyek berlangsung, manajemen proyek dianggap sebagai aktivitas untuk membentengi terjadinya peristiwa yang tidak diinginkan (De Bakker et al., 2011). Selain itu, untuk mengurangi risiko kegagalan proyek perlu rekayasa yang terintegrasi dengan metode, indentifikasi risiko, analisis risiko, pemantauan dan pengendalian risiko (Chaouch et al., 2019).

Pada penelitian terdahulu (Gupta et al., 2019), menemukan risiko kegagalan penting yang terjadi pada proyek pengembangan perangkat lunak namun Antony dan Gupta tidak menyajikan strategi mitigasi risiko untuk tiap-tiap risiko yang ditemukan dan tidak adanya pembahasan secara detail letak risiko berada pada tahapan pengembangan mana, serta penelitian tersebut juga tidak menyajikan risiko yang paling mendominasi pada tiap tahapannya atau pada tahap pengembangan apa risiko-risiko yang ditemukan. Terdapat penelitian lain (Riaz et al., 2019) yang telah berhasil mengklasifikasikan risiko yang digunakan untuk mengidentifikasi risiko berdasarkan akar permasalahannya, namun dalam melakukan proses klasifikasi risiko, peneliti tersebut hanya beracuan pada hasil kuesioner, bukan berdasarkan tahapan pengembangan. Penelitian lain (Rahman et al., 2019) memperoleh temuan bahwa perubahan persyaratan merupakan risiko terbesar dalam proyek pengembangan perangkat lunak dan penelitian tersebut hanya berfokus pada analisis yang dampak yang dapat ditimbulkan dari risiko perubahan persyaratan saja. Penelitian yang terkait dengan risiko kegagalan proyek pengembangan perangkat lunak sebelumnya telah dilakukan di berbagai negara (Antony & Gupta, 2019; Gupta et al., 2019; Montequin et al., 2016; Rahman et al., 2019; Riaz et al., 2019). Akan tetapi, penerimaan dan pengembangan teknologi informasi di Indonesia berbeda dengan negara lain yang dipengaruhi oleh perbedaan sektoral, geografis dan budaya (Montequin et al., 2016; Sriwindono & Yahya, 2012).

Di Indonesia penelitian seperti ini belum pernah dilakukan, oleh karena itu penulis bertujuan untuk melakukan penelitian terkait risiko kegagalan proyek pengembangan perangkat lunak serta apakah terdapat perbedaan tahap pengembangan perangkat lunak untuk risiko yang ditemukan dan strategi mitigasi risiko yang tepat untuk setiap risiko yang ditemukan. Dengan menggunakan metode Delphi yang pengembangannya dilakukan oleh Rand Corporation di tahun 1950-an (Cid et al., 2015; Maher et al., 2015). Metode ini berbeda dengan metode lain karena metode ini dapat lebih diandalkan untuk menyelesaikan masalah dengan ketidakpastian yang tinggi (Ameyaw et al., 2016). Metode ini juga dapat menampung pengetahuan atau pendapat dari para panelis (Lyu et al., 2020).

Tujuan dilakukan penelitian ini untuk mengetahui risiko yang dapat berakibat pada kegagalan proyek pengembangan perangkat lunak dan apakah terdapat perbedaan tahap pengembangan perangkat lunak untuk risiko yang ditemukan serta mitigasi risiko untuk setiap risiko kegagalan yang ditemukan.

Dengan begitu harapan dilakukannya penelitian ini dapat memberikan hasil untuk meningkatkan pengetahuan organisasi atau perusahaan pengembang perangkat lunak untuk menangani atau menghadapi risiko yang mungkin akan timbul pada saat proyek pengembangan perangkat lunak berlangsung dan *stakeholder* maupun pengembang penting untuk mengetahui risiko yang mungkin terjadi dalam proyek pengembangan perangkat lunak di Indonesia guna meminimalkan kegagalan proyek.

12 Rumusan Masalah

Dalam proyek pengembangan perangkat lunak terdapat banyak hambatan bahkan tidak jarang terjadinya kegagalan dalam proses pengembangannya. Untuk itu, penelitian ini dirancang untuk menjawab tiga pertanyaan terkait permasalahan yang akan dibahas pada penelitian ini yaitu:

1. Apa risiko dalam proyek pengembangan perangkat lunak?
2. Apakah terdapat perbedaan tahap pengembangan perangkat lunak untuk risiko yang ditemukan?
3. Apa strategi untuk mitigasi risiko yang ditemukan?

13 Tujuan Penelitian

Tujuan dilakukannya penelitian ini adalah untuk mengetahui risiko yang seringkali menjadi pemicu terjadinya kegagalan dalam proyek pengembangan perangkat lunak serta mengetahui apakah terdapat perbedaan tahap pengembangan perangkat lunak untuk risiko yang ditemukan dan juga menyajikan strategi mitigasi risiko untuk setiap risiko yang ditemukan.

BAB 2

Tinjauan Pustaka

2.1 Tahap Pengembangan Perangkat Lunak

Pada penelitian ini penulis menggunakan 6 tahap pengembangan untuk mengetahui perbedaan tahap pengembangan bagi setiap risiko yang ditemukan. Penulis mengumpulkan 6 tahap pengembangan yang diolah dari berbagai sumber dan dapat dilihat seperti pada Gambar 2.1. Sebagai contohnya pada metode pengembangan *Scrum* dengan tahapan perencanaan, analisis serta pengembangan (Chaouch et al., 2019; Kussunga & Ribeiro, 2019; Nidagundi & Novickis, 2016). Pada metode pengembangan *Agile* terdapat enam tahap pengembangan yaitu perencanaan, analisis, desain, implementasi, pengembangan serta pemeliharaan (Cooper & Sommer, 2016; Khalil, Reza, Junaedi, & Kanigoro, 2015; Loiro, et al., 2019) (Cooper & Sommer, 2016; Khalil et al., 2015; Loiro et al., 2019). Untuk metode pengembangan *Waterfall* tahapan pengembangan yang umum digunakan adalah analisis, desain, implementasi, dan pemeliharaan (Andrei et al., 2019; Bhavsar et al., 2020; Sommerville, 2003), sedangkan pada metode pengembangan *Software Development Life Cycle* (SDLC) tahapan pengembangan yang digunakan adalah, analisis, implementasi serta desain (L. Ahmed et al., 2020; Cai et al., 2020; Lee et al., 2020), dan pada metode pengembangan *Rapid Application Developmen* (RAD) tahapan yang umum dilakukan yaitu perencanaan, analisis, desain dan pengembangan (Andreswari et al., 2020; Chrismanto et al., 2019).



Gambar 2.1. Tahap Pengembangan.
(diolah dari berbagai sumber)

1. Perencanaan

Dalam pengembangan perangkat lunak perencanaan proyek merupakan hal yang sangat penting untuk menunjang keberhasilan (Simarmata, 2010). Pada umumnya tahap perencanaan melibatkan pembuatan perencanaan proyek, validasi, anggaran, mitigasi risiko, waktu pengembangan, kualitas, manajemen konfigurasi, pemeliharaan pengadaan, dan lain-lain (Sommerville, 2003; Westland, 2007).

2. Analisis

Tahap ini merupakan proses untuk memperoleh informasi, spesifikasi, dan model terkait perangkat lunak yang diinginkan oleh pengguna atau klien dimana informasi yang didapatkan dari pengguna atau klien akan dijadikan acuan guna melaksanakan desain perangkat lunak (Simarmata, 2010).

3. Desain

Tahap ini merupakan proses untuk pemecahan masalah serta merencanakan pengembangan perangkat lunak dengan melibatkan perancang dan pengembang perangkat lunak. Pada tahap ini peran dari perancang dan pengembang perangkat lunak adalah merencanakan desain arsitektur perangkat lunak, desain diagram logis, desain antarmuka, desain konsep, desain struktur data dan desain algoritma (Bassil, 2012).

4. Pengembangan

Tahap ini bertujuan untuk mengevaluasi persyaratan yang ditetapkan dalam tahap *planning* yang meliputi pengembangan sistem dan peninjauan desain arsitektur sistem guna memastikan bahwa integrasi yang direncanakan sesuai dengan persyaratan yang telah diidentifikasi sebelumnya (Broad, 2013).

5. Implementasi

Tahap ini merupakan proses penerapan dan penyebaran yang mengacu pada realisasi persyaratan dan spesifikasi desain menjadi program yang dapat dieksekusi melalui pemrograman. Proses ini mengubah seluruh persyaratan agar perangkat lunak dapat dioperasikan (Bassil, 2012).

6. Pemeliharaan

Tahap ini merupakan proses modifikasi solusi setelah perangkat lunak berhasil di implementasikan guna memperbaiki kesalahan, memperbaiki *output* serta meningkatkan kinerja dan kualitas. Tahapan ini juga berkaitan dengan kegiatan pemeliharaan tambahan, memfasilitasi kebutuhan pengguna baru, penyesuaian

perangkat lunak terhadap lingkungan serta meningkatkan fungsionalitas perangkat lunak (Bassil, 2012).

2.2 Manajemen Risiko

Manajemen risiko merupakan upaya untuk mengatasi risiko guna mengurangi terjadinya kerugian (Hanafi, 2014). Oleh karena itu, risiko yang muncul harus dikelola secara optimal sehingga memiliki rencana mitigasi risiko (A. Ahmed et al., 2007). Manajemen risiko mempunyai tujuan untuk mengidentifikasi, menganalisis serta merespon semua risiko proyek yang terjadi (Simarmata, 2010). Terdapat 3 proses dalam manajemen risiko seperti:

1. Identifikasi Risiko

Identifikasi risiko bertujuan untuk mengenali risiko apa saja yang sedang dihadapi oleh suatu organisasi. Salah satu teknik untuk mengidentifikasi risiko yaitu dengan mencari tahu sumber utama risiko tersebut (Hanafi, 2014).

2. Pengukuran dan Evaluasi Risiko

Pengukuran dan evaluasi risiko bertujuan untuk mendalami karakteristik dari risiko yang ada dengan baik agar mudah dipahami dan mudah dalam penanganannya (Hanafi, 2014).

3. Mitigasi Risiko

Mitigasi risiko bertujuan untuk mengelola atau mencari jalan keluar terhadap risiko yang terjadi. Apabila risiko yang ada tidak dapat ditangani dengan baik, maka dapat menyebabkan kerugian yang cukup besar (Hanafi, 2014). Ada beberapa cara untuk mitigasi risiko yaitu (Hopkin, 2012):

A. *Tolerate*

Apabila risiko yang terjadi dirasa cukup besar namun tidak dapat dihindari sehingga akan berdampak pada kerusakan secara alamiah, maka tindakan “*tolerate*” atau menerima risiko tersebut perlu diambil oleh perusahaan.

B. *Treat*

Apabila risiko yang terjadi dirasakan dapat ditangani oleh perusahaan, maka tindakan “*treat*” dapat diambil guna mengontrol risiko tersebut sehingga tingkat risiko dapat diturunkan.

C. *Transfer*

Terdapat beberapa risiko dimana respon terbaik adalah “*transfer*” risiko kepada pihak asuransi atau mungkin dapat dilakukan dengan membayar pihak ketiga agar bersedia mengambil risikonya.

D. *Terminate*

Apabila risiko dirasa akan berdampak kerugian yang besar maka tindakan yang dapat diambil adalah “*terminate*” dengan menghentikan aktivitas atau proses yang berkaitan dengan risiko.

Berikut merupakan contoh *tolerate*, *treat*, *transfer* dan *terminate* (Dewantara, 2016; Raka & Liangrokopart, 2015; Sangwan & Liangro, 2015).

- a. Contoh mitigasi dengan tindakan “*tolerate*” jika risiko yang didapat terlalu besar dan tidak dapat untuk dihindari, sehingga perusahaan mengalami kerusakan secara alamiah maka perusahaan dapat mengambil tindakan “*tolerate*”. Hal tersebut dikarenakan perusahaan tidak dapat melawan risiko yang diakibatkan oleh bencana alam seperti gempa bumi, banjir, badai, dll. Namun, terdapat juga peneliti lain yang menyebutkan bahwa risiko yang diakibatkan oleh bencana alam dapat diatasi dengan “*transfer*” karena perusahaan sebelumnya dapat mengasuransikan produk atau alat. Peneliti lain juga menyebutkan kalau tindakan “*terminate*” juga dapat dilakukan apabila terdapat perkiraan cuaca untuk mencegah terjadi kerusakan yang berlebih dan/atau dapat menunda jika ingin mengeluarkan produk IT baru.
- b. Contoh mitigasi dengan tindakan “*treat*” seperti melakukan kontrol atau mitigasi terhadap risiko yang ada sehingga perusahaan dapat mencegah timbulkan risiko lain yang lebih besar dan dapat menurunkan tingkat level risiko yang sedang terjadi. Contoh lain yang ditemukan oleh peneliti lain yaitu dalam kategori risiko permintaan *customer*. Perusahaan dapat mengambil tindakan “*treat*” dengan menerapkan *customer relationship management* atau dengan menggunakan model *forecasting*.
- c. Contoh mitigasi dengan tindakan “*transfer*” jika perusahaan kurang handal dalam hal mengelola finansial perusahaan, maka langkah yang dapat diambil adalah dengan tindakan “*transfer*” yaitu dengan menyerahkan ke pihak lain sesuai dengan bidang ahlinya, contohnya ke pihak finansial seperti akutansi. Contoh lain, misalkan perusahaan kewalahan dalam menangani pembuatan produk baru, maka perusahaan dapat melakukan tindakan “*transfer*” yang berarti perusahaan dapat berbagi tanggung jawab dengan pihak lain misalkan dengan *sub-*

contractors yang bertujuan untuk memperkecil tingkat risiko yang didapat.

- d. Contoh mitigasi dengan tindakan “*terminate*” jika perusahaan mendapatkan tawaran pembuatan produk IT baru, namun dirasa akan menimbulkan banyak risiko yang buruk bagi perusahaan, maka perusahaan dapat mengambil tindakan “*terminate*” atau menghindari dan tidak mau mengambil tawaran tersebut dengan cara membatalkan rencana pembuatan produk IT di perusahaan tersebut.

23 Tabulasi Tinjauan Pustaka

2.3.1 Risiko Kegagalan

Risiko kegagalan proyek pengembangan perangkat lunak sering terjadi di berbagai negara dan masing-masing peneliti telah menemukan 7 sampai 31 risiko kegagalan pengembangan perangkat lunak. Namun, pada penelitian ini penulis hanya merangkum risiko kegagalan pengembangan perangkat lunak yang mempunyai kesamaan risiko kegagalan dengan penelitian lainnya. Dengan adanya kesamaan risiko antar peneliti satu dengan yang lainnya, maka penulis menyimpulkan bahwa risiko-risiko tersebut merupakan risiko yang umum terjadi dan dianggap cukup penting.

Tabel 2.1. Tabulasi Risiko Kegagalan Perangkat Lunak.

