

BAB 1

PENDAHULUAN

1.1 Latar Belakang

Menurut Kamus Besar Bahasa Indonesia (KBBI), risiko adalah akibat yang kurang menyenangkan (merugikan, membahayakan) dari suatu perbuatan atau tindakan. Risiko merupakan keadaan yang tidak pasti dan mempunyai dampak negatif terhadap suatu tujuan yang ingin dicapai (ISO31000, 2018). Risiko mengacu pada kemungkinan kerugian ekonomi atau keuangan, cedera fisik atau kerusakan atau keterlambatan, yang dihasilkan dari ketidakpastian terkait dengan pekerjaan tertentu saat ini (Zavvar et al., 2017).

Proyek adalah upaya sementara yang dilakukan untuk menciptakan produk, layanan, atau hasil yang unik (Demir, 2017). Manajemen proyek menurut *Project Management Body of Knowledge* (PMBOK) adalah aplikasi dari pengetahuan, keahlian, alat dan teknik untuk melaksanakan aktivitas sesuai dengan kebutuhan proyek (Naveed et al., 2017). Manajemen risiko proyek adalah mengidentifikasi, menganalisis, dan menanggapi setiap risiko yang muncul selama siklus hidup proyek untuk membantu proyek tetap pada jalurnya dan memenuhi tujuannya (PMI, 2017). Risiko merupakan bagian integral dari setiap fase proyek dan manajemen risiko merupakan bagian penting dari proses pengambilan keputusan di setiap tahapan proyek. Oleh karena itu, manajemen risiko tidak hanya reaktif, tetapi harus menjadi bagian dari proses perencanaan untuk mengetahui risiko yang mungkin terjadi dalam proyek dan bagaimana mengendalikan risiko itu jika terjadi. Dengan melaksanakan manajemen risiko, sehingga dapat membantu organisasi dengan menghemat pengeluaran sambil mengamankan apa yang akan datang. Karena manajemen risiko yang tepat akan membantu organisasi dengan membangun strategi untuk menghindari risiko, membatasi konsekuensi yang merugikan, dan mengontrol risiko tersebut. Kapasitas untuk memahami dan mengendalikan risiko membuat organisasi lebih positif dalam menentukan pilihan bisnis.

Keberhasilan atau kegagalan suatu proyek sangat bergantung pada pendekatan potensi munculnya risiko yang dapat mempengaruhi produktivitas,

kualitas, ketepatan waktu dan atau biaya proyek (Crnković & Vukomanović, 2016). Joshua Partogi yang merupakan pelatih Scrum Profesional mengatakan bahwa pekerjaan tambahan yang menyebabkan biaya *software development* menjadi lebih mahal merupakan risiko yang dapat dihilangkan dalam lingkungan yang aman untuk gagal (Partogi, 2015).

Umumnya, risiko utama dalam proyek dengan metode tradisional seperti Waterfall adalah kegagalan dalam memenuhi kriteria batasan (*scope*) proyek (Rudy, 2016). Perkembangan manajemen risiko dalam metodologi rekayasa perangkat lunak, bermula ketika manajemen risiko diakui sebagai bidang penelitian independen pada tahun 1989. Model Spiral merupakan metode siklus pengembangan yang pertama kali menggabungkan manajemen risiko secara eksplisit di dalamnya (Boehm, 1991). Namun, 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 menggunakan pendekatan iteratif dan inkremental untuk mengoptimalkan prediktabilitas dan mengendalikan risiko (Schwaber & Sutherland, 2017). Sehingga para profesional Scrum tidak sepenuhnya menyadari risiko, walaupun tetap mengikuti proses manajemen risiko secara implisit. Mereka berpikir bahwa siklus hidup pengembangan telah meminimalkan efek tak terduga dari pengembangan perangkat lunak (Chaouch et al., 2019).

