

Identifikasi Risiko Program *Maintenance* dalam Pengelolaan Proyek Berbasis *Agile* Menggunakan Pohon Klasifikasi

<http://dx.doi.org/10.28932/jutisi.v7i1.3545>

Riwayat Artikel

Received: 21 Februari 2021 | Final Revision: 22 Maret 2021 | Accepted: 31 Maret 2021

Billyanto Hendrik^{#1}, Bernard Renaldy Suteja^{✉#2}

#Program Studi Magister Ilmu Komputer, Fakultas Teknologi Informasi

Universitas Kristen Maranatha

Jl. Prof. drg. Surya Sumantri No.65 Bandung

¹1879013@maranatha.ac.id

²bernard.rs@it.maranatha.edu

Abstract — Agile is a system development life cycle methodology that focuses on development interactions that involve the user with the development team led by the project manager as an intermediary between the client and the development team, with the project manager as the project leader, it is expected that this role can carry out project planning by making estimates and designing project. The worst thing that can happen if the application fails to meet client expectations is the additional development time called maintenance, this risk will create losses to the company even though maintenance is an additional service, but this risk tends to be negative because it can have a negative impact on the company and members of the development team. responsible for the project, the project manager must be able to identify risks earlier during the sprint, so in this study we will discuss the analysis and risk identification of maintenance programs in agile-based project management, as a research analyst method will use a classification tree to group them so that It can be found at the sprint stage how much risk has started to be made, so that the project manager can make corrections at the next sprint to reduce maintenance risk.

Keywords— Agile Project Management; Classification Tree; Maintenance Risk Factor; Random Tree

I. PENDAHULUAN

Agile adalah metodologi dalam pengembangan *software* dengan proses-proses kecil yang berulang-ulang atau yang sering disebut dengan iterasi, pada tahun 2017 Agile mendapat persentase sebanyak 71% pengguna sebagai pengganti sistem *development Waterfall* [1]. *Agile Software Development Life Cycle* (SDLC) ini berfokus pada interaksi dan pengguna aplikasi yang dibuat, sehingga aplikasi tersebut seharusnya dapat memenuhi kebutuhan pengguna

secara baik. Hasil akhir diharapkan akan jauh lebih sempurna dan tepat sasaran karena dalam proses pengembangan sudah dilakukan analisis dan komunikasi kepada pengguna.

Salah satu skenario terburuk yang dapat terjadi adalah saat proses *development* selesai, pengguna mengajukan *maintenance* program karena ada terdapat ketidakcocokan program yang mengakibatkan program tidak sesuai dengan kebutuhan pengguna. Di samping itu, faktor internal dapat memberikan dampak pada terjadinya *maintenance*. Proses *maintenance* adalah proses *development* yang tidak secara formal terdefinisi dalam metodologi Agile, karena proses tersebut sudah tergantikan pada *sprint review*, maka harus ada cara untuk melakukan identifikasi lebih dini agar risiko terjadinya *maintenance* dapat dihindari.

Maintenance akan dilakukan saat pengembangan sistem sudah memasuki tahap *production*. Pada tahap *maintenance* ini pihak pengembang sistem biasanya tidak akan menerima pemasukan terkait dengan biaya *development* karena tujuan dari *maintenance* untuk meningkatkan kepuasan pelanggan dan pemegang saham [2]. Oleh karena itu, pengawasan manajemen proyek perlu dipantau dengan baik. Manajer proyek memiliki tugas untuk mengawasi dan memimpin jalannya *development* dapat terarah kepada *trade-off triangle* yang sudah ditentukan oleh manajer proyek.

Manajer proyek perlu memiliki kemampuan memprediksi hasil akhir dari proyek, sehingga pada saat proses *development* berlangsung manajer proyek dapat melakukan negosiasi untuk melakukan perubahan pada *task backlog* pada *sprint* yang sedang berlangsung atau *sprint* yang akan datang sehingga pada saat *end sprint* maupun saat memasuki fase produksi tujuan dapat tercapai. Namun yang

terjadi di lapangan seringkali satu manajer proyek ditugaskan untuk mengelola lebih dari satu proyek, sehingga fokus pengelolaan akan terpecah, dan persentase terjadinya *maintenance* akan meningkat. Manajer proyek memerlukan alat bantu yang dapat melakukan identifikasi risiko *maintenance* lebih dini agar proyek yang sedang berjalan tidak terganggu oleh beban *maintenance*.