NO	Faktor	Referensi
1.	Kurangnya komitmen top manajemen	(Antony & Gupta, 2019; Chien et al., 2014; Gupta et al., 2019; Montequin et al., 2016)
2.	Kurangnya keterlibatan pengguna	(Dwivedi et al., 2014; Kaur & Sengupta, 2013)
3.	Persyaratan tidak jelas	(Hu et al., 2013; Kaur & Sengupta, 2013; Montequin et al., 2016; Song & Jiang, 2016; Taherdoost & Keshavarzsaleh, 2016)
4.	Ruang lingkup proyek tidak jelas	(Antony & Gupta, 2019; Whitney & Daniels, 2013)
5.	Kurangnya kerja sama tim	(Antony & Gupta, 2019; Lehtinen et al., 2014; Montequin et al., 2016)
6.	Penggunaan teknologi baru	(Hu et al., 2013; Montequin et al., 2016; Song & Jiang, 2016; Taherdoost & Keshavarzsaleh, 2016; Whitney & Daniels, 2013)
7.	Kurangnya keterampilan tim pengembang	(Antony & Gupta, 2019; Hu et al., 2013; Montequin et al., 2016; Song & Jiang, 2016; Wong et al., 2017)
8.	Ukuran tim tidak sesuai	(Antony & Gupta, 2019; Gupta et al., 2019; Hu et al., 2013; Kaur & Sengupta, 2013; Song & Jiang, 2016)
9.	Estimasi biaya tidak tepat	(Chien et al., 2014; Montequin et al., 2016; Wong et al., 2017)
10.	Estimasi waktu tidak sesuai	(Kaur & Sengupta, 2013; Montequin et al., 2016; Song & Jiang, 2016)
11.	Manajemen proyek yang buruk	(Gupta et al., 2019; Kaur & Sengupta, 2013; Montequin et al., 2016; Taherdoost & Keshavarzsaleh, 2016; Verner et

		al., 2014; Whitney & Daniels, 2013)
12.	Gagal mendapatkan komitmen pengguna	(Hu et al., 2013; Kaur & Sengupta, 2013)
13.	Gagal memenuhi permintaan pengguna	(Montequin et al., 2016; Wong et al., 2017)
14.	Komunikasi yang buruk	(Antony & Gupta, 2019; Fang & Marle, 2012; Gupta et al., 2019; Montequin et al., 2016; Verner et al., 2014; Wong et al., 2017)
15.	Perbedaan Budaya	(Gupta et al., 2019; Montequin et al., 2016; Verner et al., 2014; Wan Husin et al., 2019)

Pada penelitian ini, penulis akan melakukan penelitian untuk mengetahui risiko kegagalan pengembangan perangkat lunak yang ada di Indonesia dengan mengetahui risiko apa yang paling penting dan seringkali ditemui oleh banyak pengembang perangkat lunak di Indonesia. Kemudian mencari tahu risiko yang mendominasi di setiap tahap pengembangan serta menyajikan strategi mitigasi risiko untuk setiap risiko yang ditemukan.

2.3.2 Risiko Berdasarkan Tahap Pengembangan

Proyek pengembang perangkat lunak merupakan proses sistematis dimana risiko kegagalan dapat terjadi pada setiap tahap pengembangan perangkat lunak (Roy & Dasgupta, 2016). Peneliti sebelumnya telah melakukan identifikasi dan mengkategorikan risiko berdasarkan setiap tahap pengembangannya yang dapat dilihat pada Tabel 2.2 sampai dengan Tabel 2.5.

Tabel 2.2. Risiko Tahap Analisis Kebutuhan dan Perencanaan

No.	Risiko Kegagalan
1.	Persyaratan yang berubah-ubah
2.	Jadwal waktu tidak realistis
3.	Ukuran hasil kerja tidak akurat
4.	Persyaratan tidak jelas / kurang tepat
5.	Analisis risiko perangkat lunak tidak memadai
6.	Estimasi biaya tidak tepat
7.	Estimasi kualitas tidak tepat
8.	Kebijakan perangkat lunak tidak memadai
9.	Kurang pengetahuan tentang ketersediaan sumber daya
10.	Estimasi sumber daya salah
11.	Tidak mampu menggunakan kembali modul yang sama
12.	Tidak tersedianya dokumentasi
13.	Kurangnya persyaratan
14.	Matrix tidak akurat
15.	Penilaian tidak memadai

Tabel 2.3. Risiko pada Tahap Desain

No.	Risiko Kegagalan
1.	Perancang modul perangkat lunak yang tidak berpengalaman
2.	Alat dan Metode yang tidak memadai untuk Kualitas asuransi
3.	Alat dan Metode yang tidak memadai untuk perangkat lunak
4.	Kurangnya data yang dapat digunakan kembali
5.	Kurangnya desain yang dapat digunakan kembali
6.	Kurangnya arsitektur yang dapat digunakan kembali
7.	Desain antarmuka salah
8.	Modul rawan kesalahan dirancang
9.	Perancang modul perangkat lunak yang tidak berpengalaman
10.	Simulasi atau antarmuka perangkat keras salah

Tabel 2.4. Risiko pada Tahap *development*

No.	Risiko Kegagalan
1.	Staf teknis tidak berpengalaman
2.	Kurangnya pengalaman dari tim pengembang
3.	Adanya malpraktek yang dilakukan oleh staf teknis
4.	Modul tidak realistis
5.	Integrasi modul proyek yang sulit
6.	Integrasi modul yang salah dikembangkan
7.	Tidak tersedianya teknologi canggih
8.	Teknologi yang ada pada tahap awal
9.	Lebih sedikit kode yang dapat digunakan kembali yang tersedia di organisasi
10.	Pelatihan teknis yang tidak memadai untuk staf
11.	Kurangnya antarmuka yang dapat digunakan
12.	Kurangnya spesialisasi
13.	Investasi teknologi buruk
14.	Transfer teknologi lambat
15.	Rumitnya pengerjaan perangkat lunak
16.	Beban berlebihan untuk tim pengembangan
17.	Lebih dari jadwal karena pengkodean yang salah
19.	Pengkodean yang berlebihan karena kurangnya pedoman pemrograman standar

Tabel 2.5. Risiko Teknis pada implementasi dan Pemeliharaan

No.	Risiko Kegagalan
1.	Tidak ada atau kurang ruang untuk perubahan dalam sistem
2.	Aplikasi yang bergantung pada platform
3.	Biaya pemeliharaan tinggi
4.	Pelanggan yang tidak puas akan produk yang dikembangkan
5.	Produk yang dikembangkan sebagian
6.	Produktivitas rendah
7.	Kepuasan pengguna rendah
8.	Kelebihan Anggaran karena pemeliharaan korektif
9.	Kurangnya bantuan atau manual yang tepat untuk pengguna

10.	Alat dan Metode untuk Dokumen Teknis yang tidak memadai
11.	Kurangnya keamanan yang memadai dalam produk yang dikembangkan

2.3.3 Referensi Risiko dan Mitigasinya

Terdapat peneliti lain yang telah berhasil mengumpulkan risiko-risiko penyebab kegagalan pengembangan perangkat lunak secara global. Selain itu, penelitian tersebut juga mengumpulkan mitigasi risikonya untuk tiap risiko yang telah ditemukan (Verner et al., 2014). Penulis telah merangkum kumpulan risiko-risiko tersebut pada Tabel 2.6 dan telah disesuaikan berdasarkan risiko yang mempunyai kemiripan dengan risiko yang penulis temukan.

Tabel 2.6. Referensi Risiko dan Mitigasi risiko

No.	Risiko	Mitigasi risiko
1.	Tim pengembang tidak memiliki insfrastruktur yang sesuai.	Pastikan kesesuaian insfrastruktur yang dimiliki tim pengembang.
2.	Perbedaan bahasa dan budaya antara tim pengembang dan klien.	Pilih tim pengembang yang memiliki bahasa dan budaya yang sama
3.	Kurangnya komunikasi antara tim pengembang dan klien.	Berikan fasilitas untuk melakukan komunikasi.
4.	Kurangnya komunikasi dan pemahaman terkait persyaratan dari para pemangku kepentingan.	Menentukan penanggungjawab dari prioritas persyaratan dan spesifikasi.
5.	Kurangnya ketersediaan alat untuk pengembangan.	Manajemen sumberdaya harus memastikan ketersediaan alat serta keterampilan yang diperlukan oleh tim pengembang.
6.	Kurangnya komunikasi antar tim pengembang.	Meningkatkan komunikasi dengan menyesuaikan jam kerja, melakukan rapat singkat dan membuat <i>backlog</i> sebelum rapat didistribusikan.
7.	Modul yang kurang jelas.	Melakukan integrasi progresif dan membagi pekerjaan menjadi modul yang diterapkan dengan baik.
8.	Manajemen konfigurasi tidak diterapkan dengan baik.	Menerapkan dan menggunakan manajemen konfigurasi dan semua anggota tim wajib menyetujui manajemen konfigurasi umum.
9.	Kurangnya interaksi tatap muka, buruknya hubungan social, dan perbedaan budaya.	Membuat tim dengan budaya dan keterampilan yang saling melengkapi.
10.	Kurangnya pemahaman tim pengembang terkait masalah pada transfer pengetahuan.	Melakukan pelatihan kepada anggota tim terkait teknologi pengembangan perangkat lunak yang akan digunakan.
11.	Kurangnya perencanaan proyek.	Melakukan perencanaan secara detail dan manajer proyek menyampaikan secara langsung akan visi dan misi secara keseluruhan, strategi dan tugas yang harus dikerjakan.
12.	Identifikasi tanggungjawab serta peran yang buruk.	Manajer proyek wajib memberikan arahan tim terkait tanggungjawab, tujuan proyek dan peran dari masing-masing anggota tim pengembang.
13.	Tidak adanya keselarasan proses seperti metode, pengembangan, tradisi serta penekanan terkait keterlibatan pengguna.	Melembagakan serta menciptakan kesepakatan terkait norma-norma operasi.

BAB 3

Metodologi Penelitian

3.1 Metode Pengumpulan Data

Metode pengumpulan data yang digunakan dalam penelitian ini yaitu metode Delphi dengan melakukan serangkaian putaran guna memperoleh kesepakatan dari para panelis. Putaran pertanyaan pada metode Delphi membutuhkan setidaknya dua hingga tiga putaran pertanyaan untuk mendapatkan kesepakatan berdasarkan feedback yang terkontrol dari panelis (Widiasih et al., 2015). Putaran pertanyaan dapat dihentikan apabila jawaban panelis telah mencapai kata sepakat, dan apabila pada putaran ketiga belum memperoleh kesepakatan maka putaran pertanyaan dapat ditambahkan hingga memperoleh kesepakatan antar panelis (Widiasih et al., 2015). Penggunaan metode Delphi dapat memberikan hasil yang dapat dikatakan lebih baik dari pada metode survei tradisional (Okoli & Pawlowski, 2004). Metode Delphi saat ini telah digunakan dalam berbagai penelitian seperti kesehatan (Denost et al., 2019; Gunadi et al., 2020), peramalan (Flostrand et al., 2020), mengkaji prioritas suatu pilihan (Nakatsu & Iacovou, 2009), pendidikan (Knox et al., 2019), teknologi informasi (Antony & Gupta, 2019; Chien et al., 2014; Montequin et al., 2016; Wong et al., 2017) serta evaluasi (Balki et al., 2017).

3.2 Metode Delphi

Metode Delphi pertama kali diperkenalkan oleh Rand Corporation pada awal tahun 1950 ketika terjadinya ketegangan perang dingin antara Amerika Serikat dan Soviet yang bertujuan untuk mengetahui ketersediaan persenjataan nuklir serta strategi penargetan Soviet. Pada tahun 1962 Delkey, Helmer dan asosiasinya di Rand Corporation yang berada di California mulai mengembangkan metode Delphi (Dalkey & Helmer, 1962). Metode Delphi menggunakan serangkaian kuesioner berulang yang bertujuan untuk memperoleh kesepakatan berdasarkan pendapat dari para ahli (Dolan et al., 2020). Untuk memperoleh kesepakatan dari para panelis atau untuk memperoleh hasil yang diinginkan metode ini sangat bergantung pada keikutsertaan panelis yang dilakukan secara berkelanjutan (Knox et al., 2019). Metode ini dapat dikatakan berbeda dari metode-metode lainnya karena metode ini dapat lebih diandalkan untuk mengatasi masalah dengan ketidakpastian yang tinggi (Ameyaw et al., 2016). Keutungan dari penggunaan Delphi ialah memungkinkan peneliti untuk memperoleh pendapat dari para ahli yang berada diberbagai wilayah yang

berbeda karena berbasis internet, pengurangan biaya manajemen dan kekayaan informasi yang diperoleh (Fernández-Ávila et al., 2020). Disamping itu, teknik ini juga digunakan agar tidak adanya pertemuan langsung antar panelis dengan tujuan para panelis memiliki kesempatan untuk menyampaikan pandangan mereka tanpa adanya intimidasi dari panelis lain (Daniel et al., 2020). Metode Delphi klasik terdiri dari 3 tahap (Ameyaw et al., 2016), namun seperti pada Tabel 3.1 terdapat beberapa peneliti yang menggunakan metode ini dengan 2 sampai 4 tahap.

Tabel 3.1. Studi Empiris Metode Delphi.

Judul	Referensi	Tahapan Delphi
Phase 2 of CATALISE: A multinational and multidisciplinary Delphi consensus study of problems with language development: Terminology	(Bishop et al., 2017)	2 Tahap
Core outcome sets for prevention and treatment of postpartum haemorrhage: an international Delphi consensus study	(Meher et al., 2019)	2 Tahap
A systematic review and Delphi study to ascertain common risk factors for T2DM and dementia and brain-related complications of diabetes	(Dolan et al., 2020)	2 Tahap
What do we need to know about anatomy in gynaecology: A Delphi consensus study	(Koppes et al., 2020)	2 Tahap
Identifying the key characteristics of clinical fear of cancer recurrence: An international Delphi study	(Mutsaers et al., 2020)	3 Tahap
The utility of the delphi method in defining anastomotic leak following colorectal surgery	(Daniel et al., 2020)	3 Tahap
Understanding current and future issues in collaborative consumption: A four-stage Delphi study	(Barnes & Mattsson, 2016)	4 Tahap

3.3 Peserta Delphi

Sebelum melakukan penelitian ini, penulis menghubungi 52 target panelis untuk melakukan identifikasi persyaratan peserta survei Delphi, dan meminta ketersediaan untuk menjadi peserta survei Delphi. Panelis yang akan terlibat dalam penelitian ini yaitu panelis dengan pengalaman lebih dari 3 tahun dalam pengembangan perangkat lunak di Indonesia. Dari 52 target panelis yang telah diidentifikasi dan dimintai ketersediaannya diperoleh 41 panelis dengan peran yang berbeda-beda seperti pada yang tersebar di berbagai wilayah di Indonesia.

3.4 Tahap Pengumpulan Data

Penulis menggunakan 4 putaran pertanyaan untuk menjawab 3 pertanyaan pada penelitian ini. Putaran pertama bertujuan untuk mengumpulkan risiko kegagalan berdasarkan sudut pandang panelis. Putaran kedua bertujuan untuk mencari konsensus terkait risiko

kegagalan pengembangan proyek perangkat lunak yang dianggap penting. Putaran ketiga bertujuan untuk mengetahui pada tahap pengembangan mana risiko tersebut muncul. Putaran pertanyaan keempat bertujuan untuk memperoleh mitigasi risiko dari sudut pandang panelis berdasarkan risiko yang ditemukan. Penyebaran pertanyaan menggunakan *link* google form yang kemudian dikirimkan melalui *Email*.



Gambar 3.1. Tahapan Pengumpulan Data.

1. Pertanyaan Tahap Pertama

Pada pertanyaan tahap pertama ini, biasa dikenal juga dengan tahap *brainstorming* dimana penulis memberikan pertanyaan terbuka kepada para panelis. Pada tahap ini panelis diminta untuk memberikan pendapat mereka terkait risiko kegagalan proyek pengembangan perangkat lunak berdasarkan pengalaman para panelis. Adapun pertanyaan yang digunakan pada tahap pertama ini adalah “Berdasarkan pengalaman Anda apa lima (5) risiko yang paling berdampak pada kegagalan proyek pengembangan perangkat lunak?”

2. Pertanyaan Tahap Kedua

Pertanyaan pada tahap kedua ini berasal dari hasil analisis risiko kegagalan proyek pengembangan perangkat lunak yang sebelumnya diperoleh dari pertanyaan yang disebar pada tahap pertama. Hasil analisis risiko tersebut disebar kepada para panelis dengan tujuan memperoleh kesepakatan dari para panelis. Pertanyaan tahap kedua disebar dalam bentuk skala likert 1-5 dengan ketentuan sebagai berikut:

1. Risiko kurang berdampak pada kegagalan proyek pengembangan perangkat lunak.
2. Risiko berdampak pada kegagalan proyek pengembangan perangkat lunak.
3. Risiko cukup berdampak pada kegagalan proyek pengembangan perangkat lunak.
4. Risiko sangat berdampak pada kegagalan proyek pengembangan perangkat lunak.
5. Risiko amat sangat berdampak pada kegagalan proyek pengembangan perangkat lunak.

