



Tersedia online di [www.journal.unipdu.ac.id](http://www.journal.unipdu.ac.id)  
Unipdu  
Terakreditasi Sinta S5

Halaman jurnal di [www.journal.unipdu.ac.id/index.php/teknologi](http://www.journal.unipdu.ac.id/index.php/teknologi)



## Risiko pengembangan perangkat lunak di indonesia: Temuan studi delphi

### *Software development risk in indonesia: Findings from a delphi study*

Setio Ardy Nuswantoro <sup>a</sup>, Fathul Wahid <sup>b</sup>, Hanson Prihantoro Putro <sup>c</sup>

<sup>a,b,c</sup> Teknik Informatika, Universitas Islam Indonesia, Yogyakarta, Indonesia

email: <sup>a</sup> [17917222@students.uii.ac.id](mailto:17917222@students.uii.ac.id), <sup>b</sup> [fathul.wahid@uui.ac.id](mailto:fathul.wahid@uui.ac.id), <sup>c</sup> [hanson@uui.ac.id](mailto:hanson@uui.ac.id)

#### INFO ARTIKEL

##### Sejarah artikel:

Menerima 22 Juli 2020  
Revisi 6 Agustus 2020  
Diterima 14 Agustus 2020  
Online 16 Agustus 2020

##### Kata kunci:

manajemen risiko TI  
metode Delphi  
pengembangan perangkat lunak  
risiko kegagalan

##### Keywords:

Delphi method  
IT risk management  
risk of failure  
software development

##### Style APA dalam menyitasi artikel ini:

Nuswantoro, S. A., Wahid, F., & Putro, H. P. (2020). Risiko pengembangan perangkat lunak di indonesia: Temuan studi delphi. *Teknologi: Jurnal Ilmiah Sistem Informasi*, 10(1), 43-54.

#### ABSTRAK

Proyek pengembangan perangkat lunak telah berhasil digunakan di berbagai bidang, tetapi persentase kegagalan masih cukup tinggi. Di Indonesia tingkat keberhasilan proyek hanya mencapai 50%. Penelitian ini bertujuan mengidentifikasi risiko pengembangan perangkat lunak dengan konteks Indonesia. Penelitian menjawab dua pertanyaan: (a) Apa risiko dalam proyek pengembangan perangkat lunak, dan (b) Apakah terdapat perbedaan tahap pengembangan perangkat lunak untuk risiko yang ditemukan. Metode Delphi digunakan untuk memperoleh pendapat ahli dengan serangkaian kuesioner berulang yang bertujuan untuk mencapai konsensus. Penelitian ini menggunakan metode Delphi dengan melibatkan 41 panelis dengan beragam latar belakang peran dalam proyek pengembangan perangkat lunak, penelitian ini menemukan 17 risiko berdasar konsensus panelis. Pemetaan risiko untuk setiap tahap pengembangan perangkat lunak (perencanaan, analisis, desain, pengembangan, implementasi, dan pemeliharaan) menemukan bahwa setiap tahap mempunyai risiko dominan yang bervariasi. Pada tahap perencanaan, risiko perencanaan sistem tidak jelas menjadi risiko yang paling mendominasi.

#### ABSTRACT

*Software development projects have been used successfully in various fields, but failure percentage is still high. In Indonesia, project's success rate is only 50%. This research aims to identify software developments risk with Indonesia context. This research answered two questions. (a) What are the risks in a software development project; (b) Is there any difference stages of software development for the risk found. The Delphi method is used to obtain expert opinion by a series of repeated questionnaires aimed at reaching consensus. This research uses the Delphi method involving 41 panelists with various background of roles in software development projects. This study found 17 risks based on panelist consensus. Risk mapping for each software development stage, (planning, analysis, design, development, implementation, and maintenance) found that each stage has various dominant risks. At the planning stage, the risk of system planning is not clear to be the most dominating risk.*

Teknologi: Jurnal Ilmiah Sistem Informasi dengan lisensi CC BY NC SA.

## 1. Pendahuluan

Proyek pengembangan perangkat lunak merupakan proses yang melibatkan perusahaan maupun individu dengan pengetahuan dan keahlian yang beragam (Akgün, 2020; Gupta, et al., 2019). Meskipun perangkat lunak telah berhasil diterapkan di berbagai bidang, namun reputasi kegagalannya masih cukup tinggi (Savolainen, Ahonen, & Richardson, 2012). Standish Group pada tahun 2015 melaporkan tingkat keberhasilan pengembangan perangkat lunak di Asia hanya 22%, 58% bermasalah dan 20% berakhir dengan kegagalan (International., 2015). Di Indonesia, tingkat keberhasilan implementasi

proyek masih relatif rendah yaitu 50%, hal tersebut dikarenakan dalam implementasinya sangat bergantung pada *Project Management Office* (PMO) (Raharjo, Purwandari, Satria, & Solichah, 2018). Terdapat lebih dari satu alasan kegagalan dalam proyek pengembangan perangkat lunak (Antony & Gupta, 2019; Haq, Gu, Liang, & Abdullah, 2019; Wong, Li, & Laplante, 2017). Faktor penyebab kegagalan proyek pengembangan perangkat lunak telah ditemukan. Ada 6 faktor penyebab kegagalan proyek pengembangan perangkat lunak, yaitu 1) Kurangnya kepedulian *top management*; 2) Estimasi biaya yang kurang tepat; 3) Risiko laten yang diabaikan; 4) Estimasi waktu yang sudah tepat; 5) Pengembang kurang percaya diri; dan 6) Komunikasi yang buruk (Sardjono & Retnowardhani, 2019). Langkah pencegahan dan menganalisis penyebab gagalnya proyek pengembangan perangkat lunak penting untuk dipertimbangkan (Lehtinen, Mäntylä, Vanhanen, Itkonen, & Lassenius, 2014).

Pada proyek pengembangan perangkat lunak terdapat sebuah integrasi yang kompleks, di mana satu kegagalan dapat memicu kegagalan dalam skala besar yang berakibat pada kerugian bagi organisasi dan dapat mengakibatkan kehancuran proyek (Masso, Pino, Pardo, García, & Piattini, 2020; Sriwindono & Yahya, 2012). Hal tersebut juga dapat mengeksploitasi pertumbuhan ekonomi dan kualitas hidup perusahaan (Zahid, Haider, Farooq, Abid, & Ali, 2018). 'Kegagalan' dapat dipahami sebagai ketidakpatuhan secara sistematis yang meluas dari kriteria suatu proyek yang berhasil, maka hal tersebut dapat menjadi salah satu penyebab kehancuran proyek. Namun, pendapat tersebut dapat berbeda tergantung pada jenis dan sektor proyek, sehingga pola penyebab kegagalan akan berbeda (Montequin, Cousillas, Alvarez, & Villanueva, 2016). Kualitas perangkat lunak juga dapat dipengaruhi oleh berbagai tahap pengembangan perangkat lunak (Singh & Gautam, 2016). Kombinasi dari beberapa faktor dan tingkat pengaruh yang berbeda dalam berbagai tahapan pengembangan proyek dapat mempengaruhi keberhasilan atau kegagalan suatu proyek (Montequin, Cousillas, Alvarez, & Villanueva, 2016). Evaluasi terhadap keberhasilan manajemen proyek berdasarkan tiap tahap pengembangan proyek diperlukan bantuan dari responden (Alzahrani & Emsley, 2013). Pada penelitian ini akan dikumpulkan panelis untuk memperoleh tanggapan terkait risiko penyebab kegagalan dalam setiap tahap pengembangan perangkat lunak.