Kerap kali dalam proyek pengembangan perangkat lunak, banyak waktu dihabiskan untuk menangani dan memperbaiki *bug* (Lam et al., 2016). Padahal sebelum ini terjadi, *developer* telah mengidentifikasi dan mencatat risiko dalam daftar risiko (*risk register*) (Chaouch et al., 2019; Hammad & Inayat, 2018; Wan Husin et al., 2019). Namun, dalam memprioritaskan risiko sering kali memakan waktu yang lama, karena para pemangku kepentingan (*stakeholder*) non-teknis

seperti *Project Manager* (PM) dan *Product Owner* (PO) kurang memahami konteks risiko yang telah ditulis oleh tim pengembang (M. S. A. Lee & Singh, 2021; Mahfoodh & Obediat, 2020). Oleh karena itu, diperlukan suatu sistem pendukung keputusan untuk membantu mengklasifikasikan risiko-risiko yang termasuk dalam temuan atau laporan. Pendekatan otomatis harus diusulkan untuk membantu *stakeholder* dalam pengambilan respon terhadap risiko. Pendekatan dapat dibagi menjadi dua variabel yang saling melengkapi (Finger et al., 2021; Paltrinieri et al., 2019; Qazi et al., 2021; Xu et al., 2018):

- 1) Pendekatan pertama adalah klasifikasi risiko berdasarkan kemungkinan terjadinya (*likelihood*).
- 2) Pendekatan kedua menganalisis isi dari risiko yang diberikan untuk menentukan seberapa parah risiko tersebut akan berdampak (*severity*).

Kombinasi dari variabel-variabel tersebut membentuk matriks risiko, yang telah banyak digunakan untuk penentuan prioritas risiko dan sudah menjadi standar internasional (Bahrudin, 2016; Wiradarma & Sasmita, 2019). Matriks risiko umumnya dinilai menggunakan skala Likert untuk menentukan peringkat probabilitas dan dampak risiko (Aven, 2017). Seperti yang terlihat pada Gambar 1, semakin tinggi nilai risiko, semakin tinggi pula prioritas terhadap risiko tersebut.

Probability	Risk Magnitude				
Certain 5	5	10	15	20	25
Very Likely 4	4	8	12	16	20
Likely 3	3	6	9	12	15
Unlikely 2	2	4	6	8	10
Very Unlikely 1	1	2	3	4	5
Cost / Time Impact	Very Low 1	Low 2	Medium 3	High 4	Very High 5

Gambar 1. Matriks prioritas risiko (sumber: Hammad & Inayat, 2018)

Namun, pada banyak organisasi *risk register* hanya bersifat musiman dan sering dianggap sebagai selesainya tugas pengelolaan risiko. Sehingga pemilik risiko (*risk owner*) kurang memahami apa yang harus dilakukan setelah *risk*

register tersusun. Daftar risiko bisa saja berupa kejadian-kejadian yang sudah pernah dilalui pada proyek yang lain atau sudah diidentifikasi oleh para pengembang lain dari seluruh dunia pada proyek *open source* di repositori GitHub.com, hanya saja ditulis dengan diksi yang berbeda. Untuk itulah dilakukan penelitian agar menghasilkan model yang dapat memprediksi besarnya risiko dengan memanfaatkan *machine learning* pada pengolahan teks yang berasal dari daftar risiko.

Solusi perhitungan risiko non-kuantitatif dengan penggunaan teknik pengklasifikasi Naive Bayes, Jaringan Syaraf Tiruan (Naseem et al., 2021; Shivhare & Rath, 2014), klasifikasi *development issues* dengan Support Vector Machine (SVM) (Beyer et al., 2018; Iwata et al., 2017; Tripathy et al., 2017; Zavvar et al., 2017), deteksi polaritas sentimen untuk pengembangan perangkat lunak (Calefato et al., 2018), Lexicon-Based yang digunakan untuk analisis sentimen pada proses pengembangan perangkat lunak (Jurado & Rodriguez, 2015), dan menganalisis opini *developer* dari postingan GitHub dengan BERT (Batra et al., 2021). Segmentasi dampak risiko dari banyak proyek yang dikembangkan di berbagai industri perangkat lunak menggunakan K-Means (Naseem et al., 2021; Suma et al., 2012), serta mekanisme pembelajaran mesin berbasis fuzzy hybrid untuk melakukan penilaian risiko dalam proyek perangkat lunak (Sangaiah et al., 2018; Suresh & Dillibabu, 2020), telah dilakukan oleh banyak peneliti sebelumnya. Namun hingga saat ini, belum ditemukan penelitian yang mengkombinasikan antara penentuan probabilitas terjadinya risiko, analisis polaritas sentimen, dan klasifikasi prioritas berbasis matriks risiko dengan model *machine learning*.