3. Pertanyaan Tahap Ketiga

Hasil dari pengolahan data tahap kedua disebarkan kembali kepada para panelis untuk mengetahui perbedaan tahap pengembangan bagi tiap risiko yang ditemukan. Pertanyaan pada tahap ketiga disebarkan dalam bentuk *check box* 6 tahap pengembangan seperti pada Gambar 2.1. Adapun pertanyaan yang digunakan pada tahap ketiga adalah “Berdasarkan pengalaman Anda, risiko-risiko dibawah ini biasanya terjadi pada tahapan mana saja? (Anda diperkenankan memilih lebih dari 1)”

4. Pertanyaan Tahap Keempat

Pertanyaan tahap keempat merupakan hasil yang diperoleh dari tahap kedua yang berupa risiko yang telah mencapai kesepakatan para panelis. Kemudian, risiko tersebut disebarkan kepada para panelis untuk dimintai tanggapan atau pandangannya terkait strategi mitigasi risiko. Pertanyaan tahap ini biasa dikenal juga dengan tahap *brainstorming* dimana penulis memberikan pertanyaan terbuka kepada para panelis dengan pertanyaan “Berdasarkan sudut pandang Anda bagaimana penanganan atau mitigasi risiko yang tepat apabila risiko-risiko dibawah ini terjadi?”

3.5 Pengolahan Data

Data yang diperoleh pada pertanyaan tahap kedua dianalisis dengan menggunakan analisis statistik untuk melihat jawaban para panelis konvergensi/konsensus atau tidak. Adapun analisis statistik yang akan digunakan dalam penelitian ini adalah sebagai berikut :

A. Standar Deviasi (SD)

Ukuran pada penilaian konsensus untuk SD adalah <1,5. Dibawah ini adalah rumus dari Standar Deviasi:

Equation 1 Standar Deviasi

$$s = \sqrt{\frac{\sum (x_i - \bar{x})^2}{n}} \quad \text{atau} \quad s = \sqrt{\frac{\sum x_i^2 - \frac{(\sum x_i)^2}{n}}{n}}$$

Dimana:

x: Jawaban panelis A terhadap variabel n

: rata-rata jawaban panelis terhadap variabel n

B. *Interquartile Range (IR)*

Ukuran untuk penilaian konsensus berdasarkan IR adalah < 1. Dibawah ini adalah rumus dari IR.

Equation 2 Interquartile Range

$$IR = (Q_3 - Q_1) / (Q_3 + Q_1)$$

Dimana : Q₃ merupakan kuartil atas dan Q₁ adalah kuartil bawah

Adapun rumus kuartil Q₁, Q₂, dan Q₃ adalah sebagai berikut :

Equation 3 Q₁

$$Q_1 = \frac{(n+1) \cdot \frac{Q_1 - Q_0}{Q_1 - Q_0} \cdot 2}{n+1}$$

Equation 4 Q₂

$$Q_2 = \frac{(n+1) \cdot \frac{Q_2 - Q_0}{Q_2 - Q_0}}{n+1}$$

Equation 5 Q₃

$$Q_3 = \frac{(n+1) \cdot \frac{Q_3 - Q_0}{Q_3 - Q_0} \cdot 2}{n+1}$$

C. *Mean*

Nilai *mean* merupakan nilai rata-rata yang diberikan pada variabel tertentu dan berikut rumus untuk mencari nilai *mean*:

Equation 6 Mean

$$\bar{x} = \frac{\sum_{j=1}^n x_j}{n}$$

Hasil analisis statistik untuk menyatakan apakah setiap variabel merupakan konsensus atau konvergen adalah jika nilai SD <1,5 dan nilai IR <1 (Giannarou & Zervas, 2014). Namun, jika salah satu baik SD maupun IR tidak memenuhi syarat tersebut, maka dapat disimpulkan bahwa variabel tersebut tidak konvergen atau tidak konsensus (tidak disepakati). Setelah mengetahui nilai SD dan IR telah mencapai konsensus atas variabel yang diusulkan, maka langkah selanjutnya adalah dengan memberi peringkat dari variabel tersebut berdasarkan nilai *mean* (rata-rata) yang tertinggi untuk setiap variabelnya.

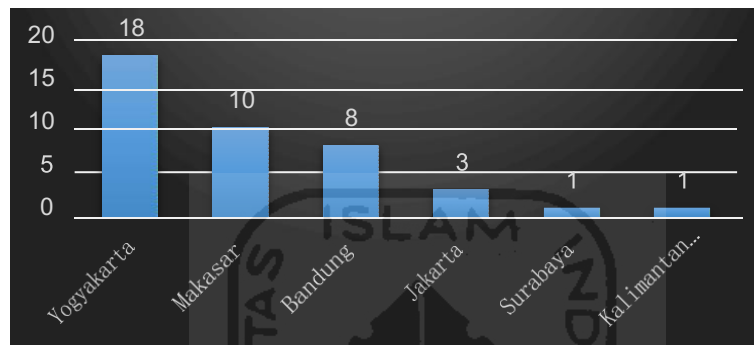
BAB 4

Hasil dan Pembahasan

4.1 Peran dan Lokasi panelis

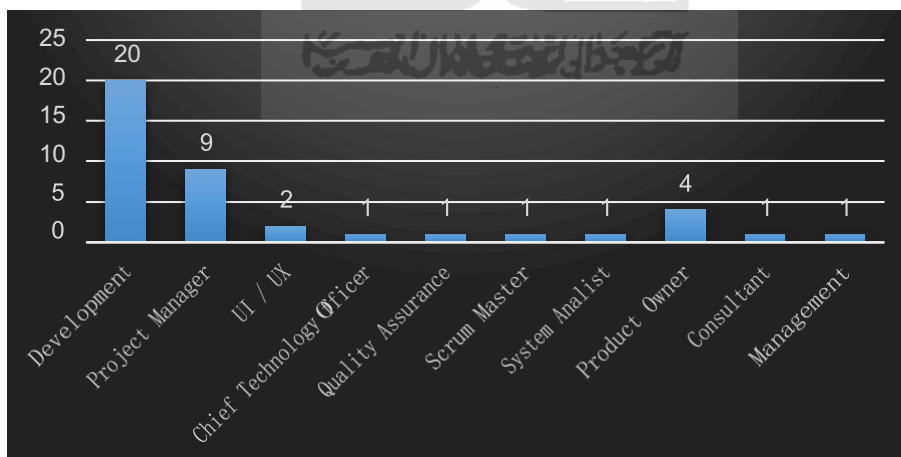
4.1.1 Tahap Pertama

Pada pengambilan data tahap pertama, penulis menyebarkan pertanyaan terbuka kepada 41 panelis untuk memberikan pandangannya terkait 5 risiko yang dianggap paling berdampak pada kegagalan proyek pengembangan perangkat lunak berdasarkan pengalaman dari masing-masing panelis.



Gambar 4.1. Lokasi Panelis Tahap Pertama.

Gambar 4.1 merupakan lokasi atau wilayah dari panelis yang berkontribusi pada pengambilan data tahap pertama dengan memberikannya. Pada pengambilan data tahap pertama ini Yogyakarta merupakan kota yang mendominasi lokasi panelis dengan jumlah 18 panelis.

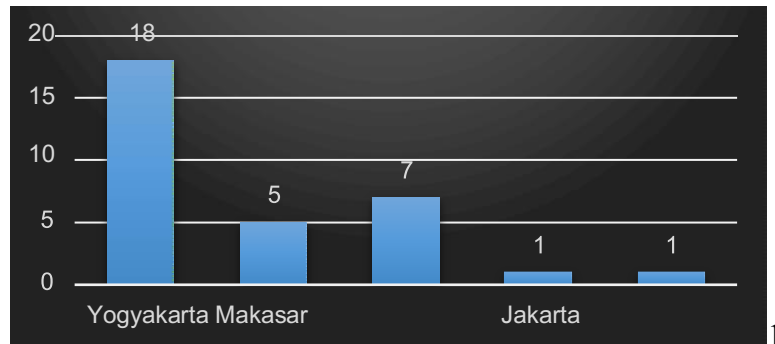


Gambar 4.2. Peran Panelis Tahap Pertama.

Gambar 4.2 menjelaskan peran tiap panelis dalam proyek pengembangan perangkat lunak yang terlibat pada tahap pertama dan didominasi oleh *development* sebanyak 20 panelis dengan peran *development*.

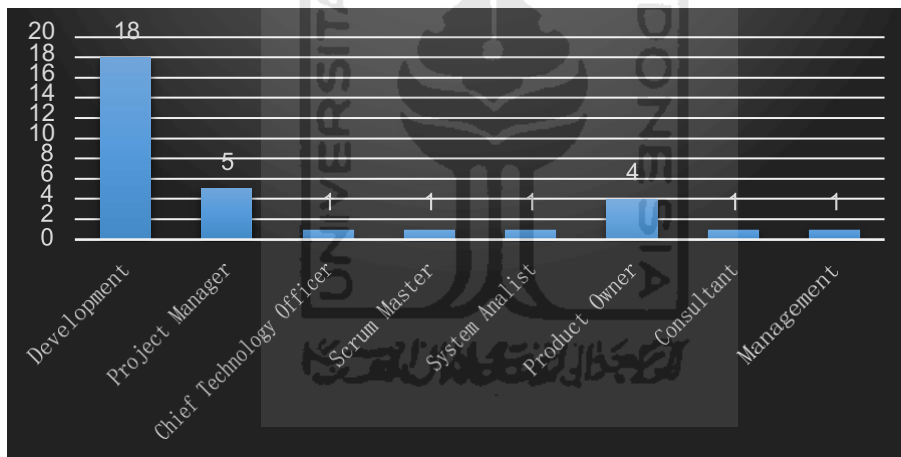
4.1.2 Tahap Kedua

Pertanyaan tahap ini disebarakan kepada 41 panelis, dan sebanyak 32 panelis bersedia memberikan tanggapannya.



Gambar 4.3. Lokasi Panelis Tahap Kedua.

Gambar 4.3 merupakan lokasi atau wilayah dari panelis yang berkontribusi pada pengambilan data tahap kedua. Lokasi yang mendominasi dalam pengambilan data tahap ini adalah kota Yogyakarta dengan jumlah 18 panelis.

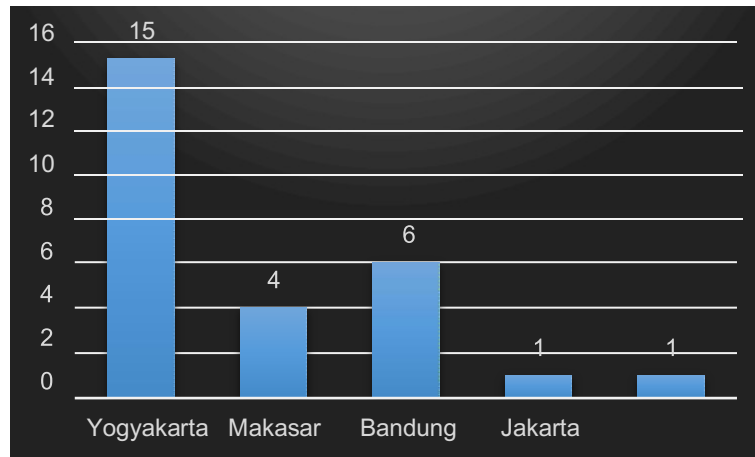


Gambar 4.4. Peran Panelis Tahap Kedua.

Gambar 4.4 menjelaskan peran tiap panelis dalam proyek pengembangan perangkat lunak yang terlibat pada tahap kedua. Peran panelis yang mendominasi dalam pengambilan data tahap ini adalah *development* dengan jumlah 18 panelis.

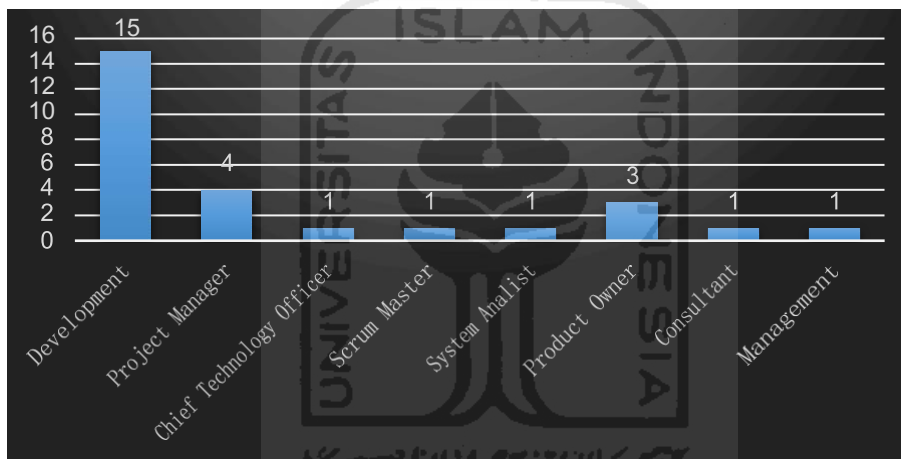
4.1.3 Tahap Ketiga

Pada pengambilan data tahap ketiga, penulis menyebarkan pertanyaan kepada 32 panelis dan 27 panelis berkontribusi pada putaran tahap ini.



Gambar 4.5. Lokasi Panelis Tahap Ketiga.

Gambar 4.5 merupakan rincian lokasi panelis mengerjakan proyek pengembangan perangkat lunak yang kembali ikut serta pada tahap ketiga, dimana kota Yogyakarta masih mendominasi lokasi panelis dengan jumlah 15 panelis.

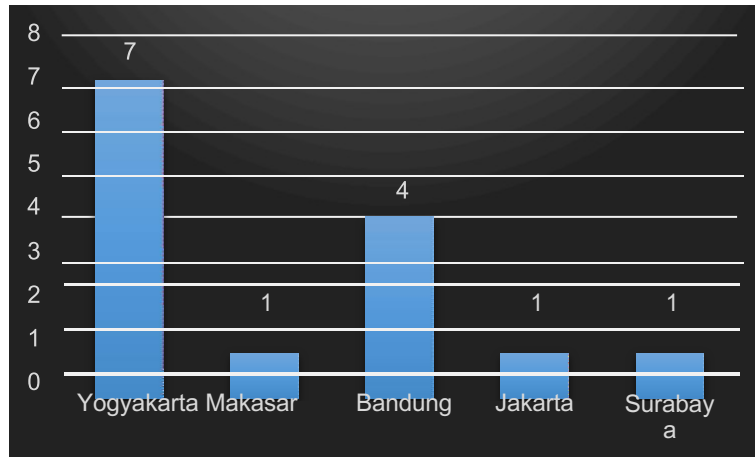


Gambar 4.6. Peran Panelis Tahap Ketiga.

Gambar 4.6 menjelaskan peran dari tiap panelis dalam proyek pengembangan perangkat lunak yang kembali ikut serta pada tahap ketiga, panelis yang berperan sebagai *development* masih mendominasi keikutsertaannya dengan jumlah 15 panelis.

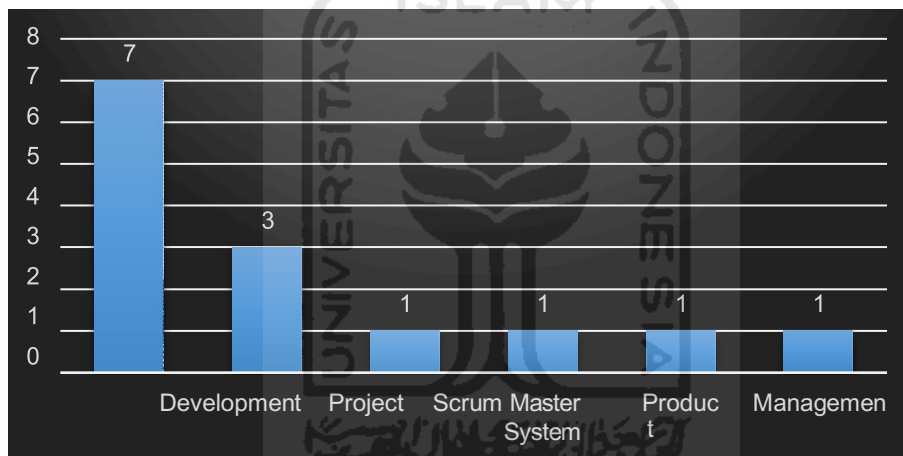
4.1.4 Tahap Keempat

Pada pengambilan data tahap keempat penulis menyebarkan pertanyaan kepada 27 panelis, 14 panelis bersedia memberikan tanggapannya dan sebanyak 13 panelis memilih mundur sebagai peserta survei Delphi.