Penelitian yang terkait dengan risiko kegagalan proyek pengembangan perangkat lunak sebelumnya telah dilakukan di berbagai negara (Gupta, et al., 2019; Montequin, Cousillas, Alvarez, & Villanueva, 2016; Taherdoost & Keshavarzsaleh, 2016; Wong, Li, & Laplante, 2017). Akan tetapi, penerimaan dan pengembangan teknologi informasi di Indonesia berbeda dengan negara lain yang dipengaruhi oleh perbedaan sektoral, geografis dan budaya (Montequin, Cousillas, Alvarez, & Villanueva, 2016; Sriwindono & Yahya, 2012). Pada penelitian ini telah dirangkum risiko kegagalan pengembangan perangkat lunak berdasarkan penelitian yang telah dilakukan sebelumnya pada Tabel 1.

Antony dan Gupta (2019) pada penelitiannya menemukan bahwa, risiko terbesar terjadi pada proyek perangkat lunak, tetapi pada penelitiannya tidak membahas secara detail mengenai letak risiko pada tahapan pengembangan, dan apakah terdapat perbedaan risiko untuk setiap tahap pengembangan serta risiko apa yang paling mendominasi pada tiap tahapannya. Rahman, Razali, dan Ismail (2019) pada penelitiannya menemukan bahwa, perubahan persyaratan merupakan risiko terbesar dalam proyek pengembangan perangkat lunak. Namun, Rahman, Razali, dan Ismail (2019) dalam penelitiannya hanya memfokuskan pada analisis dampak faktor risiko perubahan persyaratan saja. Riaz, Jahan, Arif, dan Butt (2019) telah berhasil mengklasifikasikan risiko untuk mengidentifikasi risiko dari akarnya, tetapi dalam melakukan klasifikasi risiko peneliti tersebut, hanya berdasarkan hasil kuesioner, bukan berdasarkan tahapan pengembangan. Hal tersebut menjadi dasar untuk dilakukannya penelitian ini. Oleh karena itu, penelitian ini dirancang untuk menjawab dua pertanyaan, yakni: 1) Apa risiko dalam proyek pengembangan perangkat lunak? dan 2) Apakah terdapat perbedaan tahap pengembangan perangkat lunak untuk risiko yang ditemukan?

Metode Delphi yang dikembangkan oleh Rand Corporation pada tahun 1950-an diterapkan pada penelitian ini guna menjawab pertanyaan tersebut (Cid, et al., 2015; Maher, et al., 2015). Metode Delphi dapat mengakomodasi pengetahuan yang diberikan oleh panelis (Lyu, et al., 2020). Dengan begitu, hasil dari temuan penelitian ini dapat diidentifikasi berdasarkan risiko yang dianggap paling berdampak pada kegagalan proyek dan sebagai dasar untuk melakukan pekerjaan di masa depan. Tujuan penelitian ini untuk mengetahui risiko yang menjadi pemicu terjadinya kegagalan dalam proyek

pengembangan perangkat lunak serta risiko yang mendominasi pada setiap tahapan proyek pengembangan perangkat lunak. Para pengembang maupun *stakeholder* penting untuk mengetahui risiko-risiko yang umum terjadi di Indonesia guna meminimalkan kegagalan proyek. Anggapan awal pada penelitian ini adalah risiko perbedaan budaya menjadi risiko yang paling berdampak pada kegagalan proyek pengembangan perangkat lunak di Indonesia. Karena Indonesia dipengaruhi oleh perbedaan sektoral, budaya dan geografis.

Tabel 1. Risiko kegagalan proyek pengembangan perangkat lunak berdasarkan penelitian sebelumnya

Risiko	Penelitian
Kurangnya komitmen <i>top management</i>	Antony dan Gupta (2019); Montequin, Cousillas, Alvarez, dan Villanueva (2016); Gupta, dkk (2019); Chien, Wu, dan Huang (2014).
Kurangnya keterlibatan pengguna Persyaratan yang tidak jelas	Kaur dan Sengupta (2011); Dwivedi, dkk (2015). Taherdoost dan Keshavarzsaleh (2016); Kaur dan Sengupta (2011); Montequin, Cousillas, Alvarez, dan Villanueva (2016); Hu, Zhang, Ngai, Cai, dan Liu (2013); Song dan Jiang (2016).
Ruang lingkup proyek tidak jelas Kurangnya kerjasama tim	Antony dan Gupta (2019); Whitney dan Daniels (2013). Antony dan Gupta (2019); Montequin, Cousillas, Alvarez, dan Villanueva (2016); Lehtinen, Mäntylä, Vanhanen, Itkonen, dan Lassenius (2014).
Penggunaan teknologi baru	Montequin, Cousillas, Alvarez, dan Villanueva (2016); Taherdoost dan Keshavarzsaleh (2016); Whitney dan Daniels (2013); Hu, Zhang, Ngai, Cai, dan Liu (2013); Song dan Jiang (2016).
Kurangnya keterampilan tim pengembang	Antony dan Gupta (2019); Montequin, Cousillas, Alvarez, dan Villanueva (2016); Wong, Li, dan Laplante (2017); Hu, Zhang, Ngai, Cai, dan Liu (2013); Song dan Jiang (2016).
Ukuran tim tidak sesuai	Antony dan Gupta (2019); Gupta, dkk (2019); Kaur dan Sengupta (2011); Hu, Zhang, Ngai, Cai, dan Liu (2013); Song dan Jiang (2016).
Estimasi biaya tidak tepat	Montequin, Cousillas, Alvarez, dan Villanueva (2016); Wong, Li, dan Laplante (2017); Chien, Wu, dan Huang (2014).
Estimasi waktu tidak sesuai	Kaur dan Sengupta (2011); Montequin, Cousillas, Alvarez, dan Villanueva (2016); Song dan Jiang (2016).
Manajemen proyek yang buruk	Gupta, dkk (2019); Kaur dan Sengupta (2011); Montequin, Cousillas, Alvarez, dan Villanueva (2016); Taherdoost dan Keshavarzsaleh (2016); Whitney dan Daniels (2013); Verner, Brereton, Kitchenham, Turner, dan Niazi (2014).
Gagal mendapatkan komitmen pengguna	Kaur dan Sengupta (2011); Hu, Zhang, Ngai, Cai, dan Liu (2013).
Gagal memenuhi permintaan pengguna	Montequin, Cousillas, Alvarez, dan Villanueva (2016); Wong, Li, dan Laplante (2017).
Komunikasi yang buruk	Antony dan Gupta (2019); Montequin, Cousillas, Alvarez, dan Villanueva (2016); Gupta, dkk (2019); Wong, Li, dan Laplante (2017); Verner, Brereton, Kitchenham, Turner, dan Niazi (2014); Fang dan Marle (2012).
Perbedaan budaya	Montequin, Cousillas, Alvarez, dan Villanueva (2016); Gupta, dkk (2019); Verner, Brereton, Kitchenham, Turner, dan Niazi (2014); Husin, dkk., (2019).