Selain itu, dengan mengintegrasikan model klasifikasi risiko ke dalam kerangka kerja manajemen risiko dapat membantu pemilik risiko dalam penentuan prioritas risiko secara terjadwal. Selanjutnya, agar mendapatkan model manajemen risiko yang optimal dalam pengembangan perangkat lunak, diperlukan analisis dan perbandingan berbagai metodologi dan model manajemen risiko. Fokus yang utama pada model yang paling populer dan sering digunakan seperti model manajemen risiko dari PMBOK dan ISO 31000:2018, dan juga model-model dari penelitian terdahulu yang telah diterapkan pada kerangka kerja Scrum.

Sehubungan dengan permasalahan diatas, maka penelitian ini mengangkat topik tentang kerangka kerja manajemen risiko di Scrum dengan klasifikasi prioritas risiko berbasis matriks risiko dengan *machine learning*, dimana penentuan kemungkinan terjadinya (*likelihood of occurring*) dan prediksi keparahan dampak (*severity of impact*) menggunakan teknik pengklasteran dan deteksi polaritas sentimen para *developer*.

1.2 Identifikasi Masalah

Berdasarkan uraian latar belakang yang dijelaskan di atas, maka identifikasi masalah pada penelitian ini adalah sebagai berikut:

- a. Proses memprioritaskan risiko sering kali memakan waktu yang lama, karena para pemangku kepentingan kurang memahami konteks dari daftar risiko yang telah ditulis oleh tim pengembang. Dengan adanya model klasifikasi *machine learning*, proses penentuan prioritas dari sebuah matriks risiko menjadi lebih cepat.
- b. Diperlukan algoritma klasifikasi teks dari daftar risiko dengan hasil pengukuran metrik evaluasi yang terbaik, agar model yang diintegrasikan ke dalam kerangka kerja dapat berjalan dengan optimal.
- c. Pada saat ini para profesional Scrum menjalankan manajemen risiko masih secara implisit, yang menyebabkan penanganan risiko belum terjadwal dalam setiap *event* Scrum. Dengan adanya kerangka kerja, aturan-aturan manajemen risiko dapat dijalankan secara transparan dan termonitor.

1.3 Perumusan Masalah

Berdasarkan latar belakang dan identifikasi masalah yang telah diuraikan, dapat diambil beberapa rumusan masalah atau *Research Question* (RQ) sebagai berikut:

- RQ1. Bagaimana proses manajemen risiko dalam pengembangan perangkat lunak yang menggunakan metodologi Scrum?
- RQ2. Bagaimana cara menentukan prioritas risiko dengan model *machine learning* sebagai sistem pendukung keputusan?

RQ3. Apakah dengan menggunakan kerangka kerja manajemen risiko dapat mendukung proyek selesai tepat waktu dan sesuai dengan biaya yang dianggarkan?