Gambar 4.7. Lokasi Panelis Tahap Keempat.

Gambar 4.7 merupakan lokasi atau wilayah dari panelis yang berkontribusi pada pengambilan data tahap keempat. Pada pengambilan data tahap keempat ini Yogyakarta masih mendominasi lokasi panelis dengan jumlah 7 panelis.



Gambar 4.8. Peran Panelis Tahap Keempat.

Gambar 4.8 menjelaskan peran tiap panelis dalam proyek pengembangan perangkat lunak yang terlibat pada tahap keempat. Pada tahap ini peran *development* masih mendominasi dengan jumlah sebanyak 7 panelis dengan peran *development*.

4.2 Hasil

4.2.1 Hasil Data Tahap Pertama

Hasil dari tahap pertama ini memperoleh 205 risiko kegagalan proyek pengembangan perangkat lunak berdasarkan tanggapan dari 41 panelis untuk lebih jelas dapat dilihat pada Tabel 4.1.

Tabel 4.1. Hasil Risiko Tahap Pertama.

Kode	Risiko Kegagalan	Jumlah Tanggapan
R.01	Kurangnya analisis kebutuhan/proses bisnis	21
R.02	Estimasi waktu tidak tepat	19
R.03	Estimasi biaya tidak sesuai	18
R.04	Kurangnya komunikasi antar tim / pengguna dan pengembang	18
R.05	Perencanaan sistem tidak jelas	17
R.06	Kurangnya keahlian dan pengetahuan tim pengembang	16
R.07	Tim yang tidak profesional	12
R.08	Kurangnya tim pengembang	11
R.09	Kegagalan mendapatkan komitmen pengguna	9
R.10	Persyaratan tidak jelas	9
R.11	Manajemen proyek tidak jelas	8
R.12	Kurangnya pemahaman pengguna	7
R.13	Fasilitas kurang memadai	7
R.14	Kurangnya kepercayaan pengguna	6
R.15	Target pasar tidak tepat	5
R.16	Ruang lingkup tidak sesuai	5
R.17	Kurangnya kerjasama tim	4
R.18	Buruknya pemantauan sistem	4
R.19	Akuisisi pengguna rendah	3
R.20	Perbedaan budaya	2
R.21	Teknologi belum teruji	2
R.22	Kurangnya data	2
Total		205

Berdasarkan hasil yang diperoleh dari tahap pertama yang dapat dilihat pada Tabel 4.1 menunjukkan bahwa Kurangnya analisis kebutuhan / proses bisnis menempati risiko yang paling banyak disebutkan oleh para panelis. Sedangkan perbedaan budaya, teknologi yang belum teruji serta kurangnya data menjadi risiko yang paling jarang disebutkan oleh para panelis. Meskipun perbedaan budaya jarang disebutkan, namun terbukti bahwa perbedaan budaya menjadi salah satu risiko yang berdampak pada kegagalan proyek pengembangan perangkat lunak.

4.2.2 Hasil Data Tahap Kedua

Pengambilan data tahap kedua, bertujuan untuk memperoleh konsensus berdasarkan hasil yang diperoleh dari tahap pertama (

Tabel 4.1. Hasil Risiko Tahap Pertama). Penilaian konsensus menggunakan skala likert 1-5 dimulai dari risiko yang kurang berdampak sampai risiko yang amat sangat berdampak pada kegagalan proyek perangkat lunak. Tabel 4.2 sampai dengan Tabel 4.4 merupakan penilaian konsensus berdasarkan sudut pandang dari masing-masing peran panelis.

Tabel 4.2. Penilaian Konsensus *Development*.

Kode	Risiko Kegagalan	Mean	SD	IR	Evaluasi	
					SD	IR
R.01	Kurangnya analisis kebutuhan/proses bisnis	4,67	0,58	0,75	Kon	Kon
R.02	Estimasi waktu tidak tepat	4,33	0,58	1	Kon	Kon
R.03	Estimasi biaya tidak sesuai	4,22	0,71	1	Kon	Kon
R.04	Kurangnya komunikasi antar tim / pengguna dan pengembang	4,56	0,60	1	Kon	Kon
R.05	Perencanaan sistem tidak jelas	4,83	0,37	0	Kon	Kon
R.06	Kurangnya keahlian dan pengetahuan tim pengembang	4,44	0,68	1	Kon	Kon
R.07	Tim yang tidak profesional	4,56	0,60	1	Kon	Kon
R.08	Kurangnya tim pengembang	4,06	0,62	0	Kon	Kon
R.09	Kegagalan mendapatkan komitmen pengguna	4,28	0,73	1	Kon	Kon
R.10	Persyaratan tidak jelas	4,11	0,74	1	Kon	Kon
R.11	Manajemen proyek tidak jelas	4,50	0,69	1	Kon	Kon
R.12	Kurangnya pemahaman pengguna	3,56	0,76	1	Kon	Kon
R.13	Fasilitas kurang memadai	3,94	0,78	0	Kon	Kon
R.14	Kurangnya kepercayaan pengguna	3,61	0,76	1	Kon	Kon
R.15	Target pasar tidak tepat	4,00	0,88	1,75	Kon	Div
R.16	Ruang lingkup tidak sesuai	3,94	0,52	0	Kon	Kon
R.17	Kurangnya kerjasama tim	4,44	0,50	1	Kon	Kon
R.18	Buruknya pemantauan sistem	4,17	0,69	1	Kon	Kon
R.19	Akuisisi pengguna rendah	2,94	0,85	1,5	Kon	Div
R.20	Perbedaan budaya	2,72	0,87	1	Kon	Kon
R.21	Teknologi belum teruji	3,39	0,89	1	Kon	Kon
R.22	Kurangnya data	4,17	0,83	1	Kon	Kon

Berdasarkan *input* yang diberikan oleh *Development* pada Tabel 4.2, dari 22 risiko yang diukur menggunakan SD dan IR, diperoleh 20 risiko yang mencapai nilai konsensus dan 2 risiko tidak mencapai nilai konsensus. Risiko pertama yang tidak mencapai konsensus yaitu R.15 (Target pasar tidak tepat) dengan nilai SD 0,88 dan nilai IR 1,75 serta risiko kedua yaitu R.19 (Akuisisi pengguna rendah) dengan nilai SD 0,85 dan nilai IR 1,5.

Tabel 4.3. Penilaian Konsensus *Product Owner*.

Kode	Risiko Kegagalan	Mean	SD	IR	Evaluasi	
					SD	IR
R.01	Kurangnya analisis kebutuhan/proses bisnis	4,5	0,87	0,5	Kon	Kon
R.02	Estimasi waktu tidak tepat	4	0,71	0,5	Kon	Kon
R.03	Estimasi biaya tidak sesuai	3,75	0,83	1,25	Kon	Div

R.04	Kurangnya komunikasi antar tim / pengguna dan pengembang	3,5	1,12	1,5	Kon	Div
R.05	Perencanaan sistem tidak jelas	3,75	0,43	0,25	Kon	Kon
R.06	Kurangnya keahlian dan pengetahuan tim pengembang	4	1	2	Kon	Div
R.07	Tim yang tidak profesional	4,25	0,83	1,25	Kon	Div
R.08	Kurangnya tim pengembang	3,5	1,12	1,5	Kon	Div
R.09	Kegagalan mendapatkan komitmen pengguna	3,5	1,12	1,5	Kon	Div
R.10	Persyaratan tidak jelas	3,5	0,5	1	Kon	Kon
R.11	Manajemen proyek tidak jelas	3,5	0,5	1	Kon	Kon
R.12	Kurangnya pemahaman pengguna	2,75	0,43	0,25	Kon	Kon
R.13	Fasilitas kurang memadai	3,25	0,83	1,25	Kon	Div
R.14	Kurangnya kepercayaan pengguna	3	0,71	0,5	Kon	Kon
R.15	Target pasar tidak tepat	3,25	0,83	1,25	Kon	Div
R.16	Ruang lingkup tidak sesuai	3,25	0,83	1,25	Kon	Div
R.17	Kurangnya kerjasama tim	3,75	0,83	1,25	Kon	Div
R.18	Buruknya pemantauan sistem	2,75	0,83	1,25	Kon	Div
R.19	Akuisisi pengguna rendah	3,5	0,5	1	Kon	Kon
R.20	Perbedaan budaya	3,5	0,5	1	Kon	Kon
R.21	Teknologi belum teruji	3,25	0,83	1,25	Kon	Div
R.22	Kurangnya data	3	0,71	0,5	Kon	Kon

Berdasarkan *input* yang diberikan oleh *Product Owner* pada Tabel 4.3, dari 22 risiko yang diukur menggunakan SD dan IR, diperoleh 10 risiko yang mencapai nilai konsensus dan 12 risiko tidak mencapai nilai konsensus. Ke-12 risiko tersebut yaitu R.03 (Estimasi biaya tidak sesuai), R.07 (Tim yang tidak profesional), R.13 (Fasilitas kurang memadai), R.15 (Target pasar tidak tepat), R.16 (Ruang lingkup tidak sesuai), R.17 (Kurangnya kerjasama tim), R.18 (Buruknya pemantauan sistem) dan R.21 (Teknologi belum teruji) dengan nilai SD 0,83 dan IR 1,25. R.04 (Kurangnya komunikasi antar tim / pengguna dan pengembang), R.08 (Kurangnya tim pengembang), dan R.09 (Kegagalan mendapatkan komitmen pengguna) dengan nilai SD 1,12 dan IR 1,5. Kemudian R.06 (Kurangnya keahlian dan pengetahuan tim pengembang) dengan nilai SD 1 dan IR 2.

Tabel 4.4. Penilaian Konsensus *Project Manager*.

Kode	Risiko Kegagalan	Mean	SD	IR	Evaluasi	
					SD	IR
R.01	Kurangnya analisis kebutuhan/proses bisnis	4,6	0,8	0	Kon	Kon
R.02	Estimasi waktu tidak tepat	4	0,63	0	Kon	Kon
R.03	Estimasi biaya tidak sesuai	3,8	1,17	2	Kon	Div
R.04	Kurangnya komunikasi antar tim / pengguna dan pengembang	4,2	0,75	1	Kon	Kon
R.05	Perencanaan sistem tidak jelas	4,8	0,4	0	Kon	Kon
R.06	Kurangnya keahlian dan pengetahuan tim pengembang	4	0,63	0	Kon	Kon

R.07	Tim yang tidak profesional	4,8	0,4	0	Kon	Kon
R.08	Kurangnya tim pengembang	3,8	0,75	1	Kon	Kon
R.09	Kegagalan mendapatkan komitmen pengguna	4	0,63	0	Kon	Kon
R.10	Persyaratan tidak jelas	4,2	0,75	1	Kon	Kon
R.11	Manajemen proyek tidak jelas	4,2	0,4	0	Kon	Kon
R.12	Kurangnya pemahaman pengguna	3,6	1,02	1	Kon	Kon
R.13	Fasilitas kurang memadai	3,6	1,02	1	Kon	Kon
R.14	Kurangnya kepercayaan pengguna	3,6	0,8	1	Kon	Kon
R.15	Target pasar tidak tepat	4,6	0,49	1	Kon	Kon
R.16	Ruang lingkup tidak sesuai	4,4	0,49	1	Kon	Kon
R.17	Kurangnya kerjasama tim	4,2	0,75	1	Kon	Kon
R.18	Buruknya pemantauan sistem	4	1,1	1	Kon	Kon
R.19	Akuisisi pengguna rendah	3,8	0,98	0	Kon	Kon
R.20	Perbedaan budaya	2,2	0,75	1	Kon	Kon
R.21	Teknologi belum teruji	3,4	0,8	1	Kon	Kon
R.22	Kurangnya data	4,2	0,75	1	Kon	Kon

Berdasarkan *input* yang diberikan oleh *Project Manager* pada Tabel 4.4, dari 22 risiko yang diukur menggunakan SD dan IR, diperoleh 21 risiko yang mencapai nilai konsensus dan 1 risiko tidak mencapai nilai konsensus yaitu R.03 (Estimasi biaya tidak sesuai). dengan nilai SD 1,17 dan nilai IR 2.

Tabel 4.5. Penilaian Konsensus Kelompok.

Kode	Risiko Kegagalan	Mean	SD	IR	Evaluasi	
					SD	IR
R.01	Kurangnya analisis kebutuhan/proses bisnis	4,65	0,64	0,25	Kon	Kon
R.02	Estimasi waktu tidak tepat	4,21	0,59	1	Kon	Kon
R.03	Estimasi biaya tidak sesuai	4,06	0,82	1,25	Kon	Div
R.04	Kurangnya komunikasi antar tim / pengguna dan pengembang	4,34	0,81	1	Kon	Kon
R.05	Perencanaan sistem tidak jelas	4,68	0,52	1	Kon	Kon
R.06	Kurangnya keahlian dan pengetahuan tim pengembang	4,25	0,75	1	Kon	Kon
R.07	Tim yang tidak profesional	4,56	0,65	1	Kon	Kon
R.08	Kurangnya tim pengembang	3,84	0,75	1	Kon	Kon
R.09	Kegagalan mendapatkan komitmen pengguna	3,96	0,91	2	Kon	Div
R.10	Persyaratan tidak jelas	3,90	0,80	1,25	Kon	div
R.11	Manajemen proyek tidak jelas	4,28	0,67	1	Kon	Kon
R.12	Kurangnya pemahaman pengguna	3,40	0,82	1	Kon	Kon
R.13	Fasilitas kurang memadai	3,68	0,84	1	Kon	Kon
R.14	Kurangnya kepercayaan pengguna	3,59	0,78	1	Kon	Kon
R.15	Target pasar tidak tepat	3,93	0,86	2	Kon	Div
R.16	Ruang lingkup tidak sesuai	3,93	0,65	0	Kon	Kon
R.17	Kurangnya kerjasama tim	4,31	0,63	1	Kon	Kon
R.18	Buruknya pemantauan sistem	3,93	0,86	0,25	Kon	Kon
R.19	Akuisisi pengguna rendah	3,12	0,85	1	Kon	Kon
R.20	Perbedaan budaya	2,81	0,88	1	Kon	Kon
R.21	Teknologi belum teruji	3,34	0,81	1	Kon	Kon

R.22	Kurangnya data	3,87	0,89	2	Kon	Div
------	----------------	------	------	---	-----	-----

Berdasarkan penilaian konsensus kelompok diperoleh 17 risiko yang mencapai nilai konsensus dan 5 risiko yang tidak mencapai nilai konsensus seperti R.03 (Estimasi biaya tidak sesuai) dengan nilai SD 0,82 dan IR 1,25, R.09 (Kegagalan mendapatkan komitmen pengguna) dengan nilai SD 0,82 dan IR 1,25, R.10 (Persyaratan tidak jelas) dengan nilai SD 0,82 dan IR 1,25 R.15 (Target pasar tidak tepat) dengan nilai SD 0,82 dan IR 1,25, dan R.22 (Kurangnya data) dengan nilai SD 0,82 dan IR 1,25.

4.2.3 Hasil Data Tahap Ketiga

Pertanyaan tahap ini berbentuk *check box* pada google form dimana panelis diminta untuk memberikan tanggapannya terkait ke-17 risiko yang telah mencapai konsensus (Tabel 4.5). Adapun tahapan pengembangannya dapat dilihat pada Gambar 2.1. Tahap Pengembangan. Data yang diperoleh dari tahap ini dirangkum dan disajikan dengan bentuk persentase untuk mempermudah membaca pola risiko berdasarkan tahap pengembangannya. Hasil data tahap ketiga dapat dilihat pada Tabel 4.6.