## 2. State of the Art

### 2.1. Risiko proyek perangkat lunak

Risiko diidentifikasi sebagai sebuah peristiwa yang dapat berdampak positif maupun negatif terhadap suatu proyek jika risiko itu terjadi (Kwan & Leung, 2011). Risiko proyek perangkat lunak dapat didefinisikan sebagai suatu kejadian atau ancaman yang tidak diharapkan karena dapat berpotensi mengakibatkan kerugian dan sering kali dianggap sebagai suatu kondisi yang dapat menghambat keberlangsungan proyek TI (Teknologi Informasi) (Indonesia, 2015; Keil, Rai, & Liu, 2013). Selama pengembangan perangkat lunak berlangsung, kemungkinan terjadinya risiko tidak dapat dihindari (Sangaiah, Samuel, Li, Abdel-Basset, & Wang, 2018). Risiko dapat merujuk pada properti atau aspek tertentu seperti tugas dalam pengembangan perangkat lunak, proses dan lingkungan yang apabila diabaikan kemungkinan kegagalan proyek pengembangan perangkat lunak akan semakin tinggi (Verner, Brereton, Kitchenham, Turner, & Niazi, 2014).

Risiko kegagalan proyek pengembangan perangkat lunak sering terjadi di berbagai negara, dan pada penelitian ini telah dirangkum seperti yang telah disajikan pada Tabel 1. Masing-masing peneliti telah menemukan 7 hingga 31 risiko kegagalan pengembangan perangkat lunak. Namun, pada

penelitian ini hanya merangkum risiko kegagalan pengembangan perangkat lunak yang memiliki kesamaan risiko kegagalan dengan peneliti lainnya. Berdasarkan hal tersebut, maka pada penelitian ini disimpulkan bahwa risiko-risiko tersebut merupakan risiko yang umum terjadi dan dianggap cukup penting.

## 2.2. Metode Delphi

Delphi digunakan untuk memperoleh pendapat ahli dengan serangkain kuesioner berulang dengan tujuan untuk mencapai konsensus (Dolan, Glynn, & Lawlor, 2020). Metode Delphi berbeda dengan metode lain karena dapat lebih diandalkan untuk menyelesaikan masalah dengan ketidakpastian yang tinggi (Ameyaw, Hu, Shan, Chan, & Le, 2016). Agar konsensus tercapai, setidaknya dibutuhkan dua atau lebih putaran pertanyaan berdasarkan umpan balik yang terkontrol dari panelis (Widiasih, Karningsih, & Ciptomulyono, 2015). Metode Delphi juga digunakan untuk menghindari pertemuan langsung antar panelis agar para panelis mempunyai kesempatan untuk mengutarakan pendapat mereka tanpa adanya intimidasi dari panelis lain (Daniel, et al., 2020). Metode Delphi bergantung pada keikutsertaan panelis secara berkelanjutan hingga mendapatkan hasil yang diinginkan (Knox, Snodgrass, Southgate, & Rivett, 2019).

Keuntungan yang diperoleh dari penggunaan metode ini yaitu memungkinkan untuk memiliki pendapat pakar dari berbagai wilayah karena berbasis internet, dapat mengurangi biaya manajemen serta kekayaan informasi yang diperoleh dengan memanfaatkan pendapat dari panelis (Fernández-Ávila, Rojas, & Rosselli, 2020). Delphi saat ini telah digunakan dalam berbagai penelitian, seperti peramalan (Flostrand, Pitt, & Bridson, 2020), Kesehatan (Denost, et al., 2020; Gunadi, et al., 2020), teknologi informasi (Antony & Gupta, 2019; Chien, Wu, & Huang, 2014; Montequin, Cousillas, Alvarez, & Villanueva, 2016; Wong, Li, & Laplante, 2017), Pendidikan (Knox, Snodgrass, Southgate, & Rivett, 2019), serta evaluasi (Balki, et al., 2017). Metode Delphi klasik terdiri dari 3 putaran pertanyaan (Ameyaw, Hu, Shan, Chan, & Le, 2016). Namun, putaran pertanyaan dapat dihentikan apabila telah mencapai konsensus dan jika pada putaran ketiga, konsensus belum diperoleh maka dapat ditambahkan putaran selanjutnya hingga mencapai konsensus (Widiasih, Karningsih, & Ciptomulyono, 2015).

## 3. Metode Penelitian

Metode Delphi diterapkan pada penelitian ini untuk memperoleh pendapat para ahli dengan menggunakan serangkaian kuesioner berulang yang bertujuan untuk mencapai konsensus. Tahapan pada penelitian ini terdiri dari: Perekrutan peserta Delphi, pengumpulan data (menggunakan tahapan metode Delphi seperti pada Gambar 1).

Batasan penelitian ini meliputi: 1) Risiko pengembangan perangkat lunak yang terjadi di Indonesia; 2) Panelis yang terlibat dalam penelitian adalah profesional TI dengan pengalaman lebih dari 3 tahun; 3) Metode yang digunakan adalah metode Delphi dengan 3 putaran pertanyaan 4) Penelitian hanya terpusat pada risiko dalam proyek pengembangan perangkat lunak dan risiko yang mendominasi pada setiap tahapan proyek pengembangan perangkat lunak; dan 5) Tahap pengembangan dalam penelitian ini hanya mencakup perencanaan, analisis, desain, pengembangan, implementasi dan pemeliharaan.