Dalam riset ini, pendekatan terstruktur dan sistematis terhadap *software risk management* dilakukan dengan metode empiris-kuantitatif. Data yang diteliti berasal dari kuesioner hasil masukan dari para manajer proyek. Kuesioner dibuat berdasarkan faktor-faktor risiko pemeliharaan perangkat lunak. Dalam penelitian ini, rumusan masalah utama yang hendak dijawab adalah bagaimana cara mengidentifikasi risiko terjadinya pemeliharaan program dalam pengembangan perangkat lunak berbasis Agile?. Berpijak pada rumusan masalah tersebut, maka tujuan utama penelitian ini untuk menghasilkan model identifikasi faktor-faktor penyebab terjadinya risiko pemeliharaan program. Dengan adanya model identifikasi tersebut diharapkan manajer proyek dapat mengantisipasi jalannya proses pengembangan dengan tindakan-tindakan yang mengurangi risiko pemeliharaan program.

II. KAJIAN LITERATUR

A. System Development Life Cycle

System Development Life Cycle (SDLC) adalah pendekatan sistematis pada saat proses pembuatan atau pengembangan perangkat lunak dalam batasan waktu yang spesifik dengan pengawasan atas kualitas, dan disertai penerapan siklus hidup sesuai metodologi dipilih. Para pengembang perangkat lunak mengerjakan serangkaian aktivitas yang dilakukan selama pengembangan sistem dan sering disebut sebagai siklus hidup pengembangan perangkat lunak [3]. Dengan SDLC proses *development* akan lebih terstruktur dan terdokumentasi. SDLC memiliki banyak metode alternatif yang cocok untuk pengerjaan perangkat lunak, salah satunya adalah metodologi Agile.

B. Agile

Agile adalah metodologi dalam pengembangan *software* dengan proses-proses kecil yang berulang-ulang atau yang sering disebut dengan iterasi yang memiliki proses pengumpulan kebutuhan, analisis, desain, kode, testing, dan *user feedback* [4]. SDLC ini berfokus pada interaksi dengan pengguna pada saat proses pengembangan berlangsung. Dengan demikian diharapkan bahwa hasil akhir dalam masa *development* akan jauh lebih sempurna dan tepat sasaran dibandingkan metode *waterfall* karena dalam proses *development* terdapat banyak pihak yang ikut ambil bagian dalam menyempurnakan aplikasi.

Dalam metodologi Agile terdapat tahap *user feedback* atau *user review* setelah berakhirnya *sprint*. Dengan

demikian risiko terjadinya *maintenance* seharusnya memiliki tingkat yang rendah karena pada tahap akhir *sprint*, tim pengembang akan melakukan perbaikan sehingga pada *sprint* berikutnya tidak ada lagi pekerjaan tambahan untuk memperbaiki, karena akan membuat *timeline* pengerjaan mengalami kemunduran. Oleh karena itu, pada setiap tahap *sprint* identifikasi risiko harus diketahui lebih cepat guna memastikan keberhasilan berlangsungnya proses *sprint* tersebut.

C. Manajemen risiko

Manajemen risiko adalah proses pengelolaan risiko-risiko organisasi terhadap ketidakpastian yang berkaitan dengan ancaman yang menyebabkan tumbuhnya risiko [5]. Tugas manajemen risiko adalah menetapkan risiko dan mengkategorikan risiko [6]. Dengan demikian, risiko akan dapat dikelola dengan lebih baik dan lebih dini agar kerugian dalam organisasi dikurangi. Pendekatan manajemen risiko pada penelitian ini akan diarahkan sebagai model untuk mengidentifikasi risiko yang kemungkinan muncul selama proses pengembangan perangkat lunak, sehingga jenis dan tingkat risiko akan tercatat serta dapat dihindari.

D. Klasifikasi data

Klasifikasi data adalah pengelompokan dan pengaturan data untuk proses analisis dan pengambilan keputusan yang lebih baik. Klasifikasi data dapat menjadi bagian dari program manajemen risiko [7]. Dengan menggunakan klasifikasi data dapat melakukan identifikasi *descriptive* dalam mempelajari tentang data yang diamati sehingga dapat digunakan untuk mengetahui karakteristik dari data, dan mengetahui fungsi bersifat *predictive* dalam menghitung pola pada komponen yang akan ditentukan. Pada penelitian ini klasifikasi data akan dilakukan untuk mengelompokkan data *sprint* menjadi tiga bagian.

