

## Studi Komparasi Kerangka Kerja Manajemen Risiko dalam Scrum

Marzuki Pilliang<sup>1</sup>, Budi Tjahjono<sup>2\*</sup>

<sup>1</sup>Fakultas Ilmu Komputer, Universitas Esa Unggul, Jakarta Barat, DKI Jakarta, Indonesia 11510  
e-mail: <sup>1</sup>marzuki.pilliang@student.esaunggul.ac.id, <sup>2</sup> budi.tjahjono@esaunggul.ac.id

Submitted Date: January 16<sup>th</sup>, 2022  
Revised Date: June 06<sup>th</sup>, 2022

Reviewed Date: February 01<sup>st</sup>, 2022  
Accepted Date: August 16<sup>th</sup>, 2022

### Abstract

*Risk management in software engineering projects describes an integrated design with methods, processes, and artifacts that continuously identify, analyze, control, and monitor risks, to prevent the project from failing. Agile methodology is an alternative to the traditional sequential software development process. Scrum is the most frequently used method based on the 2016 Agile development survey results. In recent years, there have been many studies that have produced a risk management framework for Scrum. However, the process of risk analysis and the selection of responses to risks becomes a burden for stakeholders, so a framework is needed that can become a support system to help make decisions. This paper uses a comparative study of risk management framework literature and literature that utilizes tools for risk management. The research resulted in a new framework that integrates datasets and machine learning into a risk management framework, so further work can be done to test the effectiveness of the new framework.*

*Keywords: comparative study; framework; risk management; software engineering project; scrum*

### Abstrak

Manajemen risiko dalam proyek rekayasa perangkat lunak menggambarkan perancangan terpadu dengan metode, proses, dan artefak yang terus mengidentifikasi, menganalisa, mengontrol, dan memantau risiko, guna mencegah proyek menjadi gagal. Metodologi Agile dikenal sebagai alternatif untuk proses pengembangan perangkat lunak tradisional yang sekuensial. Scrum adalah metode yang paling sering digunakan berdasarkan hasil survey *Agile development* tahun 2016. Tahun-tahun belakangan ini, telah banyak penelitian yang menghasilkan kerangka kerja manajemen risiko untuk Scrum. Namun pengulangan proses analisa risiko dan pemilihan respon terhadap risiko menjadi beban tersendiri bagi para pemangku kepentingan, sehingga dibutuhkan kerangka kerja yang bisa menjadi sebuah *support system* untuk membantu membuat keputusan. Makalah ini menggunakan metode studi komparasi literatur kerangka kerja manajemen risiko serta literatur yang memanfaatkan *tools* untuk manajemen risiko. Penelitian menghasilkan sebuah kerangka kerja baru yang memadukan *dataset* dan *machine learning* ke dalam kerangka kerja manajemen risiko, dengan begitu penelitian lanjutan dapat dilakukan guna menguji efektivitas dari kerangka kerja baru tersebut.

Kata Kunci: studi komparasi; kerangka kerja; manajemen risiko; proyek rekayasa perangkat lunak; scrum

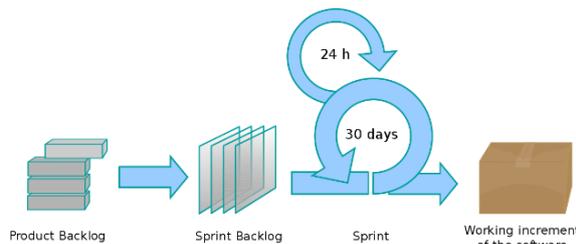
### 1. Pendahuluan

Risiko utama dalam suatu proyek adalah kegagalan proyek dalam mencapai tujuan maupun memenuhi kriteria batasan proyek (Rudy, 2016). Risiko juga mengacu pada kondisi di masa yang akan datang atau suatu kondisi yang terjadi di luar kendali, sehingga dapat menimbulkan dampak yang merugikan proyek (Dey et al., 2007). Sejak

awal tahun 2000, industri perangkat lunak telah beralih untuk mengadopsi metode Agile yang tanggap dan fleksibel terhadap perubahan, dibandingkan dengan metode tradisional (Nyfjord, 2008). Perpindahan dari model tradisional, seperti model sekuensial linier ke metode Agile telah menciptakan tantangan baru di bidang manajemen risiko. Scrum dan metodologi Agile lainnya,

memang secara umum, tidak menyarankan aktivitas khusus untuk mendukung manajemen risiko (Moran, 2014). Scrum mencakup sekelompok individu yang secara kolektif memiliki semua kemampuan dan penguasaan untuk melakukan pekerjaan dan berbagi atau memperoleh kemampuan yang diperlukan (Schwaber Ken & Sutherland Jeff, 2020). Scrum dipimpin oleh seorang Master Scrum untuk memelihara lingkungan (*life cycle*) dimana:

1. *Product Owner* mengatur tugas-tugas untuk permasalahan yang kompleks ke dalam *Product Backlog*.
2. *Scrum Team* membuat sebagian tugas tersebut menjadi suatu nilai yang meningkat (*increment version*) selama satu Sprint.
3. *Scrum Team* serta para *stakeholder* menguji hasil dan menyelaraskannya untuk Sprint berikutnya.
4. Pengulangan kembali ke proses atas, lihat ilustrasi alur proses pada Gambar 1.



Gambar 1 Alur proses Scrum

Sumber:

[https://en.wikipedia.org/wiki/File:Scrum\\_process.svg](https://en.wikipedia.org/wiki/File:Scrum_process.svg)

Pondasi Scrum terletak pada empirisme dan pemikiran lean. Empirisme menekankan bahwa pengetahuan bersumber dari pengalaman dan membuat pilihan berdasarkan apa yang diamati. Pemikiran lean memangkas pemborosan dan berpusat pada hal-hal utama. Scrum memakai pendekatan berulang dan bertahap untuk mengoptimalkan konsistensi dan mengontrol risiko (Schwaber Ken & Sutherland Jeff, 2020). Para profesional Scrum tidak sepenuhnya menyadari risiko, walaupun tetap mengikuti proses manajemen risiko secara implisit. Karena mereka berpikir bahwa siklus hidup pengembangan telah meminimalkan efek tak terduga dari pengembangan perangkat lunak (Chaouch et al., 2019).

Namun hal demikian tidaklah cukup, sehingga sampai saat ini belum ada kerangka kerja manajemen risiko yang secara lengkap dapat dijadikan suatu fasilitas bagi para tim Scrum dalam mengelola risiko di proyek-proyek Scrum secara mudah, serta dapat meningkatkan level dari keberhasilan proyek itu sendiri.

Dalam makalah ini dilakukan studi komparasi kerangka kerja manajemen risiko dan *risk management tool* yang diintegrasikan dengan prinsip-prinsip manajemen risiko ke dalam Scrum *life cycle* berdasarkan literatur-literatur dari penelitian sebelumnya.

Makalah ini disusun sebagai berikut: bagian 2 akan menjelaskan tentang konsep proses manajemen risiko beserta kerangka kerja manajemen risiko yang diusung oleh para peneliti sebelumnya; bagian 3 menjelaskan tentang pentingnya adaptasi pilar Scrum dalam manajemen risiko; bagian 4 membahas konsep-konsep penggunaan *dataset* dan *machine learning* sebagai *tools* untuk manajemen risiko; terakhir merupakan saran untuk penelitian berikutnya.