### 3.1. Peserta Delphi

Sebelum melakukan penelitian ini, 52 target panelis dengan peran yang berbeda-beda yang tersebar di berbagai wilayah di Indonesia dihubungi. Calon peserta Delphi diidentifikasi dan diminta kesediaannya. Pada tahap ini diperoleh 41 panelis dengan pengalaman lebih dari tiga tahun dalam proyek pengembangan perangkat lunak yang bersedia terlibat dalam penelitian ini. 41 panelis terdiri dari berbagai peran, diantaranya adalah: 40 terlihat dalam pengembangan, 9 menjadi manajer proyek, 4 pemilik produk, dan 2 terlibat dalam pengembangan UI/UX (*User Interface/User Experience*).

Panelis berasal dari beberapa daerah: 18 dari Yogyakarta, 10 Makassar, 8 Bandung, 3 Jakarta, dan 1 Kalimantan Timur. Pelibatan panelis dari beragam latar belakang ini, diharapkan dapat memberikan perspektif yang saling melengkapi.

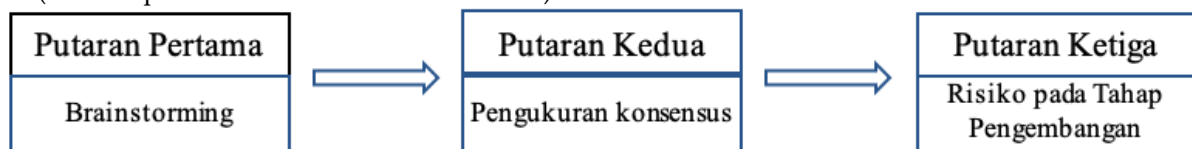
Dalam setiap pengembangan proyek, panelis pada penelitian ini beranggotakan antara 2-7 orang dalam satu tim pengembang, dengan proyek yang dikerjakan selama kurun waktu setahun mencapai

3-7 proyek yang mencakup sektor pendidikan, industri, kesehatan maupun pemerintahan. Panelis pada penelitian ini memiliki peran yang berkisar antara 40%-80% dalam pengembangan perangkat lunak, dan mempunyai peran yang berbeda-beda pada setiap tahap pengembangan.

### 3.2. Pengumpulan data

Penelitian ini menggunakan tiga putaran untuk menjawab dua pertanyaan yang menjadi fokus penelitian ini, seperti pada Gambar 1. Putaran pertama bertujuan untuk mengumpulkan risiko kegagalan berdasarkan sudut pandang panelis. Putaran kedua bertujuan untuk mencari konsensus terkait risiko kegagalan. Putaran ketiga bertujuan untuk mengetahui pada tahap pengembangan mana risiko tersebut muncul. Penyebaran pertanyaan menggunakan GoogleForm yang dikirim ke panelis melalui email. Adapun pertanyaan yang diberikan ke para panelis sebagai berikut:

1. Berdasarkan pengalaman Anda, apa lima (5) risiko yang paling berdampak pada kegagalan proyek pengembangan perangkat lunak?
2. Berikanlah penilaian untuk tiap-tiap risiko di bawah ini dengan ketentuan:
  - 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. Berdasarkan pengalaman Anda, risiko-risiko di bawah ini biasanya terjadi pada tahapan mana saja? (Anda diperkenankan memilih lebih dari 1).



Gambar 1. Tahap metode Delphi

Penjelasan secara detail terkait Gambar 1 adalah sebagai berikut:

1. Putaran Pertanyaan Pertama  
Putaran ini dikenal juga dengan tahap *brainstorming*. Pada penelitian ini diberikan pertanyaan terbuka dengan membebaskan panelis untuk memberikan pandangan mereka terkait 5 risiko kegagalan yang dianggap paling berdampak pada kegagalan pengembangan perangkat lunak.
2. Putaran Pertanyaan Kedua  
Putaran kedua ini berasal dari hasil analisis pada putaran pertama yang kemudian disebar secara anonim untuk mempertimbangkan dan mengevaluasi kembali input yang diperoleh sebelumnya. Penyebaran pertanyaan pada putaran ini menggunakan penilaian skala *likert* 1-5. Pengukuran konsensus dilakukan dengan menggunakan *Standard Deviation* (SD) dan *Interquartile Range* (IR). Hasil analisis statistik digunakan untuk menyatakan apakah setiap variabel merupakan konsensus adalah jika nilai SD <1,5 dan nilai IR <1 (Giannarou & Zervas, 2014). 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. Persamaan 1 adalah persamaan SD,

$$s = \sqrt{\frac{\sum(x_i - \bar{x})^2}{n-1}} \text{ atau } \sqrt{\frac{\sum x_i^2 - \frac{(\sum x_i)^2}{n}}{n-1}} \quad (1)$$

di mana  $x$  adalah jawaban panelis A terhadap variabel  $n$  dan  $\bar{x}$  adalah rata-rata jawaban panelis terhadap variabel  $n$ .

- b. *Interquartile Range* (IR)

Selain SD, ukuran penilaian konsensus lainnya yaitu IR, di mana nilai nya <1. Persamaan IR disajikan pada Persamaan 2,

$$IR = Q_3 - Q_1 \quad (2)$$

di mana  $Q_3$  merupakan kuartil atas dan  $Q_1$  adalah kuartil bawah. Adapun persamaan kuartil  $Q_1$ ,  $Q_2$ , dan  $Q_3$  dijamin pada Persamaan 3, Persamaan 4, dan Persamaan 5,

$$Q_1 = \frac{x^{\frac{(n-1)}{4}} + x^{\frac{(n+3)}{4}}}{2} \quad (3)$$

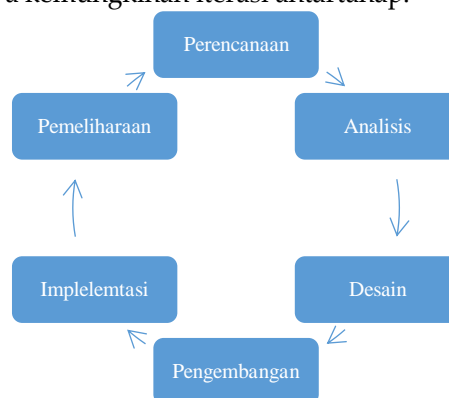
$$Q_2 = x^{\frac{2(n+1)}{4}} \quad (4)$$

$$Q_3 = \frac{x^{\frac{(3n+1)}{4}} + x^{\frac{(3n+5)}{4}}}{2} \quad (5)$$

### 3. Putaran Pertanyaan Ketiga

Putaran pertanyaan ini merujuk dari hasil risiko yang diperoleh pada putaran kedua yang telah mencapai konsensus. Pada penelitian ini, panelis diminta untuk memberikan pandangannya guna mengkaji pada tahap pengembangan mana saja risiko-risiko tersebut muncul.