Tabel 4.6. Risiko berdasarkan tahap pengembangan

Kode	Risiko Kegagalan	Tahapan Pengembangan Perangkat Lunak					
		Planning	Analyze	Design	Develop	Implement	Maintain
R.01	Kurangnya analisis kebutuhan/proses bisnis	27,14%	37,14%	20,00%	8,57%	5,71%	1,43%
R.02	Estimasi waktu tidak tepat	29,41%	25,00%	16,18%	17,65%	10,29%	1,47%
R.04	Kurangnya komunikasi antar tim / pengguna dan pengembang	15,07%	17,81%	19,18%	27,40%	16,44%	4,11%
R.05	Perencanaan sistem tidak jelas	35,19%	31,48%	14,81%	7,41%	7,41%	3,70%
R.06	Kurangnya keahlian dan pengetahuan tim pengembang	15,94%	15,94%	17,39%	28,99%	14,49%	7,25%
R.07	Tim yang tidak profesional	18,60%	16,28%	17,44%	23,26%	15,12%	9,30%
R.08	Kurangnya tim pengembang	19,61%	17,65%	11,76%	29,41%	13,73%	7,84%
R.11	Manajemen proyek tidak jelas	30,88%	20,59%	13,24%	17,65%	10,29%	7,35%
R.12	Kurangnya pemahaman pengguna	19,64%	26,79%	16,07%	5,36%	19,64%	12,50%
R.13	Fasilitas kurang memadai	22,22%	11,11%	12,70%	23,81%	22,22%	7,94%
R.14	Kurangnya kepercayaan pengguna	23,73%	13,56%	10,17%	15,25%	16,95%	20,34%
R.16	Ruang lingkup tidak	25,81%	32,26%	16,13%	14,52%	8,06%	3,23%

	sesuai						
R.17	Kurangnya kerjasama tim	15,29%	15,29%	17,65%	23,53%	16,47%	11,76%
R.18	Buruknya pemantauan sistem	14,29%	17,46%	12,70%	14,29%	20,63%	20,63%
R.19	Akuisisi pengguna rendah	14,89%	12,77%	17,02%	4,26%	27,66%	23,40%
R.20	Perbedaan budaya	19,40%	22,39%	14,93%	11,94%	19,40%	11,94%
R.21	Teknologi belum teruji	17,86%	16,07%	8,93%	17,86%	23,21%	16,07%

4.2.4 Hasil Data Tahap Keempat

Data yang diperoleh dari tahap keempat bersifat kualitatif, analisis data yang digunakan menggunakan teknik reduksi data dengan merangkum, mengelompokkan dan memilih hal pokok guna menyajikan gambaran yang lebih jelas. Setelah data direduksi, langkah selanjutnya penulis menyajikan data dalam bentuk tabel yang dapat dilihat pada Tabel 4.7.

Tabel 4.7. Hasil Data Tahap Keempat.

Kode	Risiko Kegagalan	Strategi Mitigasi Risiko
R.01	Kurangnya analisis kebutuhan / proses bisnis	Adakan komunikasi dengan klien untuk memperjelas proses bisnis dan <i>scope</i> yang belum terpenuhi.
		Lakukan analisis dan identifikasi secara menyeluruh dengan membagi skala prioritas untuk proses bisnis yang akan dibuat.
		Mendatangkan ahli pada bidang yang sedang dianalisa agar tidak terjadi <i>gap requirement system</i> dan pengembang melakukan pendekatan kepada klien untuk menjalin komunikasi antarkeduanya.
R.02	Estimasi waktu tidak tepat	Membuat jadwal, target proyek dan menambah waktu kerja (lembur) di fase awal proyek dengan tujuan utama untuk memastikan semua fungsi sistem yang akan dikerjakan sesuai ekspektasi dan rencana.
		Membuat dokumentasi yang berfungsi sebagai riwayat aktivitas / pekerjaan apa saja yang telah dilakukan. Sehingga pekerjaan yang dilakukan dapat dianalisis rekam jejaknya.
R.04	Kurangnya komunikasi antar tim / pengguna dan pengembang	Selalu melakukan evaluasi dan memperbaiki pola komunikasi yang ada dengan mendekati tempat duduk antar tim agar komunikasi antar tim dapat dilakukan dengan lebih intens.
		Membuat dokumentasi tiap-tiap pekerjaan buka peluang komunikasi yang lebih mudah baik antar tim pengembang maupun pengguna dengan tim dengan cara membuka forum diskusi menyampaikan pendapat.
R.05	Perencanaan sistem tidak jelas	Tahan proyek dan segera lakukan perencanaan ulang sistem dengan klien dan tim pengembang sampai menemui sebuah kesepakatan terkait perangkat lunak yang akan dikembangkan.
		Mengevaluasi dilakukan secara berkala terkait perancangan proses bisnis dan SOP serta lakukan analisis kebutuhan perangkat lunak yang akan pengembangan.

R.06	Kurangnya keahlian / pengetahuan tim pengembang	Memasukkan riset ke dalam masa pengerjaan proyek atau selama proses <i>pitching</i> atau, menambah tim khusus yang melakukan riset pada awal pekerjaan dimulai.
		Datangkan pelatih untuk memberikan pelatihan kepada tim pengembang dan tambah literasi untuk mengatasi kurangnya keahlian dan pengetahuan sampai dilihat cukup mampu untuk mengerjakan proyek.
		Rekrut ahli atau pengembang senior yang mampu mengerjakan proyek yang akan dikembangkan.
R.07	Tim yang tidak profesional	Komunikasikan titik permasalahan di internal tim untuk menekankan langkah-langkah perbaikan, target, aturan, kerjasama, untuk meningkatkan profesionalitas.
		Berikan kesempatan tim untuk meningkatkan profesionalitas dan berikan hadiah atau hukuman.
		Buat ulang tim baru dengan mengganti tim yang tidak profesional dengan yang lebih profesional.
R.08	Kurangnya tim pengembang	Tingkatkan <i>cross function</i> setiap anggota tim sebelum melakukan perekrutan, kemudian jadwalkan ulang pekerjaan sesuai kemampuan tim dan komunikasikan pada klien untuk menambah waktu pengerjaan atau menyederhanakan desain antarmuka dan fungsi.
		Apabila klien tidak setuju segera lakukan evaluasi pada bagian mana tim dibutuhkan dan tambah anggota tim sesuai dengan posisi dan jumlah tim yang dibutuhkan.
R.11	Manajemen proyek tidak jelas	Memperjelas <i>scope</i> yang diinginkan manajemen, dengan membuat jadwal yang harus dilakukan dan <i>project charter</i> yang lengkap dengan melibatkan semua anggota tim. Sehingga tim dapat mengetahui target yang harus dicapai dan pekerjaan menjadi lebih terstruktur.
		Membuat atau menerapkan <i>project management framework</i> agar lebih jelas dan jika diperlukan melakukan perekrutan <i>project manager</i> dengan pengalaman yang tinggi.
R.12	Kurangnya pemahaman pengguna	Melakukan sosialisasi dan <i>transfer knowledge</i> yang diadakan bertahap dengan cara pendampingan intensif kepada mengenai perangkat lunak yang akan digunakan.
		Membuat buku panduan atau Kerangka acuan terhadap pengguna sebagai panduan umum mengenai perangkat lunak.
R.13	Fasilitas kurang memadai	Optimalkan fasilitas yang tersedia atau melakukan pengadaan fasilitas yang dibutuhkan pada saat pengembangan, apabila dimungkinkan minta klien untuk menyediakan fasilitas yang dibutuhkan.
R.14	Kurangnya kepercayaan pengguna	Melakukan komunikasi dengan pengguna untuk menyusun komitmen ulang antara kedua pihak dan adakan acara tertentu untuk menjembatani pertemuan tim pengembang dan pengguna.
		Meyakinkan pengguna terhadap tim pengembang dengan memberikan timeline dan progres pengerjaan tepat waktu dan meyakinkan user bahwa ekspektasi mereka dapat dipenuhi.

		Menyusun strategi digital dengan memanfaatkan <i>analytic tools</i> sebagai <i>review</i> atau <i>report</i> terhadap pengguna.
R.16	Ruang lingkup tidak sesuai	Lakukan komunikasi secara lanjut dengan Klien memperjelas tujuan awal pembuatan produk dan pelajari ruang lingkup yang dibutuhkan
		Segera melakukan analisis ulang dan evaluasi ruang lingkup pekerjaan dengan menentukan skala prioritas agar ruang lingkup bisa lebih sesuai dan lakukan diskusi secara lanjut dengan klien untuk mempelajari ruang lingkup yang dibutuhkan.
R.17	Kurangnya kerjasama tim	Memperkuat peran <i>Project Manager</i> , <i>Product Owner</i> , dan <i>Scrum Master</i> untuk melakukan manajemen, identifikasi serta evaluasi kondisi tim.
		Berikan arahan untuk meningkatkan kerjasama dengan cara memberikan <i>brainstorming</i> , kolaborasi tim secara rutin, dan jika diperlukan memperkecil anggota tim.
R.18	Buruknya pemantauan sistem	Setiap divisi perlu melakukan penjadwalan pemantauan sistem, kemudian dokumentasikan serinci mungkin dan melaporkannya untuk dilakukan evaluasi serta verifikasi.
		Meminta keterlibatan pengguna untuk ikut serta melakukan pemantauan sistem.
R.19	Akuisisi pengguna rendah	Melakukan promosi dengan cara memperbanyak komunikasi dan melakukan pendekatan persuasif kepada pengguna
		Melakukan segmentasi, dan membagi pengetahuan secara berkala kepada pengguna dengan melakukan pendampingan secara intensif.
R.20	Perbedaan budaya	Mengakomodir perbedaan budaya dengan cara mengadakan sosialisasi, <i>training</i> , <i>event gathering</i> , <i>tech talk</i> untuk menjalin kerjasama antar tim.
		Mencoba memberikan toleransi, menghormati dan melakukan komunikasi yang baik antar tim agar proyek dapat diterima oleh semua budaya.
R.21	Teknologi belum teruji	Lakukan penelitian dan pengujian teknologi dengan mengoptimalkan peran <i>Research and Development</i> (R&D) untuk mensimulasikan perangkat lunak di lapangan (CRUD data sederhana) agar dapat mengetahui keefektifan teknologi yang digunakan.
		Batasi penggunaan teknologi baru atau hindari penggunaan teknologi yang belum teruji.

4.3 Pembahasan

4.3.1 Risiko Kegagalan

Pada penilaian konsensus kelompok dan individu ditemukan bahwa terdapat perbedaan risiko seperti yang dapat dilihat pada Tabel 4.2. Penilaian Konsensus *Development* diperoleh 20 risiko kegagalan yang mencapai konsensus, pada

Tabel 4.3. Penilaian *Konsensus Product Owner* diperoleh 10 risiko kegagalan yang mencapai konsensus, pada Tabel 4.4. Penilaian Konsensus *Project Manage* diperoleh 21 risiko kegagalan yang mencapai konsensus. Sedangkan penilaian konsensus yang dihitung

dari keseluruhan panelis atau kelompok yang terdapat pada Tabel 4.5 memperoleh 17 risiko yang mencapai konsensus antar panelis.

Dari 22 risiko yang diukur menggunakan SD dan IR, diperoleh 5 risiko yang tidak mencapai konsensus (Tabel 4.5. Penilaian Konsensus Kelompok.). Hal tersebut menarik karena ke-5 risiko tersebut memiliki jumlah tanggapan (Tabel 4.1) dan nilai *mean* (Tabel 4.5) yang tinggi. Seperti estimasi biaya tidak sesuai dengan 18 tanggapan dan nilai *mean* 4,06, kegagalan mendapatkan komitmen pengguna dengan 9 tanggapan dan nilai *mean* 3,96, persyaratan tidak jelas dengan 9 tanggapan dan nilai *mean* 3,90, target pasar tidak tepat dengan 5 tanggapan dan nilai *mean* 3,93 dan terakhir kurangnya data tepat dengan 2 tanggapan dan nilai *mean* 3,87. Untuk ke-17 risiko yang mencapai konsensus kelompok kemudian di ranking berdasarkan nilai rata-rata tertinggi, seperti pada Tabel 4.8.

Tabel 4.8. Risiko Kegagalan yang Mecipai Nilai Konsensus Kelompok.

Kode	Risiko Kegagalan	Mean	Rank
R.05	Perencanaan sistem tidak jelas	4,68	1
R.01	Kurangnya analisis kebutuhan/proses bisnis	4,65	2
R.07	Tim yang tidak profesional	4,56	3
R.04	Kurangnya komunikasi antar tim / pengguna dan pengembang	4,34	4
R.17	Kurangnya kerjasama tim	4,31	5
R.11	Manajemen proyek tidak jelas	4,28	6
R.06	Kurangnya keahlian dan pengetahuan tim pengembang	4,25	7
R.02	Estimasi waktu tidak tepat	4,21	8
R.16	Ruang lingkup tidak sesuai	3,93	9
R.18	Buruknya pemantauan sistem	3,93	9
R.08	Kurangnya tim pengembang	3,84	10
R.13	Fasilitas kurang memadai	3,68	11
R.14	Kurangnya kepercayaan pengguna	3,59	12
R.12	Ruang lingkup tidak sesuai	3,40	13
R.21	Teknologi belum teruji	3,34	14
R.19	Akuisisi pengguna rendah	3,12	15
R.20	Perbedaan budaya	2,81	16

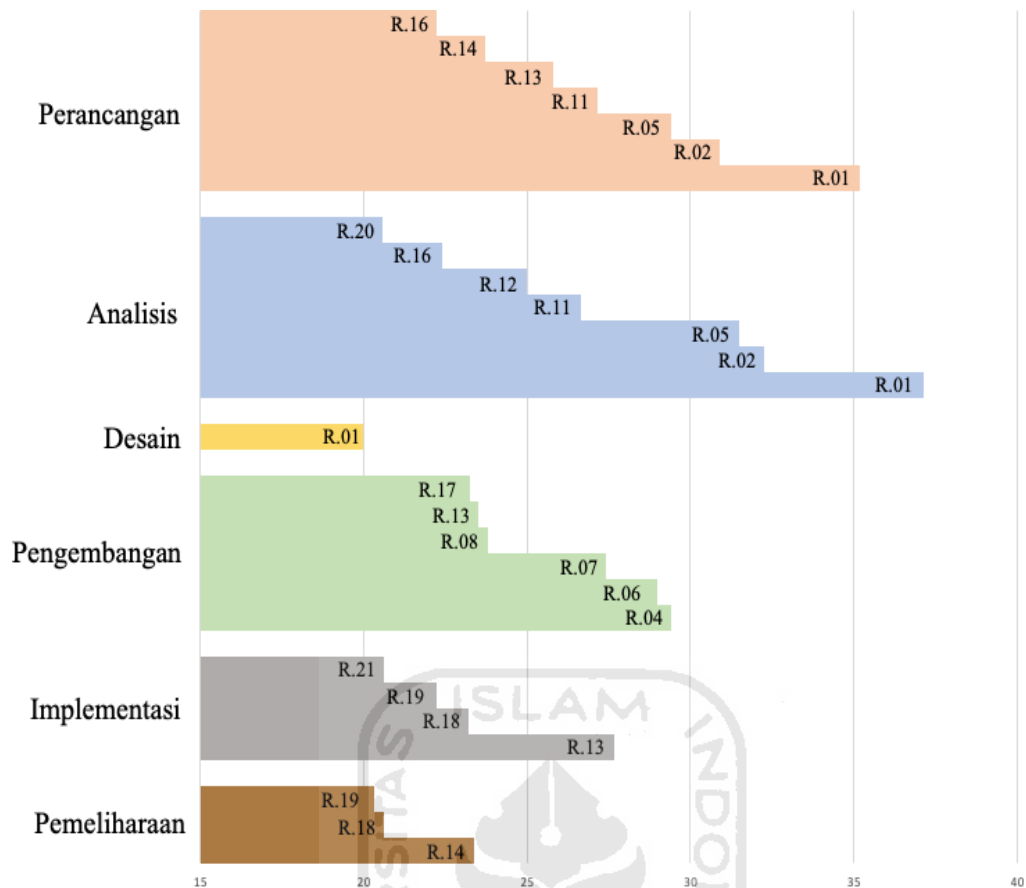
Dilihat dari nilai rata-rata risiko kegagalan seperti pada Tabel 4.8, ditemukan bahwa tiap risiko tidak memiliki rentang nilai yang terlalu jauh, seperti pada risiko perencanaan sistem tidak jelas dan risiko kurangnya analisis kebutuhan/proses bisnis yang hanya memiliki rentang nilai 0,03. Hal tersebut menandakan bahwa risiko-risiko tersebut memiliki dampak yang tidak jauh berbeda terhadap kegagalan proyek pengembangan perangkat lunak. Bahkan terdapat juga risiko dengan nilai rata-rata yang sama yaitu risiko ruang lingkup tidak sesuai dan risiko buruknya pemantauan sistem dengan nilai rata-rata 3,93. Adapun risiko dengan rentang nilai terjauh terdapat pada risiko akuisisi pengguna rendah dan risiko perbedaan budaya dengan rentang nilai mencapai 0,31. Meskipun perbedaan budaya

memiliki rentang nilai terjauh dan nilai rata-rata terendah, namun perbedaan budaya masih menjadi salah satu risiko yang berdampak pada kegagalan proyek pengembangan perangkat lunak.