1.4 Batasan Penelitian

Untuk menghindari meluasnya pokok bahasan, penelitian ini dibatasi untuk hal-hal berikut:

- a. Lingkungan sistem (*environment*) untuk pembentukan model ML (*Machine Learning*) menggunakan bahasa pemrograman Python di Kaggle.com¹ versi non-berbayar, dengan 2 Core CPU, 13 GB RAM, Tesla P100 GPU, dan 9 jam maksimum waktu untuk eksekusi.
- b. *Dataset* yang digunakan pembentukan model sebanyak 803.417 baris, yang merupakan data sekunder hasil dari penelitian terdahulu (Siddiq & Santos, 2022). Data tersebut merupakan kumpulan GitHub Issue yang diambil dari Windows.net². Data yang diproses untuk pemodelan hanya yang berbahasa Inggris, senada dengan data primer yang ditulis dengan bahasa Inggris juga.
- c. Untuk membuat model ML menggunakan metode *Natural Language Processing* (NLP), algoritma untuk pengklasteran yang diperbandingkan adalah K-Means dan Fuzzy C-Means yang merupakan algoritma untuk *clustering* yang populer (Anam et al., 2019; Suyal et al., 2014; Weißer et al., 2020). Sedangkan algoritma untuk klasifikasi teks yang diperbandingkan adalah Naïve Bayes Multinomial (NBM), Support Vector Machine (SVM), Long Short-Term Memory (LSTM) Neural Network, dan Bidirectional Encoder Representations from Transformers (BERT) Neural Network, karena sering digunakan untuk memodelkan klasifikasi risiko dalam proyek pengembangan perangkat lunak (Beyer et al., 2018; Kumar et al., 2016; Tripathy et al., 2017; Zavvar et al., 2017).
- d. Pengukuran performa algoritma NLP yang diperbandingkan antara lain meliputi akurasi *training* data, akurasi *test* data, dan waktu yang dibutuhkan untuk menghasilkan model ML (Iftikhar et al., 2021; Lin et al., 2018; Nassif

¹ <https://www.kaggle.com/code/marzukipilliang/803k-issues-keras-classify-risk>

² <https://tickettagger.blob.core.windows.net/datasets/github-labels-top3-803k-train.tar.gz>

- et al., 2021). Dan juga *Precision* dan *Recall* untuk mencari nilai *F-Score* atau yang sering juga di sebut *F1-Score* (Derczynski, 2016).
- e. Matriks risiko yang digunakan adalah skala 5x5 dengan 4 (empat) level prioritas risiko: 1 (*Small*); 2 (*Medium*); 3 (*High*); 4 (*Extreme*). Mengacu pada penelitian terdahulu (Hammad & Inayat, 2018).
 - f. Metodologi pengembangan perangkat lunak yang digunakan dalam studi kasus adalah Scrum dengan *life-cycle* yang diteliti sebanyak 5 (lima) Sprint atau 70 hari kalender, berdasarkan *project milestone* dari *Term of Reference* (TOR) di LAMPIRAN A.
 - g. *Tools* manajemen proyek yang digunakan adalah Trello[®] versi non berbayar. Seperti terlihat pada LAMPIRAN B, *Board* yang digunakan terdiri dari beberapa *List Card* diantaranya adalah:
 - *Backlog*, merupakan medium untuk mendaftarkan *Product Backlog Item* (PBI) dan *Change Request* (CR) yang dilakukan oleh PO.
 - *User Stories*, yang merupakan penjelasan fitur terperinci yang ditulis oleh PM dalam kalimat sederhana dari sudut pandang pengguna produk, baik menggunakan bahasa Indonesia maupun bahasa Inggris.
 - *Sprint Backlog*, merupakan daftar item yang harus diselesaikan oleh tim pengembang dalam satu siklus Sprint.
 - *Doing*, yang merupakan daftar item yang sedang dikerjakan oleh tim pengembang pada *event* Sprint berjalan.
 - *Risk Register*, merupakan daftar risiko hasil identifikasi *stakeholder* (PO, PM dan tim pengembang) selama proses pengembangan perangkat lunak, dari fase Sprint pertama hingga fase Sprint terakhir, ditulis menggunakan bahasa Inggris. Daftar tersebut merupakan data primer yang digunakan untuk validasi model ML.
 - *Testing*, merupakan daftar item yang sedang dalam masa pengujian kelayakan dan kesesuaian.
 - *Done*, merupakan daftar item yang telah selesai di uji dan di *review* oleh *user* pada *event* Sprint Review.
 - h. Google Colaboratory[®] sebagai *staging environment* digunakan untuk menjembatani antara model ML hasil Kaggle dengan data JSON hasil

eksport Trello[®]. Seperti terlihat pada LAMPIRAN C, proses klasifikasi prioritas risiko terhadap data primer (daftar risiko studi kasus) dilakukan disana.