Dalam penelitian ini, tahap pengembangan perangkat lunak diadopsi untuk memudahkan menangkap perbedaan antar tahap, adalah model sekuensial seperti pada Gambar 2 yang terdiri dari enam tahap: Perencanaan, analisis, desain, pengembangan, implementasi, dan pemeliharaan. Tahap ini merupakan sintesis dari beragam model. Di lapangan, model seperti ini merupakan penyederhaan karena adanya kemungkinan iterasi antartahap.



Gambar 2. Tahap pengembangan

Sebagai contoh, seperti pada metode Agile dengan tahap perencanaan, analisis, desain, implementasi, pengembangan, dan pemeliharaan (Cooper & Sommer, 2016; Khalil, Reza, Junaedi, & Kanigoro, 2015; Loiro, et al., 2019). Pada metode Scrum menggunakan tahapan perencanaan, analisis, dan pengembangan (Chaouch, Mejri, & Ghannouchi, 2019; Kussunga & Ribeiro, 2019; Nidagundi & Novickis, 2017). Metode *Software Development Life Cycle* (SDLC) menggunakan tahapan pengembangan, analisis, implementasi dan desain (Ahmed, Quddus, Kannan, Peres, & Mannan, 2020; Cai, Wu, Zhou, Liu, & Zhang, 2020; Lee, et al., 2020), pada metode *Waterfall* menggunakan tahapan analisis, desain, implementasi, dan pemeliharaan (Andrei, Casu-Pop, Gheorghe, & Boiangiu, 2019; Bhavsar, Shah, & Gopalan, 2020; Sommerville, 2003), sedangkan pada metode *Rapid Application Development* (RAD) menggunakan tahapan perencanaan, analisis, desain dan pengembangan (Andreswari, et al., 2020; Chrismanto, Santoso, Wibowo, Delima, & Kristiawan, 2019).

Tahap perencanaan merupakan hal yang penting dalam pengembangan perangkat lunak, biasanya melibatkan pembuatan perencanaan, anggaran, mitigasi risiko, waktu pengembangan dan lain-lain. Tahap analisis merupakan proses untuk memperoleh informasi, spesifikasi serta model yang diinginkan pengguna agar dijadikan acuan pelaksanaan desain perangkat lunak (Simarmata, 2010). Tahap desain merupakan proses pemecahan masalah dan perencanaan pengembangan perangkat lunak dengan melibatkan pengembang perangkat lunak (Bassil, 2012).

Tahap pengembangan bertujuan untuk evaluasi persyaratan agar integrasi yang direncanakan sesuai dengan persyaratan sebelumnya (Broad, 2013). Tahap implementasi merupakan proses pengubahan persyaratan dan spesifikasi desain menjadi program yang dapat dioperasikan (Bassil, 2012). Tahap pemeliharaan merupakan proses modifikasi dari hasil implementasi agar dapat memperbaiki kesalahan serta meningkatkan kinerja dan kualitas program (Bassil, 2012).

### 4. Hasil dan Pembahasan

Pertanyaan putaran pertama disebar pada 41 panelis, dan 41 panelis menyelesaikan putaran tersebut. Pada putaran kedua, pertanyaan disebar kepada 41 panelis dan 32 panelis menyelesaikan putaran

pertanyaan kedua. Untuk putaran ketiga, pertanyaan disebarkan kepada 32 panelis dan 27 panelis menyelesaikan putaran ketiga.

#### 4.1. Risiko kegagalan

Dari survei yang telah dilakukan, diperoleh 17 risiko kegagalan berdasarkan konsensus panelis terkait risiko kegagalan proyek pengembangan perangkat lunak, seperti yang disajikan pada Tabel 2. Risiko yang telah ditemukan pada penelitian ini merupakan risiko dalam proyek pengembangan perangkat lunak yang umum dijumpai oleh para panelis maupun oleh para pelaku bisnis di bidang pengembangan perangkat lunak. Ke 17 risiko tersebut, kemudian dilakukan pemeringkatan berdasarkan tingkat kepentingan risiko yang diperoleh dari masukan panelis.

Tabel 2. Risiko kegagalan

Kode	Risiko Kegagalan	Rerata	SD	IR	Ranking
R.01	Perencanaan sistem tidak jelas	4,68	0,52	1	1
R.02	Kurangnya analisis kebutuhan/proses bisnis	4,65	0,64	0,25	2
R.03	Tim yang tidak profesional	4,56	0,65	1	3
R.04	Kurangnya komunikasi	4,34	0,81	1	4
R.05	Kurangnya kerjasama tim	4,31	0,63	1	5
R.06	Manajemen proyek tidak jelas	4,28	0,67	1	6
R.07	Kurangnya keahlian/pengetahuan tim	4,25	0,75	1	7
R.08	Estimasi waktu tidak tepat	4,21	0,59	1	8
R.09	Ruang lingkup yang tidak sesuai	3,93	0,65	0	9
R.10	Buruknya pemantauan sistem	3,93	0,86	0,25	10
R.11	Kurangnya tim pengembang	3,84	0,75	1	11
R.12	Fasilitas kurang memadai	3,68	0,84	1	12
R.13	Kurangnya kepercayaan pengguna	3,59	0,78	1	13
R.14	Kurangnya pemahaman pengguna	3,40	0,82	1	14
R.15	Teknologi belum teruji	3,34	0,81	1	15
R.16	Akuisisi pengguna yang rendah	3,12	0,85	1	16
R.17	Perbedaan budaya	2,81	0,88	1	17

Temuan yang menarik untuk dibahas yaitu, meskipun panelis memiliki latar belakang peran dan wilayah yang berbeda, tetapi secara independen panelis menyepakati 17 risiko sebagai risiko yang dianggap berpengaruh terhadap keberlangsungan proyek pengembangan perangkat lunak di Indonesia.