## 2. Proses Manajemen Risiko

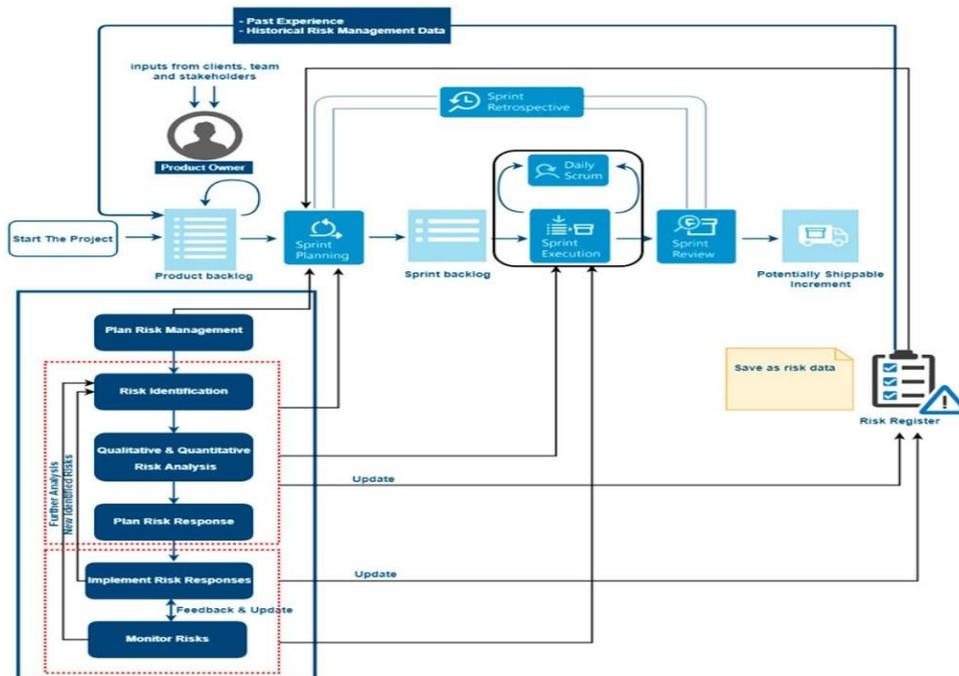
Menurut Bart Jutte manajemen risiko proyek adalah desain sistematis, implementasi, dan pemantauan tindakan untuk mengidentifikasi, memprioritaskan, dan menganalisis risiko proyek, dan untuk merancang, memilih, serta menerapkan tanggapan untuk mengoptimalkan risiko (Jutte, 2012). Konsep dari definisi tersebut dijelaskan sebagai berikut:

- **Sistematis:** Manajemen risiko proyek adalah metode terstruktur untuk menangani risiko, termasuk tanggung jawab, prioritas, dan tugas yang jelas. Ini kontras dengan pendekatan *ad hoc* yang me-ngandalkan keberuntungan untuk berhasil.
- **Pemantauan Tindakan:** Melakukan tindakan adalah pusat manajemen risiko. Ini tentang kepala, tangan, dan mata yang memikirkan tugas, melaksanakannya, dan memantau jika tugas itu terwujud.
- **Mengidentifikasi:** Proses ini untuk menemukan risiko proyek yang mungkin ada. Risiko apa yang akan membentuk peluang atau ancaman bagi proyek.
- **Prioritaskan:** Mengurutkan risiko dalam urutan kepentingan. Ini memungkinkan

tim proyek untuk menangani risiko terbesar lebih dahulu.

- **Tanggapan:** Respon yang bisa diterapkan untuk mengubah profil risiko proyek.

Chaouch dkk mengusulkan kerangka kerja manajemen risiko yang terlihat pada Gambar 2. Model tersebut terdiri dari 6 fase yaitu: Perencanaan manajemen risiko; Identifikasi risiko; Analisa risiko secara kualitatif dan kuantitatif;



Gambar 2 Kerangka kerja manajemen risiko

Sumber: *A framework for risk management in Scrum development process* (Chaouch et al., 2019)

Perancangan penanganan risiko; Implementasi penanganan risiko; dan Pemantauan risiko (Chaouch et al., 2019).

Kerangka kerja Chaouch dkk terbilang menyeluruh, dimana tahap-tahap manajemen risiko dimasukkan ke dalam proses *Sprint Planning* dan *Sprint Execution*. Semua daftar risiko (*risk register*) dicatat dan disimpan, lalu dibuka kembali setiap fase *Sprint Planning* berikutnya dimulai.

Namun dalam kerangka kerja Syrine Chaouch beberapa langkah masih terlewat, langkah-langkah tersebut adalah: Memprioritaskan risiko (*Prioritize risks*); dan Pemilihan respon (*Select responses*). Sebagaimana yang dimaksud oleh Bart Jutte dalam

Gambar 3.



Gambar 3 Proses manajemen risiko

Sumber: *Project risk management handbook* (Jutte, 2012)

Chaouch dkk juga tidak menjelaskan mekanisme dan prosedur pencatatan daftar risiko dan bagaimana para pengembang (*developer*) dan pemangku kepentingan dapat mengakses daftar risiko tersebut.

Sebagai usulan tambahan pada makalah ini, dikategorikan juga tipe-tipe respon yang dapat diambil guna menurunkan risiko atau ancaman negatif terhadap proyek, respon tersebut diantaranya: *risk avoidance* (menghindari risiko), *risk transfer* (memindahkan risiko), *risk mitigation* (mengurangi risiko), *risk monitoring* (memantau risiko), dan *risk acceptance* (menerima risiko), lalu apakah diperlukan rencana cadangan, dan agenda strategis (Pandian, 2006).