- i. Aspek-aspek yang dievaluasi dalam studi kasus adalah *Cost Performance Index* (CPI) dan *Schedule Performance Index* (SPI), yang merupakan parameter keberhasilan model dan kerangka kerja di Scrum (Prasetya et al., 2021).

1.5 Tujuan Penelitian

Adapun tujuan dari penelitian ini adalah sebagai berikut:

- 1) Menganalisis model klasifikasi teks untuk mendapatkan nilai metrik evaluasi terbaik dengan waktu pembentukan yang singkat.
- 2) Mengintegrasikan model *machine learning* ke dalam kerangka kerja manajemen risiko di Scrum.
- 3) Mengevaluasi CPI dan SPI sebagai tolok ukur keberhasilan kerangka kerja manajemen risiko di dalam proyek.

1.6 Manfaat Penelitian

Dengan penelitian ini, dapat diketahui bahwa kerangka kerja manajemen risiko yang terintegrasi dengan model *machine learning* mampu mendukung proyek pengembangan perangkat lunak selesai tepat waktu dengan biaya yang sesuai anggaran.

1.7 Sistematika Penulisan

Secara garis besar, Tesis ini terdiri dari 6 (enam) BAB sebagai berikut:

BAB 1 – Pendahuluan

Menjelaskan latar belakang penelitian dengan mengobservasi metode-metode pengembangan perangkat lunak, identifikasi risiko proyek rekayasa perangkat lunak sebagai masalah penelitian, merumuskan penanganan risiko di Scrum sebagai masalah penelitian, dan mengangkat integrasi model *machine learning* ke dalam kerangka kerja manajemen risiko sebagai tujuan dan manfaat penelitian.

BAB 2 – Landasan Teori

Pemaparan teori-teori yang relevan dengan *text mining*, manajemen risiko di pengembangan perangkat lunak, metode dan hasil dari para peneliti sebelumnya tentang pemanfaatan *machine learning* untuk proyek rekayasa perangkat lunak dari tahun 2016 sampai dengan tahun 2022, dan perbandingannya dengan studi yang dilakukan oleh penulis (*State-of-The-Art*).

BAB 3 – Metodologi Penelitian

Menjelaskan tentang alur penelitian, proses *mining* data sekunder yang bersumber dari GitHub Issues, pembentukan model dan kerangka kerja yang diusulkan, studi kasus implementasi model dan kerangka kerja untuk pengumpulan data primer, serta metode analisis dan evaluasi data primer.

BAB 4 – Hasil dan Pembahasan

Memaparkan metrik evaluasi yang didapat dari algoritma K-Means, FCM, NBM, SVM, LSTM dan BERT dalam pembentukan model *machine learning*. Hasil pengujian metrik evaluasi algoritma dan penjelasan dari model yang didapat.

BAB 5 – Kerangka Kerja Manajemen Risiko di Scrum

Penjelasan alur kerangka kerja manajemen risiko yang berisi aktifitas diagram yang menggambarkan alur kerja dan sistem kerja manajemen risiko di Scrum. Hasil evaluasi model klasifikasi terhadap data primer di studi kasus, serta hasil wawancara mendalam dan analisis CPI dan SPI sebagai tolok ukur keberhasilan penerapan kerangka kerja manajemen risiko

BAB 6 – Kesimpulan dan Saran

Bagian ini berisi tentang kesimpulan dari algoritma yang digunakan dalam pembentukan model, manfaat dari penerapan kerangka kerja manajemen risiko pada metodologi Scrum. Dan potensi pengembangan penelitian untuk masa yang akan datang.