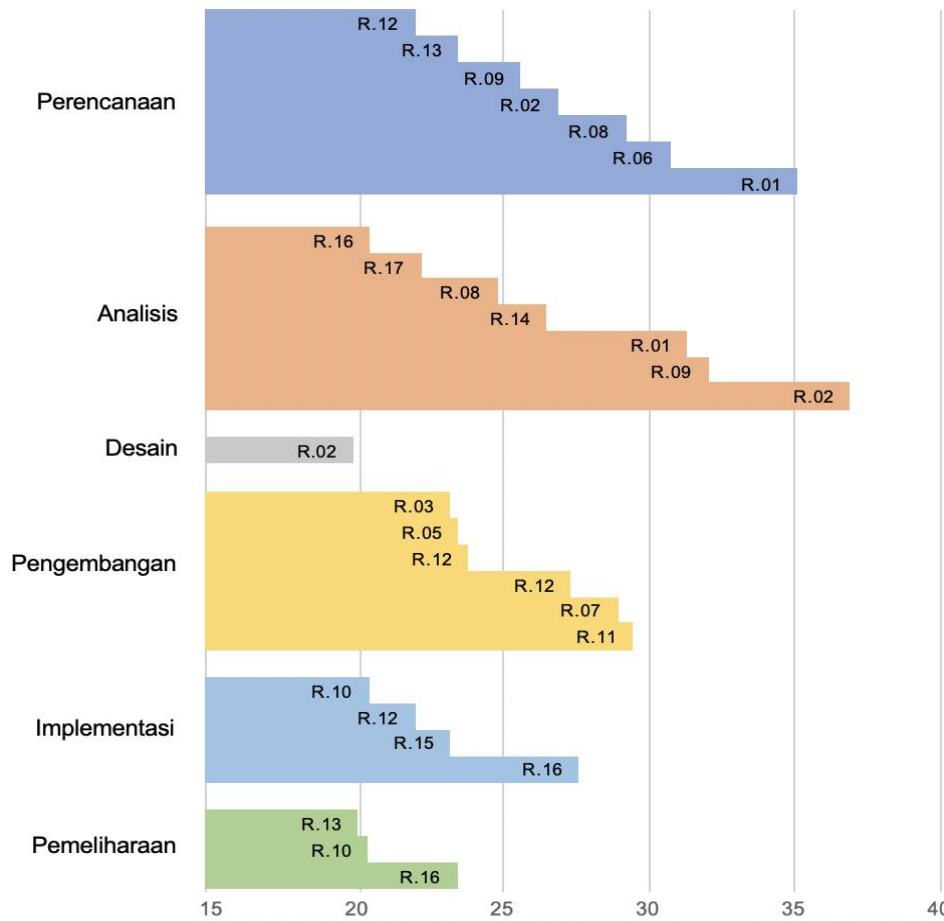
Sebagian besar dari panelis sepakat bahwa perencanaan sistem yang tidak jelas merupakan risiko yang menempati urutan pertama sebagai risiko yang paling berdampak pada kegagalan proyek pengembangan perangkat lunak di Indonesia. Adapun perencanaan sistem yang tidak jelas menurut panelis, yaitu:

1. Perencanaan yang tidak realistis.
2. Perencanaan sistem kurang matang.
3. Kurangnya informasi terkait detail desain sistem yang dibuat.
4. Tidak ditentukannya minimum variabel produk.
5. Terlambat melakukan pivot.

#### 4.2. Tahap pengembangan

Penerapan strategi manajemen risiko tentu melibatkan penilaian risiko dan mitigasi risiko dengan tujuan membantu dalam identifikasi risiko dan mitigasinya selama pengembangan proyek perangkat lunak berlangsung. Risiko selanjutnya divisualisasikan berdasarkan risiko yang paling dominan, seperti yang disajikan pada Gambar 3. Pemetaan risiko yang paling dominan tersebut diperoleh berdasarkan hasil putaran pertanyaan tahap ke-3 (Gambar 1). Para panelis diminta untuk memetakan risiko berdasarkan tahapan pengembangan menurut pengalaman para panelis. Panelis dapat memetakan satu risiko ke dalam lebih dari satu tahap pengembangan. Hasil pemetaan para panelis tersebut kemudian dirangkum dalam bentuk persentase, di mana persentase lebih dari 20% dianggap memenuhi syarat sebagai risiko yang paling dominan atau risiko tertinggi, seperti yang disajikan pada Gambar 3. Risiko dengan angka di bawah 20% tidak dimasukkan ke dalam Gambar 3 dengan tujuan untuk memudahkan keterbacaan.

Dengan demikian, hasil dari penelitian ini dapat dipemetaan risiko yang paling mendominasi sehingga dapat dicegah dengan melakukan beberapa hal. Dengan hasil penelitian ini dapat diantisipasi terjadinya risiko yang terjadi pada tiap tahap pengembangan seperti pada tahap perencanaan, risiko yang akan muncul adalah perencanaan sistem tidak jelas. Sehingga perusahaan perlu merencanakan perencanaan sistem sedetail mungkin untuk menghindari risiko tersebut.



Gambar 3. Pemetaan risiko paling dominan

Ke 17 risiko tersebut tersebar di enam tahap pengembangan seperti perencanaan, analisis, desain, pengembangan, implementasi, dan pemeliharaan. Dari risiko yang tersebar tersebut, ternyata terdapat perbedaan tahap pengembangan perangkat lunak untuk risiko yang ditemukan.

Tahap perencanaan didapatkan tujuh risiko yaitu R.01 (perencanaan sistem tidak jelas), R.06 (manajemen proyek tidak jelas), R.08 (Estimasi waktu tidak tepat), R.02 (kurangnya analisis kebutuhan/proses bisnis), R.09 (ruang lingkup yang tidak sesuai), R.13 (kurangnya kepercayaan pengguna), dan R.12 (fasilitas kurang memadai). Tahap analisis didapatkan tujuh risiko yaitu R.02 (kurangnya analisis kebutuhan/proses bisnis), R.09 (ruang lingkup yang tidak sesuai), R.01 (perencanaan sistem tidak jelas), R.14 (kurangnya pemahaman pengguna), R.08 (estimasi waktu tidak tepat), R.17 (perbedaan budaya), dan R.16 (akuisisi pengguna yang rendah). Temuan menarik terdapat pada tahap desain, karena hanya terdapat satu risiko yaitu R.02 (kurangnya analisis kebutuhan/proses bisnis).

Tahap pengembangan didapatkan enam risiko yaitu R.11 (kurangnya tim pengembang), R.07 (kurangnya keahlian/pengetahuan tim), R.04 (kurangnya komunikasi), R.12 (fasilitas kurang memadai), R.05 (kurangnya kerjasama tim), dan R.03 (tim yang tidak profesional). Tahap implementasi didapatkan empat risiko yaitu risiko R.16 (akuisisi pengguna yang rendah), R.15 (teknologi belum teruji), R.12 (fasilitas kurang memadai), dan R.10 (buruknya pemantauan sistem). Sementara itu, tahap pemeliharaan hanya didapatkan tiga risiko yaitu R.16 (akuisisi pengguna yang rendah), R.10 (buruknya pemantauan sistem), dan R.13 (kurangnya kepercayaan pengguna).

Dari 17 risiko yang tersebar di enam tahap pengembangan, ternyata terdapat satu risiko yang paling dominan diantara enam tahap pengembangan tersebut. Risiko tersebut yaitu R.02 (kurangnya



analisis kebutuhan/proses bisnis) yang terdapat pada tahap analisis dengan persentase sebanyak 37,14%. Tetapi risiko yang sama, R.02, juga menjadi yang paling rendah persentasenya berdasarkan pemetaan risiko pada tahap desain dengan nilai persentase hanya mencapai 20,00%.