### 3. Adaptasi Pilar Scrum dalam Manajemen Risiko

Scrum menyatukan empat agenda formal untuk peninjauan dan penyesuaian yang diakomodasi dalam suatu acara, yaitu Sprint. Hal tersebut akan berhasil karena menerapkan pilar Scrum empiris yaitu transparansi, inspeksi, dan adaptasi (Schwaber Ken & Sutherland Jeff, 2020).

Penyesuaian ketiga pilar Scrum ini ke dalam struktur manajemen risiko suatu proyek, bisa membuat pengendalian risiko yang lebih efektif. Transparansi memerlukan komitmen dan kerelaan untuk membuka informasi secara penuh dan benar. Sedikitnya ada dua bagian yang perlu dibuat transparan dalam struktur manajemen risiko, yaitu pencapaian target yang diketahui oleh pemilik risiko (*risk owner*) dan *progress* pelaksanaan dalam unit kerja (Kurniawan, 2020).

*Visual management tools* dapat digunakan dalam mewujudkan prinsip-prinsip transparansi seperti data yang perlu diperbaharui agar mampu memberikan informasi sekini mungkin, dan *tools* yang dapat diakses oleh tim Scrum kapanpun dan dimanapun. Menginspeksi risiko-risiko yang ada pada daftar risiko (*risk register*) dengan terus-menerus memeriksa status dan perubahan yang terjadi pada lingkungan eksternal ataupun internal. Melakukan adaptasi dengan cara mengaktifkan langkah-langkah mitigasi yang telah disusun sebelumnya dan terus memperbaharui daftar risiko sesuai dengan perkembangan terkini.

Guna mengoptimalkan adaptasi pilar Scrum dalam manajemen risiko, maka ditambahkan *dataset* dan *machine learning* sebagai *tools* ke dalam kerangka kerja manajemen risiko yang diusung oleh Chaouch dkk, seperti terlihat pada Gambar 4.

### 4. Usulan Tools untuk Kerangka Kerja Manajemen Risiko

Selesainya daftar risiko (*risk register*) sering dianggap sebagai selesainya tugas pengelolaan risiko. Pada banyak organisasi *risk register* hanya bersifat musiman, sehingga *risk owner* tidak begitu mengerti apa yang mesti dijalankan setelah daftar risiko disusun.

Risiko yang diidentifikasi dan disimpan dalam *risk register* bisa saja berupa risiko-risiko yang sudah pernah dialami oleh tim Scrum pada proyek-proyek sebelumnya. Sehingga terjadi pengulangan pembahasan analisa dan perencanaan penanganan pada setiap fase Sprint. Hal tersebut

dapat juga menimbulkan beban tersendiri bagi *risk owner* dan *Product Owner* yang baru (berbeda dari proyek sebelumnya) dalam pengambilan keputusan atau respon terhadap risiko yang telah diidentifikasi.

Beberapa hasil para peneliti terdahulu yang berkaitan dengan kerangka kerja manajemen risiko dan *tools* manajemen risiko dalam proses pengembangan perangkat lunak, diantaranya sebagai berikut.

García dkk menyajikan hasil studi kasus penerapan *tools* dalam manajemen risiko dasar di dua perusahaan. Hasil hipotesa menunjukkan bahwa penggunaan *tools* ini berguna untuk diimplementasikan pada proyek rekayasa perangkat lunak (García et al., 2018).

Puri dan Sharma mengangkat isu-isu dalam manajemen risiko dalam rekayasa perangkat lunak menggunakan *big data*. Analisis prediktif *big data* digunakan untuk membuat prediksi risiko yang pernah dialami sebelumnya dalam proyek perangkat lunak dan memberikan proposal kemungkinan risiko yang akan timbul dengan sesuai (Puri & Sharma, 2020).

Aslam dkk mengusulkan *tools* yang dapat membantu dalam pengambil keputusan selama manajemen risiko DSD (*Distributed Software Development*). Namun demikian usulan *tools* ini belumlah mengkaitkan berbagai tahap perencanaan dengan identifikasi variasi dalam keluaran DSS (*Decision Support Systems*) pada berbagai tahap proyek. Sehingga masih diperlukan penambahan fitur dalam pengembangan terdistribusi pada hasil penilaian risiko (Aslam et al., 2017).