E. Pohon keputusan

Pohon keputusan adalah sebuah struktur yang mengolah data dengan jumlah besar yang menghasilkan kumpulan data menjadi himpunan-himpunan record yang lebih kecil dengan memperhitungkan serangkaian aturan [8], kemampuan pohon keputusan dapat melakukan eksplorasi data dalam mencari hubungan antar komponen sehingga hasil akhirnya dapat diketahui sejumlah komponen yang memiliki tingkat hubungan paling kuat dan saling mempengaruhi, untuk melakukan pengujian maka dibutuhkan *data training*.

F. Random forest

Random forest adalah metode yang digunakan untuk melakukan klasifikasi dan regresi data dalam Pohon keputusan, metode ini merupakan gabungan dari masing-masing pohon keputusan sebagai pengelompokan data yang dibangun dan dikombinasikan [9]. *Training data* digunakan

sebagai masukan untuk pemrosesan algoritma *random forest* sedangkan testing akan digunakan sebagai pengujian atau evaluasi dari *output* model yang dihasilkan dari algoritma acak [10]

forest

III. METODOLOGI PENELITIAN

Dalam penelitian menggunakan metode kuantitatif, yang membantu dalam melakukan analisis dimulai dari data didapatkan dengan menggunakan kuesioner yang dirancang dengan pertanyaan yang dibuat berdasarkan kombinasi dari *risk management techniques* dan *software maintenance risk factors*, lalu ditentukannya klasifikasi data sebanyak tiga jenis berdasarkan pengelompokan *sprint timeline*, lalu dengan, dengan data yang sudah ada akan dilakukan *cleaning* data dengan *optimized forest* lalu dengan data yang sudah disaring maka dapat lah dilakukan pengolahan data

dengan *random tree*, untuk lebih detailnya akan dilakukan pembahasan seperti berikut.

A. Teknik Manajemen Risiko

Komponen teknik manajemen risiko yang digunakan juga muncul sebagai komponen pada faktor risiko pemeliharaan perangkat lunak sehingga perlu dibuat analisis cara membuat pertanyaan pada kuesioner yang dapat keduanya. Komponen teknik manajemen risiko yang digunakan dapat dilihat dalam Tabel I.

Dalam Tabel I dapat dilihat 15 komponen teknik manajemen risiko yang dianalisis dalam penelitian ini. Komponen-komponen tersebut meliputi tahapan-tahapan pengelolaan proyek perangkat lunak mulai dari tahap penentuan *requirements*, penjadwalan kerja, interaksi dengan pengguna, pengerjaan, pelatihan pengguna sampai akhirnya pada tahap evaluasi.

TABEL I

TEKNIK MANAJEMEN RISIKO

No	Kode	Teknik Manajemen Risiko	Penjelasan
1	C1	Penggunaan persyaratan <i>scrubbing</i> .	Metode pengujian proyek perangkat lunak dengan spesifikasi produk diuji cermat untuk identifikasi fitur yang tidak penting atau terlalu kompleks saat dilakukan pengurangan fitur [9].
2	C2	Menstabilkan persyaratan dan spesifikasi sedini mungkin.	Kunci untuk mengatur kebutuhan adalah melalui kemitraan yang dikembangkan dalam proyek perangkat lunak serta melakukan komunikasi dengan manajer fungsional untuk melakukan mentransfer pengetahuan bisnis ke tim proyek perangkat lunak dan berpartisipasi dalam desain proses dan persyaratan yang mendukung desain proses [12].
3	C3	Menilai biaya dan menjadwalkan dampak dari setiap perubahan pada persyaratan dan spesifikasi.	Risiko kegagalan perangkat lunak sangat positif terkait dengan pembengkakan anggaran dan jadwal [10].
4	C5	Mengembangkan dan mengikuti rencana proyek perangkat lunak.	Membuat perencanaan <i>project development software</i> untuk dijadikan patokan dalam pembiayaan dan estimasi [11].
5	C7	Mengembangkan rencana darurat untuk mengatasi masalah kepegawaian.	Memperhitungkan tindakan kontingensi yang dapat diambil jika proyek perangkat lunak berubah menjadi risiko kegagalan [12].
6	C8	Menugaskan tanggung jawab kepada anggota tim dan merotasi pekerjaan.	Menetapkan tanggung jawab dan peran yang jelas untuk anggota tim yang berkontribusi mengembangkan proyek perangkat lunak [13].
7	C9	Adakan sesi pembangunan tim.	Sesi pembangunan tim dilakukan oleh manajer proyek perangkat lunak di seluruh siklus hidup proyek perangkat lunak [14].
8	C11	Membagi proyek perangkat lunak menjadi beberapa bagian yang dapat dikontrol.	Manajer proyek perangkat lunak perlu memecah proyek perangkat lunak besar menjadi elemen kerja kecil tambahan untuk mengurangi risiko proyek perangkat lunak [12].
9	C12	Kode sumber dan metode antarmuka yang dapat digunakan kembali.	Menggunakan kode dan antarmuka yang dapat digunakan kembali pada jenis proyek serupa akan berguna mengurangi risiko [15] <i>error</i> dan <i>buffer</i> .
10	C14	<i>Database</i> yang dapat digunakan kembali dan struktur <i>data mining</i> .	Struktur <i>database</i> yang dapat digunakan kembali untuk mengurangi sebagian besar waktu yang dihabiskan untuk melakukan analisis identifikasi data, pengumpulan, pembersihan, dan pemrosesan data [16].
11	C20	Melibatkan manajemen selama seluruh siklus hidup proyek perangkat lunak.	Keterlibatan semua anggota dalam tim pengembangan perangkat lunak akan terjadi karena sifat proses kerja dan relasinya memerlukan lebih banyak keterlibatan manajemen [17].
12	C22	Memanfaatkan <i>change control board</i> dan <i>exercise quality change control practices</i> .	Dapat didefinisikan <i>change control board</i> sebagai set minimum pemangku kepentingan proyek yang perlu meninjau dan menyetujui. permintaan perubahan apa pun yang memengaruhi faktor keberhasilan kritis proyek