4.3.2 Risiko pada Tahap Pengembangan

Selama pengembangan proyek berlangsung, penerapan strategi manajemen risiko diperlukan dengan mengaitkan penilaian risiko serta mitigasinya untuk membantu dalam mengidentifikasi risiko beserta mitigasi risikonya. Pada penelitian ini, penulis memvisualisasikan risiko berdasarkan risiko paling dominan yang penulis sajikan pada (Gambar 4.9 Visualisasi Risiko pada Tahap Pengembangan) Visualisasi risiko tersebut didapat dari hasil putaran pertanyaan tahap ke-3 (Gambar 3.1. Tahapan Pengumpulan Data.). Panelis yang terlibat dalam tahap ini diminta untuk memvisualisasikan risiko berdasarkan tahapan pengembangan yang mengacu pada pengalaman para panelis.

Hasil visualisasi tersebut, kemudian penulis rangkum ke dalam bentuk persentase dengan ketentuan jika hasilnya lebih dari 20% maka dianggap memenuhi syarat sebagai risiko yang paling dominan dan penulis sajikan dalam Gambar 4.9. Sedangkan risiko dengan hasil visualisasi dibawah 20% dianggap sebagai risiko yang kurang dominan dan tidak penulis sajikan ke dalam Gambar 4.9 dengan tujuan untuk memudahkan keterbacaan. Berdasarkan hasil visualisasi risiko yang paling mendominasi pada penelitian ini, nantinya dapat dilakukan pencegahan untuk menanggulangi terjadinya risiko pada tiap tahap pengembangannya. Sehingga perusahaan dapat merencanakan perencanaan sistem dengan matang dan sedetail mungkin agar terhindar dari kegagalan proyek akibat risiko-risiko yang muncul pada tiap tahap pengembangannya.



Gambar 4.9 Visualisasi Risiko pada Tahap Pengembangan

Ke 17 risiko tersebut tersebar di enam tahap pengembangan (Gambar 2.1). Namun, terdapat perbedaan tahap pengembangan perangkat lunak untuk risiko yang ditemukan. Seperti pada tahap perencanaan, didapatkan tujuh risiko yaitu R.01 (Kurangannya analisis kebutuhan/proses bisnis), R.02 (Estimasi waktu tidak tepat) R.05 (Perencanaan sistem tidak jelas), R.11 (Manajemen proyek tidak jelas), R.13 (Fasilitas kurang memadai), R.14 (kurangnya kepercayaan pengguna), dan R.16 (Ruang lingkup tidak sesuai). Tahap analisis didapatkan tujuh risiko yaitu R.01 (kurangnya analisis kebutuhan/proses bisnis), R.02 (Estimasi waktu tidak tepat), R.05 (perencanaan sistem tidak jelas), R.11 (Manajemen proyek tidak jelas), R.12 (Kurangannya pemahaman pengguna), R.16 (Ruang lingkup tidak sesuai), dan R.20 (Perbedaan budaya). Temuan menarik terdapat pada tahap desain, karena hanya terdapat satu risiko yaitu R.01 (kurangnya analisis kebutuhan/proses bisnis).

Tahap pengembangan didapatkan enam risiko yaitu R.04 (kurangnya komunikasi antar tim/pengguna dan pengembang), R.06 (kurangnya keahlian/pengetahuan tim pengembang), R.07 (Tim yang tidak profesional), R.08 (Kurangannya tim pengembang), R.13 (Fasilitas kurang memadai), dan R.17 (Kurangannya kerjasama tim). Tahap implementasi didapatkan empat risiko yaitu risiko R.13 (Fasilitas kurang memadai), R.18

(Buruknya pemantauan sistem), R.19 (Akuisisi pengguna rendah), dan R.21 (Teknologi belum teruji). Sementara itu, tahap pemeliharaan hanya didapatkan tiga risiko yaitu R.14 (Kurangannya kepercayaan pengguna), R.18 (Buruknya pemantauan sistem), dan R.19 (Akuisisi pengguna rendah).

Adapun temuan menarik lainnya yang ditemukan dari ke-17 risiko yang tersebar di enam tahap pengembangan, yaitu terdapat satu risiko yang menjadi paling dominan sekaligus paling rendah persentasenya, risiko tersebut adalah R.01 (kurangnya analisis kebutuhan/proses bisnis) dengan persentase mencapai 37,14% pada tahap analisis. Namun, R.01 juga menjadi risiko yang paling rendah persentasenya pada tahap desain dengan hanya memperoleh 20,00%. Serta R.01 juga menjadi satu-satunya risiko yang paling mendominasi di tiga tahapan pengembangan yaitu pada tahap perencanaan, analisis dan desain.

Temuan yang diperoleh pada penelitian ini mempunyai kesamaan maupun perbedaan dengan peneliti sebelumnya (Roy & Dasgupta, 2016). Pada penelitian penulis, risiko yang tertinggi pada tahap perencanaan ternyata tidak termasuk ke dalam risiko yang ada pada peneliti sebelumnya. Pada tahap analisis dan desain dipenelitian ini, R.01 menjadi risiko yang paling dominan. Hal tersebut sejalan dengan hasil penelitian sebelumnya yang menemukan bahwa R.01 atau kurangnya analisis kebutuhan/proses bisnis menjadi salah satu risiko yang dominan pada tahap desain dan analisis bahkan risiko tersebut terdapat juga pada tahap perencanaan di penelitian sebelumnya. Di tahap pengembangan pada penelitian ini, R.08 merupakan risiko yang dominan, begitu juga pada penelitian terdahulu dimana risiko tersebut juga menjadi salah satu risiko yang terdapat pada tahap pengembangan. Sementara itu, pada penelitian penulis maupun penelitian sebelumnya sepakat bahwa R.19 merupakan risiko yang mendominasi tahap implementasi dan juga tahap pemeliharaan.

4.3.3 Risiko dan Tipe Mitigasinya

Risiko yang telah didapat, kemudian ditentukan tipe mitigasi yang sesuai. Dalam menentukan tipe mitigasi, penulis menggunakan strategi manajemen risiko 4T (*Treat, Transfer, Terminate, dan Tolerate*) (Treasury, 2016). Dalam menentukan tipe mitigasi yang tepat, penulis meminta bantuan kepada para panelis untuk membantu dalam penentuan tipe mitigasi untuk tiap risiko yang ditemukan. Hasilnya penulis sajikan dalam Tabel 4.9.

Tabel 4.9. Risiko dan Mitigasinya.

No.	Risiko	Tipe	Kata Kunci	Strategi Mitigasi Risiko
-----	--------	------	------------	--------------------------

	Kegagalan	Mitigasi		
R.01	Kurangnya analisis kebutuhan / proses bisnis	<i>Treat / Transfer</i>	Komunikasi, perjelas proses bisnis.	Adakan komunikasi dengan klien untuk memperjelas proses bisnis dan <i>scope</i> yang belum terpenuhi.
			Analisis, identifikasi.	Lakukan analisis dan identifikasi secara menyeluruh dengan membagi skala prioritas untuk proses bisnis yang akan dibuat.
			Datangkan ahli, pendekatan, komunikasi.	Mendatangkan ahli pada bidang yang sedang dianalisa agar tidak terjadi <i>gap requirement system</i> dan pengembang melakukan pendekatan kepada klien untuk menjalin komunikasi antar keduanya.
R.02	Estimasi waktu tidak tepat	<i>Treat</i>	Jadwal, target, lembur.	Membuat jadwal, target proyek dan menambah waktu kerja (lembur) di fase awal proyek dengan tujuan utama untuk memastikan semua fungsi sistem yang akan dikerjakan sesuai ekspektasi dan rencana.
			Buku panduan, riwayat aktivitas.	Membuat dokumentasi yang berfungsi sebagai riwayat aktivitas / pekerjaan apa saja yang telah dilakukan. Sehingga pekerjaan yang dilakukan dapat dianalisis rekam jejaknya.
R.04	Kurangnya komunikasi antar tim / pengguna dan pengembang	<i>Treat</i>	Evaluasi, komunikasi,	Selalu melakukan evaluasi dan memperbaiki pola komunikasi yang ada dengan mendekatkan tempat duduk antar tim agar komunikasi antar tim dapat dilakukan dengan lebih intens.
			Dokumentasi, komunikasi, diskusi.	Membuat dokumentasi tiap-tiap pekerjaan buka peluang komunikasi yang lebih mudah baik antar tim pengembang maupun pengguna dengan tim dengan cara membuka forum diskusi menyampaikan pendapat.
R.05	Perencanaan sistem tidak jelas	<i>Treat</i>	Tahan proyek, perencanaan ulang sistem.	Tahan proyek dan segera lakukan perencanaan ulang sistem dengan klien dan tim pengembang sampai menemui sebuah kesepakatan terkait perangkat lunak yang akan dikembangkan.
			Evaluasi, proses bisnis, SOP, analisis kebutuhan	Mengevaluasi dilakukan secara berkala terkait perancangan proses bisnis dan SOP serta lakukan analisis kebutuhan perangkat lunak yang akan pengembangan.
R.06	Kurangnya keahlian / pengetahuan tim pengembang	<i>Treat / Transfer</i>	Riset, tambah tim.	Memasukkan riset ke dalam masa pengerjaan proyek atau selama proses <i>pitching</i> atau, menambah tim khusus yang melakukan riset pada awal pekerjaan dimulai.
			Datangkan pelatih, tambah literasi.	Datangkan pelatih untuk memberikan pelatihan kepada tim pengembang dan tambah literasi untuk mengatasi kurangnya keahlian dan pengetahuan sampai dilihat cukup mampu untuk mengerjakan proyek.
			Rekrut ahli	Rekrut ahli atau pengembang senior yang mampu mengerjakan proyek yang akan dikembangkan.
R.07	Tim yang tidak	<i>Treat/ Terminate</i>	Komunikasi titik permasalahan.	Komunikasikan titik permasalahan di internal tim untuk menekankan langkah-langkah

	profesional			perbaiki, target, aturan, kerjasama, untuk meningkatkan profesionalitas.
			Beri kesempatan.	Berikan kesempatan tim untuk meningkatkan profesionalitas dan berikan hadiah atau hukuman.
			Buat ulang tim	Buat ulang tim baru dengan mengganti tim yang tidak profesional dengan yang lebih profesional.
R.08	Kurangnya tim pengembang	Treat	<i>cross function</i> , rekrut, komunikasi.	Tingkatkan <i>cross function</i> setiap anggota tim sebelum melakukan perekrutan, kemudian jadwalkan ulang pekerjaan sesuai kemampuan tim dan komunikasikan pada klien untuk menambah waktu pengerjaan atau menyederhanakan desain antarmuka dan fungsi.
			Evaluasi, tambah anggota	Apabila klien tidak setuju segera lakukan evaluasi pada bagian mana tim dibutuhkan dan tambah anggota tim sesuai dengan posisi dan jumlah tim yang dibutuhkan.
R.11	Manajemen proyek tidak jelas	Treat	Perjelas, jadwal, <i>project charter</i> .	Memperjelas <i>scope</i> yang diinginkan manajemen, dengan membuat jadwal yang harus dilakukan dan <i>project charter</i> yang lengkap dengan melibatkan semua anggota tim. Sehingga tim dapat mengetahui target yang harus dicapai dan pekerjaan menjadi lebih terstruktur.
			<i>Project management framework</i> , perekrutan.	Membuat atau menerapkan <i>project management framework</i> agar lebih jelas dan jika diperlukan melakukan perekrutan <i>project manager</i> dengan pengalaman yang tinggi.
R.12	Kurangnya pemahaman pengguna	Treat	Sosialisasi, <i>transfer knowledge</i> , pendampingan.	Melakukan sosialisasi dan <i>transfer knowledge</i> yang diadakan bertahap dengan cara pendampingan intensif kepada pengguna mengenai perangkat lunak yang akan digunakan.
			Buku panduan, Kerangka acuan.	Membuat buku panduan atau Kerangka acuan terhadap pengguna sebagai panduan umum mengenai perangkat lunak.
R.13	Fasilitas kurang memadai	Treat	Optimalkan, pengadaan	Optimalkan fasilitas yang tersedia atau melakukan pengadaan fasilitas yang dibutuhkan pada saat pengembangan, apabila dimungkinkan minta klien untuk menyediakan fasilitas yang dibutuhkan.
R.14	Kurangnya kepercayaan pengguna	Treat	Komunikasi, komitmen.	Melakukan komunikasi dengan pengguna untuk menyusun komitmen ulang antara kedua pihak dan adakan acara tertentu untuk menjembatani pertemuan tim pengembang dan pengguna.
			Timeline, progress, ekspektasi.	Meyakinkan pengguna terhadap tim pengembang dengan memberikan timeline dan progres pengerjaan tepat waktu dan meyakinkan user bahwa ekspektasi mereka dapat dipenuhi.
			Strategi, <i>analytic tools</i> .	Menyusun strategi digital dengan memanfaatkan <i>analytic tools</i> sebagai <i>review</i> atau <i>report</i> terhadap pengguna.
R.16	Ruang lingkup tidak sesuai	Treat	Komunikasi, perjelas, pelajari.	Lakukan komunikasi secara lanjut dengan Klien memperjelas tujuan awal pembuatan produk dan

				pelajari ruang lingkup yang dibutuhkan
			Analisis, evaluasi, skala prioritas, pelajari.	Segera melakukan analisis ulang dan evaluasi ruang lingkup pekerjaan dengan menentukan skala prioritas agar ruang lingkup bisa lebih sesuai dan lakukan diskusi secara lanjut dengan klien untuk mempelajari ruang lingkup yang dibutuhkan.
R.17	Kurangnyakerjasama tim	Treat	Perkuat peran, identifikasi, evaluasi	Memperkuat peran <i>Project Manager</i> , <i>Product Owner</i> , dan <i>Scrum Master</i> untuk melakukan manajemen, identifikasi serta evaluasi kondisi tim.
			<i>Brainstorming</i> , kolaborasi, perkecil anggota	Berikan arahan untuk meningkatkan kerjasama dengan cara memberikan <i>brainstorming</i> , kolaborasi tim secara rutin, dan jika diperlukan memperkecil anggota tim.
R.18	Buruknyapemantauan sistem	Treat	Penjadwalan, dokumentasi, evaluasi, verifikasi.	Setiap divisi perlu melakukan penjadwalan pemantauan sistem, kemudian dokumentasikan serinci mungkin dan melaporkannya untuk dilakukan evaluasi serta verifikasi.
			keterlibatan pengguna.	Meminta keterlibatan pengguna untuk ikut serta melakukan pemantauan sistem.
R.19	Akuisisi pengguna rendah	Treat	Promosi, komunikasi, persuasi.	Melakukan promosi dengan cara memperbanyak komunikasi dan melakukan pendekatan persuasif kepada pengguna
			Segmentasi, membagi pengetahuan, pendampingan.	Melakukan segmentasi, dan membagi pengetahuan secara berkala kepada pengguna dengan melakukan pendampingan secara intensif.
R.20	Perbedaan budaya	Treat	Akomodir, sosialisasi.	Mengakomodir perbedaan budaya dengan cara mengadakan sosialisasi, <i>training</i> , <i>event gathering</i> , <i>tech talk</i> untuk menjalin kerjasama antar tim.
			Beri toleransi, komunikasi.	Mencoba memberikan toleransi, menghormati dan melakukan komunikasi yang baik antar tim agar proyek dapat diterima oleh semua budaya.
R.21	Teknologi belum teruji	Treat/Terminate	Penelitian, pengujian, simulasi.	Lakukan penelitian dan pengujian teknologi dengan mengoptimalkan peran <i>Research and Development (R&D)</i> untuk mensimulasikan perangkat lunak di lapangan (CRUD data sederhana) agar dapat mengetahui keefektifan teknologi yang digunakan.
			Batasi, tolak	Batasi/tolak penggunaan teknologi baru atau hindari penggunaan teknologi yang belum teruji.