Filippetto dkk menggunakan model Atropos untuk mengukur ketidakpastian dalam proyek dengan nilai yang mendekati dampak keuangan aktual dari risiko yang telah diidentifikasi. Penerapan rekomendasi risiko berdasarkan analisis kesamaan histori konteks dengan memberikan rekomendasi dan mempertimbangkan karakteristik setiap proyek baru (Filippetto et al., 2021).

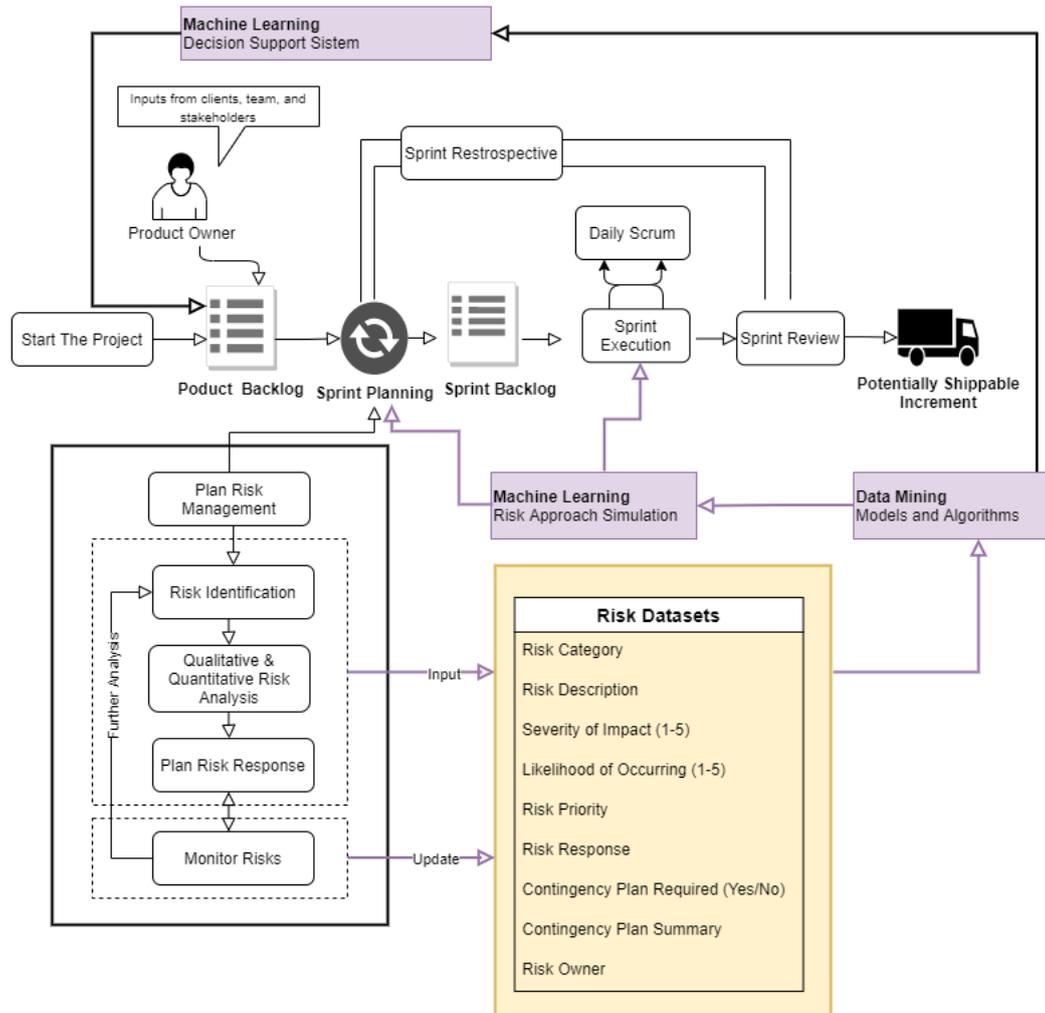
#### 4.1. Penerapan Kerangka Kerja Manajemen Risiko

Pemanfaatan *dataset* dan *machine learning* semakin marak di segala bidang dan sub-sub bidang, termasuk diantaranya dalam metodologi pengembangan perangkat lunak. Namun demikian, hingga makalah ini ditulis, belum ada penggunaan *dataset* dan *machine learning* yang digunakan

dalam kerangka kerja manajemen risiko proyek rekayasa perangkat lunak Scrum.