Temuan yang didapatkan pada penelitian ini memiliki persamaan dan perbedaan dengan peneliti sebelumnya Roy, Dasgupta, dan Chaki (2016), di mana risiko tertinggi pada tahap perencanaan pada penelitian ini ternyata tidak termasuk sebagai risiko yang ada pada penelitian sebelumnya. Pada tahap analisis dan desain, R.02 menjadi risiko yang paling dominan. Hal tersebut sejalan dengan hasil penelitian sebelumnya, bahwa R.02 atau kurangnya analisis kebutuhan/proses bisnis menjadi salah satu risiko yang dominan pada tahap desain, analisis bahkan terdapat juga pada tahap perencanaan. Di tahap pengembangan, R.11 menjadi yang paling dominan pada penelitian ini, sedangkan pada penelitian terdahulu R.11 atau kurangnya tim pengembang juga menjadi salah satu risiko yang terdapat pada tahap pengembangan. Sementara itu, pada penelitian ini maupun penelitian sebelumnya sepakat bahwa akuisisi pengguna rendah atau R.16 menjadi risiko yang dominan pada tahap implementasi dan pemeliharaan.

## 5. Kesimpulan

Identifikasi risiko yang dianggap paling berdampak dan perbedaan risiko untuk setiap tahap pengembangan dapat diperoleh dengan menggunakan metode Delphi dan pemanfaatan pengalaman 41 panelis yang terlibat dalam proyek pengembangan perangkat lunak di Indonesia. Berdasarkan hal tersebut, maka dapat disimpulkan sebagai berikut: 1) Pengumpulan data dengan menggunakan metode Delphi diperoleh 17 risiko yang mencapai nilai konsensus dengan risiko perencanaan sistem tidak jelas menjadi risiko tertinggi terjadinya kegagalan proyek. Dari 17 tersebut ternyata memiliki rentang nilai *mean* yang tidak terlalu jauh. Hal tersebut menandakan bahwa risiko-risiko tersebut memiliki dampak yang tidak jauh berbeda terhadap kegagalan proyek pengembangan perangkat lunak. 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, tetapi perbedaan budaya menjadi salah satu risiko yang berdampak pada kegagalan proyek pengembangan perangkat lunak di Indonesia; dan 2) Berdasarkan Gambar 3, tiap tahap pengembangan ternyata mempunyai dominan risiko yang berbeda-beda. Tahap perencanaan risiko perencanaan tidak jelas menjadi risiko yang dominan. Risiko kurangnya analisis kebutuhan/proses bisnis menjadi risiko yang dominan pada tahap analisis dan tahap desain. Risiko kurangnya analisis kebutuhan/proses bisnis menjadi satu-satunya risiko yang mendominasi atau yang mencapai nilai standar dominan 20% di tahap desain. Risiko kurangnya tim pengembang menjadi risiko yang mendominasi pada tahap pengembangan. Sementara, risiko akuisisi pengguna rendah mendominasi tahap implementasi dan tahap pemeliharaan.

## 6. Impikasi dan Pekerjaan Masa Depan

Pada penelitian ini ditemukan beberapa risiko kegagalan baru berdasarkan penelitian yang telah dilakukan oleh peneliti terdahulu terkait risiko kegagalan proyek pengembangan perangkat lunak, diantaranya adalah: 1) Kurangnya analisis kebutuhan/proses bisnis; 2) Tim yang tidak profesional; 3) Kurangnya pemahaman pengguna; 4) Fasilitas kurang memadai; 5) Kurangnya kepercayaan pengguna; 6) Buruknya pemantauan system; dan 7) Akuisisi pengguna rendah. Terdapat beberapa implikasi praktis yang diidentifikasi, yakni: 1) Praktisi pengembang perangkat lunak perlu menyadari ada risiko untuk setiap tahap pengembangan; dan 2) Risiko dominan untuk setiap tahap juga berbeda.

Meskipun pada penelitian ini berhasil mengidentifikasi faktor risiko kegagalan yang paling penting, pada penelitian ini juga disajikan risiko yang paling mendominasi pada setiap tahap pengembangannya, tetapi diperlukan lebih banyak penelitian untuk menyajikan mitigasi pada setiap risiko yang ditemukan. Oleh karena itu, pada penelitian selanjutnya akan disajikan strategi mitigasi risiko pada setiap risiko yang ditemukan.

## 7. Referensi

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.

- Akgün, A. E. (2020). Team wisdom in software development projects and its impact on project performance. *International Journal of Information Management*, 50, 228-243.
- Alzahrani, J. I., & Emsley, M. W. (2013). The impact of contractors' attributes on construction project success: A post construction evaluation. *International Journal of Project Management*, 31(2), 313-322.
- Ameyaw, E. E., Hu, Y., Shan, M., Chan, A. P., & 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.
- Andrei, B.-A., Casu-Pop, A.-C., Gheorghe, S.-C., & Boianuiu, C.-A. (2019). A Study On Using Waterfall And Agile Methods In Software Project Management. *Journal Of Information Systems & Operations Management*, 125-135.
- 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(1), 35-4.
- Antony, J., & Gupta, S. (2019). Top ten reasons for process improvement project failures. *International Journal of Lean Six Sigma*, 10(1), 367-374.
- 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.
- Bassil, Y. (2012). A Simulation Model for the Waterfall Software Development Life Cycle. *International Journal of Engineering & Technology (ijET)*, 2(5).
- 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 (IJITEE)*, 9(4), 2075-2084.
- Broad, J. (2013). *Risk Management Framework: A Lab-Based Approach to Securing Information Systems*. Waltham: Elsevier.
- 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.
- Chaouch, S., Mejri, A., & Ghannouchi, S. A. (2019). A framework for risk management in Scrum development process. *Procedia Computer Science*, 164, 187-192.
- 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.
- Chrismanto, A. R., Santoso, H. B., Wibowo, A., Delima, R., & 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 (IJACSA)*, 10(10).
- Cid, J., Calle, J. L., López, E., Pozo, C. D., Perucho, A., Acedo, M. S., . . . Joaquín. (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.
- 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.
- 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. *The American Journal of Surgery*, 219(1), 75-79.
- Denost, Q., Bousser, V., Morin-Porchet, C., Vincent, C., Pinon, E., Collin, F., . . . Saillour-Glénisson, F. (2020). The development of a regional referral pathway for locally recurrent rectal cancer: A Delphi consensus study. *European Journal of Surgical Oncology*, 46(3), 470-475.
- Dolan, C., Glynn, R., & Lawlor, B. (2020). A Systematic Review and Delphi Study to Ascertain Common Risk Factors for Type 2 Diabetes Mellitus and Dementia and Brain-Related Complications of Diabetes in Adults. *Canadian Journal of Diabetes*.
- Dwivedi, Y. K., Wastell, D., Laumer, S., Henriksen, H. Z., Myers, M. D., Bunker, D., . . . Srivastava, S. C. (2015). Research on information systems failures and successes: Status update and future directions. *Information Systems Frontiers*, 17, 143-157.