Tabel 4.10. Keterangan Tipe Mitigas

No.	Risiko Kegagalan	Tipe Mitigasi	Keterangan
R.01	Kurangnyaanalisis kebutuhan / proses bisnis	Treat	Karena analisis kebutuhan/proses bisnis terdapat pada perusahaan yang berkaitan, jadi masih bisa menghubungi/mengkomunikasikan kepada klien terkait kebutuhan proses bisnis.
		Transfer	Karena perusahaan perlu mendatangkan ahli pada bidang yang sedang dianalisa agar tidak terjadi <i>gap requirement system</i> .

R.02	Estimasi waktu tidak tepat	<i>Treat</i>	Karena saat ini sudah tersedia berbagai macam <i>tools</i> seperti Jira, Trello, dan Confluence yang berguna untuk membantu perhitungan atau estimasi pengerjaan proyek.
R.04	Kurangnya komunikasi antar tim / pengguna dan pengembang	<i>Treat</i>	Karena risiko tersebut dapat diatasi dengan melakukan evaluasi dan memperbanyak komunikasi terkait masalah serta solusinya.
R.05	Perencanaan sistem tidak jelas	<i>Treat</i>	Karena pada saat perancangan sistem tim pengembang (UX) dapat dimintai untuk membuat <i>prototype</i> sistem yang akan dikembangkan untuk meminimalisir terjadinya kesalahan perancangan.
R.06	Kurangnya keahlian dan pengetahuan tim pengembang	<i>Treat</i>	Karena tim pengembang dapat diberikan pelatihan peningkatan <i>skill</i> dengan cara mengikuti <i>workshop</i> atau <i>techtalk</i> yang diadakan oleh perusahaan itu sendiri.
		<i>Transfer</i>	Karena perusahaan perlu mendatangkan pelatih untuk memberikan pelatihan kepada tim pengembang dan pengetahuan sampai dilihat cukup mampu untuk mengerjakan proyek.
R.07	Tim yang tidak profesional	<i>Treat</i>	Karena dapat mencari titik permasalahan di internal tim untuk menekankan langkah-langkah perbaikan, target, aturan, kerjasama, untuk meningkatkan profesionalitas.
		<i>Terminate</i>	Karena tim yang tidak profesional tidak memenuhi aturan perusahaan bahkan dapat memberikan kerugian material kepada perusahaan. Contohnya seperti tim pengembangan diberikan upah namun tidak memberikan <i>feedback</i> yang diinginkan perusahaan.
R.08	Kurangnya tim pengembang	<i>Treat</i>	Karena perusahaan dapat melakukan perekrutan karyawan untuk melengkapi jumlah tim yang dibutuhkan.
R.11	Manajemen proyek tidak jelas	<i>Treat</i>	Karena perusahaan dapat melakukan evaluasi terkait manajemen proyek.
R.12	Kurangnya pemahaman pengguna	<i>Treat</i>	Karena tim pengembang dapat membuat <i>manual book</i> penggunaan aplikasi sehingga pengguna dapat lebih memahami tanpa adanya pendampingan langsung secara terus menerus.
R.13	Fasilitas kurang memadai	<i>Treat</i>	Karena kurangnya fasilitas masih dapat ditangani dengan mengoptimalkan fasilitas yang tersedia atau perusahaan dapat melakukan pengadaan terkait fasilitas-fasilitas yang dibutuhkan.
R.14	Kurangnya kepercayaan pengguna	<i>Treat</i>	Karena perusahaan dapat melakukan evaluasi produk dan menemukan strategi untuk memperoleh kepercayaan pengguna seperti melakukan komunikasi secara intens dan menunjukkan progres pengembangan perangkat lunak di setiap harinya.
R.16	Ruang lingkup tidak sesuai	<i>Treat</i>	Karena risiko tersebut dapat ditangani dengan cara peninjauan ulang Ruang lingkup tidak sesuai yang dapat dilakukan melalui komunikasi.
R.17	Kurangnya kerjasama tim	<i>Treat</i>	Karena perusahaan dapat melakukan evaluasi kinerja tim dan membangun kerjasama anggota tim dengan mengadakan <i>gathering</i> atau <i>meeting</i> singkat untuk meningkatkan komunikasi antar tim.
R.18	Buruknya pemantauan sistem	<i>Treat</i>	Karena perusahaan dapat meningkatkan pemantauan sistem dengan memanfaatkan <i>analytic tools</i> sehingga hasil pemantauan sistem dapat dianalisis
R.19	Akuisisi pengguna rendah	<i>Treat</i>	Karena perusahaan dapat mengakuisisi pengguna dengan memanfaatkan media promosi dan mengevaluasi penggunaan aplikasi melalui <i>google analytic</i>
R.20	Perbedaan budaya	<i>Treat</i>	Karena perusahaan dapat menerapkan aturan umum yang dapat diterima oleh semua budaya. Selain itu, perusahaan juga dapat melakukan kegiatan <i>universal</i> untuk memperkuat keberagaman

			budaya dan menyatukan <i>mindset</i> antar tim.
R.21	Teknologi belum teruji	<i>Treat</i>	Karena tim pengembang dapat melakukan penelitian dan pengujian teknologi dengan mengoptimalkan peran <i>Research and Development</i> (R&D) untuk mensimulasikan perangkat lunak di lapangan (CRUD data sederhana) agar dapat mengetahui keefektifan teknologi yang digunakan.
		<i>Terminate</i>	Karena melakukan riset teknologi yang belum teruji memerlukan waktu yang lebih lama dan biaya yang lebih besar



BAB 5

Kesimpulan dan Implikasi

5.1 Kesimpulan

Dengan menggunakan metode Delphi dan memanfaatkan pengalaman lebih dari 40 panelis yang terlibat dalam proyek pengembangan perangkat lunak di Indonesia untuk mengidentifikasi risiko yang dianggap paling berdampak, perbedaan risiko untuk setiap tahap pengembangan dan strategi mitigasi risiko diperoleh beberapa temuan menarik yang disajikan ke dalam 3 point pembahasan yaitu:

A. Risiko Kegagalan

Dari 22 risiko yang ditemukan berdasarkan tahap *brainstorming*, diperoleh 17 risiko yang mencapai nilai konsensus dan 5 risiko yang tidak mencapai konsensus. Hal ini menarik karena ke-5 risiko tersebut memiliki jumlah tanggapan dan nilai *mean* yang tinggi. Serta temuan lain menunjukkan bahwa ke-17 yang mencapai konsensus memiliki rentan nilai yang tidak terlalu jauh. Hal itu menandakan bahwa risiko-risiko tersebut memiliki dampak yang tidak jauh berbeda terhadap kegagalan proyek pengembangan perangkat lunak. Adapun rentang nilai terjauh terdapat pada risiko akuisisi pengguna rendah dan risiko perbedaan budaya dengan rentang nilai mencapai 0,31. Meskipun perbedaan budaya memiliki nilai rata-rata terendah dan jarang disebutkan, namun perbedaan budaya menjadi salah satu risiko yang berdampak pada kegagalan proyek pengembangan perangkat lunak.

B. Risiko Pada Tahap pengembangan

Berdasarkan ke-17 risiko yang telah mencapai konsensus menunjukkan bahwa R.01 (Kurangannya analisis kebutuhan/proses bisnis) menjadi satu-satunya risiko dengan persentase tertinggi dari enam tahap pengembangan persentase dengan 37,14% pada tahap analisis serta R.01 juga menjadi satu-satunya risiko dominan yang ada pada tahap pengembangan desain dengan nilai dominan 20% atau nilai dominan yang terendah. R.01 juga merupakan risiko dominan di tiga tahap pengembangan yaitu perencanaan analisis dan desain.

C. Mitigasi risiko

Berdasarkan strategi mitigasi yang diperoleh dari tanggapan panelis sebagian besar risiko kegagalan pengembangan perangkat lunak dapat ditangani dengan melakukan evaluasi dan

memperbanyak komunikasi baik komunikasi antara anggota tim pengembang maupun komunikasi dengan pengguna atau klien.

5.2 Implikasi Konseptual

Berdasarkan penelitian yang telah dilakukan oleh peneliti terdahulu terkait risiko kegagalan proyek pengembangan perangkat lunak (Tabel 2.1), pada penelitian ini menemukan beberapa risiko kegagalan baru seperti, Kurangnya analisis kebutuhan/proses bisnis, Tim yang tidak profesional, Kurangnya pemahaman pengguna, Fasilitas kurang memadai, Kurangnya kepercayaan pengguna, Buruknya pemantauan sistem, Akuisisi pengguna rendah. Penulis juga menemukan risiko yang perbedaan risiko yang dominan pada setiap tahap pengembangan serta strategi mitigasi risiko untuk risiko yang penulis temukan.

5.3 Implikasi Praktikal

1. Perlunya mempertimbangkan bahwa terdapat perbedaan risiko yang muncul pada setiap tahapan pengembangan proyek perangkat lunak.
2. Ternyata terdapat perbedaan risiko yang yang dianggap paling berdampak pada proyek pengembangan perangkat lunak (Tabel 4.8) dengan risiko yang menjadi dominan pada setiap tahap pengembangan perangkat lunak (Gambar 2.1)
3. Terdapat berbagai macam strategi mitigasi risiko untuk menangani risiko yang sama, tergantung pada kebijakan dengan strategi mitigasi yang seperti apa risiko tersebut dapat ditangani.

5.4 Saran

Masukkan untuk penelitian selanjutnya:

1. Lebih banyak mengidentifikasi panelis yang berada di Indonesia dan memperkaya peran panelis yang terlibat dalam proyek pengembangan perangkat lunak seperti *lead architect, customer representative, tester, corporate quality assurance liaison, subsystem teams lead, transition/team lead, release manager, toolsmith, technical writers*, dll. Untuk memungkinkan diperolehnya risiko-risiko kegagalan yang baru.
2. Perlu menambahkan panelis dari berbagai wilayah di Indonesia dengan suku dan budaya yang berbeda untuk membuktikan bahwa risiko perbedaan budaya mempunyai pengaruh yang kuat terhadap kekegagalan proyek pengembangan perangkat lunak di Indonesia.

REFERENSI

- Ahmed, A., Kayis, B., & Amornsawadwatana, S. (2007). A review of techniques for risk management in projects. *Benchmarking*, 14(1), 22–36.
<https://doi.org/10.1108/14635770710730919>
- Ahmed, L., Quddus, N., Kannan, P., Peres, S. C., & Mannan, M. S. (2020). Development of a procedure writers' guide framework: Integrating the procedure life cycle and reflecting on current industry practices. *International Journal of Industrial Ergonomics*, 76(November 2019), 102930.
<https://doi.org/10.1016/j.ergon.2020.102930>
- Akgün, A. E. (2020). Team wisdom in software development projects and its impact on project performance. *International Journal of Information Management*, 50(June 2019), 228–243. <https://doi.org/10.1016/j.ijinfomgt.2019.05.019>
- Ameyaw, E. E., Hu, Y., Shan, M., Chan, A. P. C., & Le, Y. (2016). Application of Delphi method in construction engineering and management research: A quantitative perspective. *Journal of Civil Engineering and Management*, 22(8), 991–1000.
<https://doi.org/10.3846/13923730.2014.945953>
- Andrei, B.-A., Casu-Pop, A., Gheorghe, S.-C., & Boianiu, C.-A. (2019). A Study on Using Waterfall and Agile Methods in Software Project Management. *Journal of Information Systems & Operations Management*, 125–135. <http://0-search.proquest.com.ditlib.dit.ie.tudublin.idm.oclc.org/docview/2237828314?accountid=10594>
- Andreswari, R., Ambarsari, N., Syahrina, A., Puspitasari, W., Novianti, A., & Darmawan, I. (2020). Design of e-Marketplace for Village-owned Small, Micro and Medium Enterprise using Rapid Application Development. *International Journal of Innovation in Enterprise System*, 4(01), 35–43. <https://doi.org/10.25124/ijies.v4i01.46>
- Antony, J., & Gupta, S. (2019). Top ten reasons for process improvement project failures. *International Journal of Lean Six Sigma*, 10(1), 367–374.
<https://doi.org/10.1108/IJLSS-11-2017-0130>
- Balki, M., Hoppe, D., Monks, D., Cooke, M. E., Sharples, L., & Windrim, R. (2017). Multidisciplinary Delphi Development of a Scale to Evaluate Team Function in Obstetric Emergencies: The PETRA Scale. *Journal of Obstetrics and Gynaecology Canada*, 39(6), 434-442.e2. <https://doi.org/10.1016/j.jogc.2017.01.030>
- Barnes, S. J., & Mattsson, J. (2016). Understanding current and future issues in

- collaborative consumption: A four-stage Delphi study. *Technological Forecasting and Social Change*, 104, 200–211. <https://doi.org/10.1016/j.techfore.2016.01.006>
- Bassil, Y. (2012). A Simulation Model for the Spiral Software Development Life Cycle. *International Journal of Engineering & Technology (IJET)*, 02(05). <https://doi.org/10.15680/ijircce.2015.0305013>
- Bhavsar, K., Shah, V., & Gopalan, S. (2020). Scrumbanfall: An Agile Integration of Scrum and Kanban with Waterfall in Software Engineering. *International Journal of Innovative Technology and Exploring Engineering*, 9(4), 2075–2084. <https://doi.org/10.35940/ijitee.d1437.029420>
- Bishop, D. V. M., Snowling, M. J., Thompson, P. A., Greenhalgh, T., Adams, C., Archibald, L., Baird, G., Bauer, A., Bellair, J., Boyle, C., Brownlie, E., Carter, G., Clark, B., Clegg, J., Cohen, N., Conti-Ramsden, G., Dockrell, J., Dunn, J., Ebbels, S., ... Whitehouse, A. (2017). Phase 2 of CATALISE: a multinational and multidisciplinary Delphi consensus study of problems with language development: Terminology. *Journal of Child Psychology and Psychiatry and Allied Disciplines*, 58(10), 1068–1080. <https://doi.org/10.1111/jcpp.12721>
- Broad, J. (2013). *Risk management framework: a lab-based approach to securing information systems*. Syngress.
- Cai, Y., Wu, Y., Zhou, J., Liu, M., & Zhang, Q. (2020). Quantitative software reliability assessment methodology based on Bayesian belief networks and statistical testing for safety-critical software. *Annals of Nuclear Energy*, 145, 107593. <https://doi.org/10.1016/j.anucene.2020.107593>
- Chaouch, S., Mejri, A., & Ghannouchi, S. A. (2019). A framework for risk management in Scrum development process. *Procedia Computer Science*, 164, 187–192. <https://doi.org/10.1016/j.procs.2019.12.171>
- Chien, K. F., Wu, Z. H., & Huang, S. C. (2014). Identifying and assessing critical risk factors for BIM projects: Empirical study. *Automation in Construction*, 45, 1–15. <https://doi.org/10.1016/j.autcon.2014.04.012>
- Chrismanto, A. R., Delima, R., Santoso, H. B., Wibowo, A., & Kristiawan, R. A. (2019). Developing agriculture land mapping using Rapid Application Development (RAD): A case study from Indonesia. *International Journal of Advanced Computer Science and Applications*, 10(10), 232–241. <https://doi.org/10.14569/ijacsa.2019.0101033>
- Cid, J., De La Calle, J. L., López, E., Del Pozo, C., Perucho, A., Acedo, M. S., Bedmar, D., Benito, J., De Andrés, J., Díaz, S., García, J. A., Gómez-Caro, L., Gracia, A.,