Pada Gambar 4 makalah ini mengusulkan pendekatan *machine learning* untuk menggali pengetahuan, memprediksi, mengembangkan algoritma berdasarkan data empiris, maupun mengenal pola masalah kompleks yang dihadapi,

sehingga dapat memodelkan aturan-aturan. Di sini *machine learning* akan membantu para *risk owner* dalam pengklasifikasian respon yang akan diambil, apakah itu *risk avoidance*, *risk transfer*, *risk acceptance*, *risk monitoring*, dan *risk mitigation*. Lalu apakah diperlukan *contingency plan*, dan *strategic plan* (Pandian, 2006).



Gambar 4 Usulan kerangka kerja manajemen risiko

*Machine learning* juga membantu *Product Owner* dalam penyusunan *Product Goal* berdasarkan dari pengalaman-pengalaman pengembangan perangkat lunak dari proyek yang sedang berjalan maupun dari proyek-proyek yang pernah dilalui oleh tim Scrum. *Dataset* merupakan kumpulan dari daftar risiko yang didapat dari proses asesmen risiko (identifikasi, analisa, perencanaan, respon dan implementasi) sepanjang tim Scrum melakukan pengembangan perangkat lunak.

Seperti yang telah dijelaskan pada bagian sebelumnya, bahwasanya Scrum adalah proses iteratif, maka diusulkan proses manajemen risiko untuk menjadi berulang juga. Dianjurkan gaya *brainstorming* untuk mengidentifikasi risiko. Tim Scrum duduk bersama dan mengidentifikasi risiko berdasarkan pengalaman masa lalu mereka dengan proyek sejenis atau tetap memperhatikan gangguan teknis apa pun. *Product Owner* harus hadir dalam semua pertemuan yang terkait dengan manajemen risiko (Hammad & Inayat, 2019). Dalam hal pengembangan produk *in-house* atau bahkan dalam

pengembangan produk berbasis *outsourcing* (vendor). *Product Owner* adalah orang yang membuat daftar fitur (*backlog*), sehingga kehadiran *Product Owner* dianggap wajib untuk rapat identifikasi risiko seperti pada rapat *Sprint Planning* dan *Sprint Review*. Oleh karena itu, identifikasi risiko awal dilakukan dalam setiap rapat *Sprint Planning*. Risiko yang teridentifikasi kemudian ditempatkan ke dalam artefak baru yang disebut daftar risiko (*risk register*). Kemudian, penilaian risiko dilakukan dalam pertemuan retrospektif *Sprint*. Pertemuan ini membahas *review* klien pada *Sprint* sebelumnya. Analisis risiko dilakukan dalam rapat-rapat tersebut. Dampak risiko (*severity of impact*) dan probabilitas risiko (*likelihood of occurring*) juga diputuskan berdasarkan pengalaman tim sebelumnya dan konsultasi dengan pendekatan respon atau *suggestion* yang dihasilkan oleh *machine learning* dari *mining* risiko-risiko sebelumnya. Kemudian rencana mitigasi risiko diputuskan untuk setiap risiko yang teridentifikasi.

Alur proses manajemen risiko yang berulang dengan proses Scrum dapat dilihat seperti pada Gambar 4.