- 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.
- 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*.
- 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.
- Giannarou, L., & Zervas, E. (2014). Using Delphi technique to build consensus in practice. *International Journal of Business Science & Applied Management (IJBSAM)*, 9(2), 65-82.
- Gunadi, G., 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, 37-40.
- 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 & Industrial Engineering*, 127, 27.
- Haq, S. U., Gu, D., Liang, C., & Abdullah, I. (2019). Project governance mechanisms and the performance of software development projects: Moderating role of requirements risk. *International Journal of Project Management*, 37(4), 533-548.
- Hu, Y., Zhang, X., Ngai, E., Cai, R., & Liu, M. (2013). Software project risk analysis using Bayesian networks with causality constraints. *Decision Support Systems*, 56, 439-449.
- Husin, W. S., Yahya, Y., Azmi, N. F., 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.
- Indonesia, I. B. (2015). *Manajemen Risiko 1*. Jakarta: Gramedia Pustaka Utama.
- International, T. S. (2015). *Chaos Report 2015*. The Standish Group International. Retrieved from [https://www.standishgroup.com/sample\\_research\\_files/CHAOSReport2015-Final.pdf](https://www.standishgroup.com/sample_research_files/CHAOSReport2015-Final.pdf)
- Kaur, R., & Sengupta, J. (2011). Software Process Models and Analysis on Failure of Software Development Projects. *International Journal of Scientific & Engineering Research*, 2(2).
- Keil, M., Rai, A., & Liu, S. (2013). How user risk and requirements risk moderate the effects of formal and informal control on the process performance of IT projects. *European Journal of Information Systems*, 22(6), 650-672.
- Khalil, A.-A., Reza, A., Junaedi, P. A., & Kanigoro, B. (2015). Data Visualization Application for Analyzing Public Company Financial Statement. *Procedia Computer Science*, 59, 45-53.
- 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.
- Kussunga, F., & Ribeiro, P. (2019). Proposal of a Visual Environment to Support Scrum. *Procedia Computer Science*, 164, 491-497.
- Kwan, T. W., & Leung, H. K. (2011). A Risk Management Methodology for Project Risk Dependencies. *IEEE Transactions on Software Engineering*, 37(5), 635-648.
- Lee, S. H., Lee, S. J., Koo, S. R., Varuttamaseni, A., Yue, M., Li, M., . . . 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.
- Lehtinen, T. O., 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.
- 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.
- Lyu, F., Zheng, C., Wang, H., Nie, C., Ma, X., Xia, X., . . . 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.
- Maher, T. M., Whyte, M. K., Hoyles, R. K., Parfrey, H., Ochiai, Y., Mathieson, 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, 929–943.
- Masso, J., Pino, F. J., Pardo, C., García, F., & Piattini, M. (2020). Risk management in the software life cycle: A systematic literature review. *Computer Standards & Interfaces*, 71.
- Montequin, V., Cousillas, S., 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.
- Nidagundi, P., & Novickis, L. (2017). Introducing Lean Canvas Model Adaptation in the Scrum Software Testing. *Procedia Computer Science*, 104, 97-103.
- Raharjo, T., Purwandari, B., Satria, R., & Solichah, I. (2018). Critical Success Factors for Project Management Office: An Insight from Indonesia. *2018 Third International Conference on Informatics and Computing (ICIC)*. Palembang: IEEE.
- Rahman, M. A., Razali, R., & Ismail, F. F. (2019). Risk Factors for Software Requirements Change Implementation. *International Journal of Advanced Computer Science and Applications (Ijacs)*, 10(3).
- Riaz, M. T., Jahan, M. S., Arif, K. S., & Butt, W. H. (2019). Risk Assessment on Software Development using Fishbone Analysis. *International Conference on Data and Software Engineering (ICoDSE)*. Pontianak: IEEE.
- Roy, B., Dasgupta, R., & Chaki, N. (2016). A Study on Software Risk Management Strategies and Mapping with SDLC. In R. Chaki, A. Cortesi, K. Saeed, & N. Chaki, *Advanced Computing and Systems for Security*. New Delhi: Springer.
- Sangaiah, A. K., Samuel, O. W., Li, X., Abdel-Basset, M., & Wang, H. (2018). Towards an efficient risk assessment in software projects–Fuzzy reinforcement paradigm. *Computers & Electrical Engineering*, 833-846.
- Sardjono, W., & Retnowardhani, A. (2019). Analysis of Failure Factors in Information Systems Project for Software Implementation at The organization. *2019 International Conference on Information Management and Technology (ICIMTech)*. Jakarta/Bali: IEEE.
- Savolainen, P., Ahonen, J. J., & Richardson, I. (2012). Software development project success and failure from the supplier's perspective: A systematic literature review. *International Journal of Project Management*, 30(4), 458-469.
- Simarmata, J. (2010). *Rekayasa Perangkat Lunak*. Yogyakarta: Andi Publisher.
- Singh, B., & Gautam, S. (2016). The Impact of Software Development Process on Software Quality: A Review. *8th International Conference on Computational Intelligence and Communication Networks (CICN)* (pp. 666-672). Tehri: IEEE. doi:<https://doi.org/10.1109/CICN.2016.137>
- Sommerville, I. (2003). *Software engineering* (9 ed.). Addison-Wesley.
- Song, H., & Jiang, J. (2016). Risks Identification in Embedded Software Development: Evidence from MVBC Project Survey. *Procedia Computer Science*, 91, 798-806.
- Sriwindono, H., & Yahya, S. (2012). Toward Modeling the Effects of Cultural Dimension on ICT Acceptance in Indonesia. *Procedia - Social and Behavioral Sciences*, 65, 833-838.
- Taherdoost, H., & Keshavarzsaleh, A. (2016). Critical Factors that Lead to Projects' Success/Failure in Global Marketplace. *Procedia Technology*, 22, 1066-1075.
- Verner, J., Brereton, O., Kitchenham, B., Turner, M., & Niazi, M. (2014). Risks and risk mitigation in global software development: A tertiary study. *Information and Software Technology*, 56(1), 54-78.
- Whitney, K. M., & Daniels, C. B. (2013). The Root Cause of Failure in Complex IT Projects: Complexity Itself. *Procedia Computer Science*, 20, 325-330.
- 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, 282-290.
- 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.
- Zahid, A. H., Haider, M. W., Farooq, M. S., Abid, A., & Ali, A. (2018). A Critical Analysis of Software Failure Causes from Project Management Perspectives. *VFAST Transactions on Software Engineering*, 13(3).