- Hernández, J. M., Insausti, J., Madariaga, M., Moñino, P., Ruiz, M., Uriarte, E., & Vidal, A. (2015). A modified delphi survey on the signs and symptoms of low back pain: Indicators for an interventional management approach. *Pain Practice, 15*(1), 12–21. <https://doi.org/10.1111/papr.12135>
- Cooper, R. G., & Sommer, A. F. (2016). Agile-Stage-Gate: New idea-to-launch method for manufactured new products is faster, more responsive. *Industrial Marketing Management, 59*, 167–180. <https://doi.org/10.1016/j.indmarman.2016.10.006>
- Dalkey, N., & Helmer, O. (1962). *An Experimental Application of The Delphi Method to The Use of Experts*.
- Daniel, V. T., Alavi, K., Davids, J. S., Sturrock, P. R., Harnsberger, C. R., Steele, S. R., & Maykel, J. A. (2020). The utility of the delphi method in defining anastomotic leak following colorectal surgery. *American Journal of Surgery, 219*(1), 75–79. <https://doi.org/10.1016/j.amjsurg.2019.05.011>
- De Bakker, K., Boonstra, A., & Wortmann, H. (2011). Risk management affecting IS/IT Project success through communicative action. *Project Management Journal, 42*(3), 75–90. <https://doi.org/10.1002/pmj.20242>
- de Vasconcelos, J. B., Kimble, C., Carreteiro, P., & Rocha, Á. (2017). The application of knowledge management to software evolution. *International Journal of Information Management, 37*(1), 1499–1506. <https://doi.org/10.1016/j.ijinfomgt.2016.05.005>
- Denost, Q., Bousser, V., Morin-Porchet, C., Vincent, C., Pinon, E., Collin, F., Martin, A., Colombani, F., Digue, L., Ravaud, A., Harji, D. P., & Saillour-Glénisson, F. (2019). The development of a regional referral pathway for locally recurrent rectal cancer: A Delphi consensus study. *European Journal of Surgical Oncology, 46*, 470–475. <https://doi.org/10.1016/j.ejso.2019.12.001>
- Dewantara, K. H. (2016). *Identifikasi, Penilaian, Dan Mitigasi Risiko Keamanan Informasi Berdasarkan Standar Iso 27001: 2005 Dan Iso 27002: 2013 Menggunakan* <http://repository.its.ac.id/1315/>
- Dolan, C., Glynn, R., & Lawlor, B. (2020). A systematic review and Delphi study to ascertain common risk factors for T2DM and dementia and brain-related complications of diabetes. *Canadian Journal of Diabetes*. <https://doi.org/10.1016/j.jcjd.2020.01.004>
- Dwivedi, Y. K., Wastell, D., Laumer, S., Henriksen, H. Z., Myers, M. D., Bunker, D., Elbanna, A., Ravishankar, M. N., & Srivastava, S. C. (2014). Research on information systems failures and successes: Status update and future directions. *Information*

- Systems Frontiers*, 17(1), 143–157. <https://doi.org/10.1007/s10796-014-9500-y>
- Fang, C., & Marle, F. (2012). A simulation-based risk network model for decision support in project risk management. *Decision Support Systems*, 52(3), 635–644. <https://doi.org/10.1016/j.dss.2011.10.021>
- Fernández-Ávila, D. G., Rojas, M. X., & Rosselli, D. (2020). The Delphi method in rheumatology research: are we doing it right? *Revista Colombiana de Reumatología (English Edition)*, x x. <https://doi.org/10.1016/j.rcreue.2019.04.007>
- Flostrand, A., Pitt, L., & Bridson, S. (2020). The Delphi technique in forecasting– A 42-year bibliographic analysis (1975–2017). *Technological Forecasting and Social Change*, 150(January 2018), 119773. <https://doi.org/10.1016/j.techfore.2019.119773>
- Ghobadi, S. (2015). What drives knowledge sharing in software development teams: A literature review and classification framework. *Information and Management*, 52(1), 82–97. <https://doi.org/10.1016/j.im.2014.10.008>
- Giannarou, L., & Zervas, E. (2014). Using Delphi technique to build consensus in practice. *Journal of Business Science and Applied Management*, 9(2).
- Gunadi, Ningtyas, H. H., Simanjaya, S., Febrianti, M., Ryantono, F., & Makhmudi, A. (2020). Comparison of pre-operative Hirschsprung-associated enterocolitis using classical criteria and Delphi method: A diagnostic study. *Annals of Medicine and Surgery*, 51(1), 37–40. <https://doi.org/10.1016/j.amsu.2020.01.007>
- Gupta, S. K., Gunasekaran, A., Antony, J., Gupta, S., Bag, S., & Roubaud, D. (2019). Systematic literature review of project failures: Current trends and scope for future research. *Computers and Industrial Engineering*, 127, 274–285. <https://doi.org/10.1016/j.cie.2018.12.002>
- Hanafi, M. M. (2014). *Risiko, Proses Manajemen Risiko, dan Enterprise Risk Management*. Manajemen Risiko.
- Hopkin, P. (2012). *Fundamentals of Risk Management: Understanding Evaluating and Implementing Effective Risk Management (2nd Edition)*. <http://site.ebrary.com/lib/unisa1/docDetail.action?docID=10555723>
- Hu, Y., Zhang, X., Ngai, E. W. T., Cai, R., & Liu, M. (2013). Software project risk analysis using Bayesian networks with causality constraints. *Decision Support Systems*, 56(1), 439–449. <https://doi.org/10.1016/j.dss.2012.11.001>
- Kaur, R., & Sengupta, J. (2013). *Software Process Models and Analysis on Failure of Software Development Projects*. 2(2), 1–4. <http://arxiv.org/abs/1306.1068>
- Khalil, A. A., Reza, A., Junaedi, P. A., & Kanigoro, B. (2015). Data Visualization

- Application for Analyzing Public Company Financial Statement. *Procedia Computer Science*, 59(Iccsci), 45–53. <https://doi.org/10.1016/j.procs.2015.07.336>
- Knox, G. M., Snodgrass, S. J., Southgate, E., & Rivett, D. A. (2019). A Delphi study to establish consensus on an educational package of musculoskeletal clinical prediction rules for physiotherapy clinical educators. *Musculoskeletal Science and Practice*, 44(March), 102053. <https://doi.org/10.1016/j.msksp.2019.102053>
- Koppes, D. M., Triepels, C. P. R., Schepens-Franke, A. N., Kruitwagen, R. F. P. M., Van Gorp, T., Scheele, F., & Notten, K. J. B. (2020). What do we need to know about anatomy in gynaecology: A Delphi consensus study. *European Journal of Obstetrics and Gynecology and Reproductive Biology*, 245, 56–63. <https://doi.org/10.1016/j.ejogrb.2019.11.015>
- Kussunga, F., & Ribeiro, P. (2019). Proposal of a Visual Environment to Support Scrum. *Procedia Computer Science*, 164, 491–497. <https://doi.org/10.1016/j.procs.2019.12.211>
- Lee, S. H., Lee, S. J., Koo, S. R., Varuttamaseni, A., Yue, M., Li, M., Cho, J., & Kang, H. G. (2020). Optimization of software development life cycle quality for NPP safety software based on a risk-cost model. *Annals of Nuclear Energy*, 135, 106961. <https://doi.org/10.1016/j.anucene.2019.106961>
- Lehtinen, T. O. A., Mäntylä, M. V., Vanhanen, J., Itkonen, J., & Lassenius, C. (2014). Perceived causes of software project failures - An analysis of their relationships. *Information and Software Technology*, 56(6), 623–643. <https://doi.org/10.1016/j.infsof.2014.01.015>
- Loiro, C., Castro, H., Ávila, P., Cruz-Cunha, M. M., Putnik, G. D., & Ferreira, L. (2019). Agile Project Management: A Communicational Workflow Proposal. *Procedia Computer Science*, 164, 485–490. <https://doi.org/10.1016/j.procs.2019.12.210>
- Lyu, F., Zheng, C., Wang, H., Nie, C., Ma, X., Xia, X., Zhu, W., Jin, X., Hu, Y., Sun, Y., Zhu, Y., Kuwabara, S., Cortese, R., Maqbool Hassan, K., Takai, K., Paredes, I., Webere, R., Turk, M., Kimura, J., & Jiang, J. (2020). Establishment of a clinician-led guideline on the diagnosis and treatment of Hirayama disease using a modified Delphi technique. *Clinical Neurophysiology*, 131(6), 1311–1319. <https://doi.org/10.1016/j.clinph.2020.02.022>
- Maher, T. M., Whyte, M. K. B., Hoyles, R. K., Parfrey, H., Ochiai, Y., Mathieson, N., Turnbull, A., Williamson, N., & Bennett, B. M. (2015). Development of a Consensus Statement for the Definition, Diagnosis, and Treatment of Acute Exacerbations of

- Idiopathic Pulmonary Fibrosis Using the Delphi Technique. *Advances in Therapy*, 32(10), 929–943. <https://doi.org/10.1007/s12325-015-0249-6>
- Meher, S., Cuthbert, A., Kirkham, J. J., Williamson, P., Abalos, E., Aflaifel, N., Bhutta, Z. A., Bishop, A., Blum, J., Collins, P., Devane, D., Ducloy-Bouthors, A. S., Fawole, B., Gülmezoglu, A. M., Gutteridge, K., Gyte, G., Homer, C. S. E., Mallalah, S., Smith, J. M., ... Alfirevic, Z. (2019). Core outcome sets for prevention and treatment of postpartum haemorrhage: an international Delphi consensus study. *BJOG: An International Journal of Obstetrics and Gynaecology*, 126(1), 83–93. <https://doi.org/10.1111/1471-0528.15335>
- Montequin, V. R., Cousillas, S. M., Alvarez, V., & Villanueva, J. (2016). Success Factors and Failure Causes in Projects: Analysis of Cluster Patterns Using Self-organizing Maps. *Procedia Computer Science*, 100, 440–448. <https://doi.org/10.1016/j.procs.2016.09.180>
- Mutsaers, B., Butow, P., Dinkel, A., Humphris, G., Maheu, C., Ozakinci, G., Prins, J., Sharpe, L., Smith, A., “Ben,” Thewes, B., & Lebel, S. (2020). Identifying the key characteristics of clinical fear of cancer recurrence: An international Delphi study. *Psycho-Oncology*, 29(2), 430–436. <https://doi.org/10.1002/pon.5283>
- Nakatsu, R. T., & Iacovou, C. L. (2009). A comparative study of important risk factors involved in offshore and domestic outsourcing of software development projects: A two-panel Delphi study. *Information and Management*, 46(1), 57–68. <https://doi.org/10.1016/j.im.2008.11.005>
- Nidagundi, P., & Novickis, L. (2016). Introducing Lean Canvas Model Adaptation in the Scrum Software Testing. *Procedia Computer Science*, 104(December 2016), 97–103. <https://doi.org/10.1016/j.procs.2017.01.078>
- Okoli, C., & Pawlowski, S. D. (2004). The Delphi method as a research tool: An example, design considerations and applications. *Information and Management*, 42(1), 15–29. <https://doi.org/10.1016/j.im.2003.11.002>
- Raharjo, T., Purwandari, B., Satria, R., & Solichah, I. (2018). Critical success factors for project management office: An insight from Indonesia. In *2018 Third International Conference on Informatics and Computing (ICIC)*, (pp. 1-6).
- Rahman, M. A., Razali, R., & Ismail, F. F. (2019). Risk factors for software requirements change implementation. *International Journal of Advanced Computer Science and Applications*, 10(3), 133–139. <https://doi.org/10.14569/IJACSA.2019.0100316>
- Raka, C., & Liangrokpart, J. (2015). www.econstor.eu.

- Riaz, M. T., Shah Jahan, M., Arif, K. S., & Haider Butt, W. (2019). Risk Assessment on Software Development using Fishbone Analysis. *Proceedings of 2019 International Conference on Data and Software Engineering, ICoDSE 2019, May*.
<https://doi.org/10.1109/ICoDSE48700.2019.9092727>
- Roy, B., & Dasgupta, R. (2016). A Study on Software Risk Management Strategies and Mapping with SDLC. *Advances in Intelligent Systems and Computing*, 396(November). <https://doi.org/10.1007/978-81-322-2653-6>
- Sangwan, T., & Liangro, J. (2015). Risk Identification for Outbound Road Freight Transportation Service. *Proceedings of the Hamburg International Conference of Logistics (HICL), August*, 391–417.
- Simarmata, J. (2010). *Rekayasa Perangkat Lunak*. ANDI.
- Sommerville, I. (2003). *Software Engineering*. ERLANGGA.
- Song, H., & Jiang, J. (2016). Risks Identification in Embedded Software Development: Evidence from MVBC Project Survey. *Procedia Computer Science*, 91, 798–806.
<https://doi.org/10.1016/j.procs.2016.07.082>
- Sriwindono, H., & Yahya, S. (2012). Toward Modeling the Effects of Cultural Dimension on ICT Acceptance in Indonesia. *Procedia - Social and Behavioral Sciences*, 65(ICIBSoS), 833–838. <https://doi.org/10.1016/j.sbspro.2012.11.207>
- Taherdoost, H., & Keshavarzsaleh, A. (2016). Critical Factors that Lead to Projects' Success/Failure in Global Marketplace. *Procedia Technology*, 22, 1066–1075.
<https://doi.org/10.1016/j.protcy.2016.01.151>
- Tavares, B. G., da Silva, C. E. S., & de Souza, A. D. (2019). Risk management analysis in Scrum software projects. *International Transactions in Operational Research*, 26(5), 1884–1905. <https://doi.org/10.1111/itor.12401>
- The Standish Group International. (2015). CHAOS Report 2015. *The Standish Group International, Inc.*, 13.
https://www.standishgroup.com/sample_research_files/CHAOSReport2015-Final.pdf
- Treasury, H. M. (2016). Risk management guidance. *Higher Education Funding Council for England*, 1–7.
- Verner, J. M., Brereton, O. P., Kitchenham, B. A., Turner, M., & Niazi, M. (2014). Risks and risk mitigation in global software development: A tertiary study. *Information and Software Technology*, 56(1), 54–78. <https://doi.org/10.1016/j.infsof.2013.06.005>
- Wan Husin, W. S., Yahya, Y., Mohd Azmi, N. F., Amir Sjarif, N. N., Chuprat, S., & Azmi, A. (2019). Risk management framework for distributed software team: A case study

- of telecommunication company. *Procedia Computer Science*, 161, 178–186.
<https://doi.org/10.1016/j.procs.2019.11.113>
- Westland, J. (2007). The Project Management Lifecycle. *Project Management ROI*, 15–30.
<https://doi.org/10.1002/9781118122587.ch2>
- Whitney, K. M., & Daniels, C. B. (2013). The root cause of failure in complex IT projects: Complexity itself. *Procedia Computer Science*, 20, 325–330.
<https://doi.org/10.1016/j.procs.2013.09.280>
- Widiasih, W., Karningsih, P. D., & Ciptomulyono, U. (2015). Development of Integrated Model for Managing Risk in Lean Manufacturing Implementation: A Case Study in an Indonesian Manufacturing Company. *Procedia Manufacturing*, 4(Lm), 282–290.
<https://doi.org/10.1016/j.promfg.2015.11.042>
- Willumsen, P., Oehmen, J., Stingl, V., & Geraldi, J. (2019). Value creation through project risk management. *International Journal of Project Management*, 37(5), 731–749.
<https://doi.org/10.1016/j.ijproman.2019.01.007>
- Wong, W. E., Li, X., & Laplante, P. A. (2017). Be more familiar with our enemies and pave the way forward: A review of the roles bugs played in software failures. *Journal of Systems and Software*, 133, 68–94. <https://doi.org/10.1016/j.jss.2017.06.069>