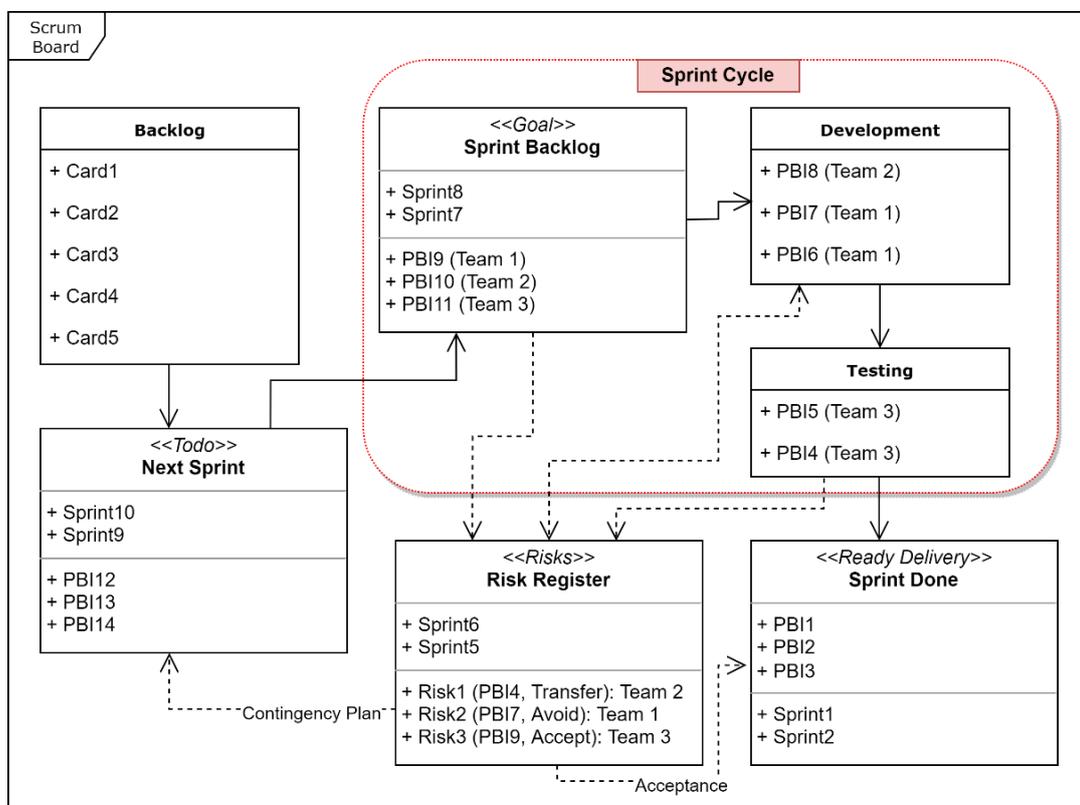
#### 4.2. Tools Manajemen Risiko

Makalah ini mengusung aplikasi Trello sebagai pendukung kerangka kerja yang diusulkan. Trello merupakan aplikasi kolaborasi yang mengawasi berbagai aktifitas dan kegiatan di satu platform. Dengan aplikasi ini, semua kelompok yang termasuk dalam proyek akan mengetahui apa saja yang sedang dikerjakan, siapa yang mengerjakannya, dan kemajuan pekerjaan itu. Trello seperti papan tulis yang diisi dengan catatan yang ditempel (*sticky note*). Dimana pada setiap catatan tempel tersusun secara lengkap tentang tugas tim secara keseluruhan. Setiap catatan tempel juga dilengkapi dengan gambar, berkas atau file laporan dari bermacam sumber, serta ruang untuk komentar.

Diusulkan juga agar *Boards* yang ada dalam Trello disesuaikan pemodelan dan alur prosesnya seperti terlihat pada Gambar 5.

#### 4.3. Struktur Dataset Risiko

*Raw data* atau data mentah bersumber dari Trello sebuah *tools* manajemen proyek yang digunakan selama proses pengembangan perangkat lunak.-



Gambar 5 Pemodelan Scrum Board dalam aplikasi Trello

Trello memiliki fitur ekspor yang disediakan secara gratis dengan format JSON (*Javascript Object Notation*). Risiko disimpan dalam sebuah skema database yang terdiri dari hal-hal berikut:

- *Risk Category*, yaitu pengklasifikasian risiko, seperti risiko operasional, risiko dalam budget, risiko dalam tengat waktu, risiko lingkungan teknis, risiko bisnis, risiko teknologi dan lainnya (Hammad et al., 2019).
- *Risk Description*, yaitu penjelasan dan penjabaran tentang risiko yang timbul.
- *Severity of Impact*, seberapa parahkan risiko ini akan berdampak, skor diisi dari 1 yang terkecil sampai 5 yang terbesar, nilai tersebut mewakili: 1 → Sangat Rendah; 2 → Rendah; 3 → Sedang; 4 → Tinggi; dan 5 → Sangat Tinggi.
- *Likelihood of Occurring*, probabilitas atau seberapa besar ukuran atau frekuensi kemungkinan risiko akan terjadi dalam setiap proyek, nilai mutu yang diisi berupa: 1 → Sangat Jarang; 2 → Jarang; 3 → Mungkin; 4 → Sangat Mungkin; dan 5 → Tertentu.
- *Risk Priority*, level risiko dibagi menjadi 4, yaitu: Tinggi; Sedang; Rendah; dan Sangat Rendah.
- *Risk Response*, respon terhadap risiko antara lain: menghindari risiko, memindahkan risiko, mengurangi risiko, memantau risiko, dan menerima risiko, lalu apakah diperlukan rencana cadangan, dan agenda strategis (Pandian, 2006).
- *Contingency Plan Required*, diisi dengan YES atau NO.
- *Contingency Plan Summary*, merupakan ringkasan dari perencanaan yang akan diambil berkaitan dengan risiko yang timbul.
- *Risk Owner*, pemilik risiko dapat berupa tim atau individu, hal ini tergantung sejauh mana risiko tersebut berdampak.

## 5. Penelitian Akan Datang

Penelitian lanjutan masih diperlukan untuk menguji efektivitas dari kerangka kerja manajemen risiko yang penulis usung. Disarankan menggunakan metodologi studi kasus dengan pendekatan kualitatif untuk menganalisa apakah pengintegrasian kerangka kerja manajemen risiko ke dalam alur proses Scrum dapat meningkatkan

keberhasilan proyek, khususnya untuk proyek pengembangan perangkat lunak (*software*).

## References

- Aslam, A., Ahmad, N., Saba, T., Almazyad, A. S., Rehman, A., Anjum, A., & Khan, A. (2017). Decision Support System for Risk Assessment and Management Strategies in Distributed Software Development. *IEEE Access*, 5, 20349–20373. <https://doi.org/10.1109/ACCESS.2017.2757605>
- 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>
- Dey, P. K., Kinch, J., & Ogunlana, S. O. (2007). Managing risk in software development projects: a case study. *Industrial Management & Data Systems*.
- Filippetto, A. S., Lima, R., & Barbosa, J. L. V. (2021). A risk prediction model for software project management based on similarity analysis of context histories. *Information and Software Technology*, 131. <https://doi.org/10.1016/j.infsof.2020.106497>
- García, Y. M., Muñoz, M., Mejía, J., Gasca, G. P., & Mireles, A. (2018). Application of a risk management tool focused on helping to small and medium enterprises implementing the best practices in software development projects. *Advances in Intel*, 746, 429–440. [https://doi.org/10.1007/978-3-319-77712-2\\_41](https://doi.org/10.1007/978-3-319-77712-2_41)
- Hammad, M., & Inayat, I. (2019). Integrating risk management in scrum framework. *Proceedings - 2018 International Conference on Frontiers of Information Technology, FIT 2018*, 158–163. <https://doi.org/10.1109/FIT.2018.00035>
- Hammad, M., Inayat, I., & Zahid, M. (2019). Risk management in agile software development: A survey. *Proceedings - 2019 International Conference on Frontiers of Information Technology, FIT 2019*, 162–166. <https://doi.org/10.1109/FIT47737.2019.00039>
- Jutte, B. (2012). *Project Risk Management Handbook: the invaluable guide for managing project risks*. Xlibris Corporation.
- Kurniawan, J. (2020). *Kelola Risiko dengan Scrum Pillar: Transparansi, Inspeksi, dan Adaptasi*. In *buntrack.co.id* (p. 1).
- Moran, A. (2014). *Agile risk management*. In *Agile Risk Management* (pp. 33–60). Springer.
- Nyford, J. (2008). *Towards integrating agile development and risk management*.

- Pandian, C. R. (2006). Applied software risk management: A guide for software project managers. Auerbach Publications.
- Puri, A., & Sharma, S. (2020). Risk Management in Software Engineering Using Big Data. *Proceedings of International Conference on Intelligent Engineering and Management, ICIEM 2020*, 63–68.  
<https://doi.org/10.1109/ICIEM48762.2020.9160170>
- Rudy, T. (2016). Manajemen Proyek Sistem Informasi, bagaimana mengolah proyek sistem informasi secara efektif & efisien. *Andi Offset: Yogyakarta*.
- Schwaber Ken, & Sutherland Jeff. (2020). Panduan Definitif untuk Scrum: Aturan Permainan. *Scrum.Org, November*, 1–17.