No	Kode	Teknik Manajemen Risiko	Penjelasan
			perangkat lunak [18].
13	C23	Mendidik pengguna tentang dampak perubahan selama proyek perangkat lunak.	Mendidik pengguna tentang teknologi perangkat lunak dalam proyek perangkat lunak penting untuk mengurangi risiko [19].
14	C24	Memastikan hasil faktor kualitas dan analisis tugas.	Memastikan kiriman berkualitas tinggi sesuai jadwal penting untuk mengurangi risiko dalam proyek perangkat lunak. Selanjutnya [20].
15	C26	<i>Incremental development</i> (menunda perubahan ke kenaikan selanjutnya).	<i>Increment development</i> adalah proses <i>development</i> yang dilakukan berdasarkan kebutuhan fitur yang ditentukan namun prosesnya berdasarkan pengembangan bukan berdasarkan pembangunan, tidak didasarkan pada ruang lingkup tertentu (subset persyaratan) tetapi didasarkan pada ukuran upaya untuk perbaikan [21].

B. Faktor Risiko Pemeliharaan Perangkat Lunak

Penerapan teknik manajemen risiko memungkinkan pendekatan terstruktur dan sistematis untuk mengelola risiko pada saat proses pengembangan perangkat lunak. Hal ini akan efektif juga untuk mendeteksi, mengevaluasi, dan memitigasi risiko. Fokus pada *s. maintenance risk factors* adalah mencari faktor risiko pemeliharaan perangkat lunak

sehingga dapat dimitigasi dengan teknik manajemen risiko yang sudah dibahas pada Tabel I.

Pada penelitian ini dipilih sepuluh *software maintenance risk factors* [22]. Dengan ke sepuluh *software maintenance risk factors* yang sudah ditentukan tersebut maka dapat ditentukan juga *risk management techniques* yang cocok dalam melakukan identifikasi *software maintenance risk factors*, sebagaimana dituliskan dalam Tabel II.

TABEL II

FAKTOR RISIKO PEMELIHARAAN PERANGKAT LUNAK DIMITIGASI OLEH TEKNIK MANAJEMEN RISIKO

No	Faktor Risiko Pemeliharaan Perangkat Lunak	Penjelasan	Teknik Manajemen risiko
1	Pengetahuan / keterampilan yang tidak memadai.	Dengan menggunakan Anggota tim proyek memiliki pengetahuan yang tidak memadai tentang teknologi yang dipakai pada proyek, atau bisnis, atau mungkin tidak memiliki pengalaman untuk menangani proyek [23], pengelolaan manajemen risiko ini adalah hal dasar dalam membentuk tim <i>development</i> karena pengetahuan dan ketrampilan dapat mempengaruhi performa.	C1
2	Manajemen perubahan yang tidak memadai.	Manajemen perubahan diartikan sebagai upaya mengelola manusia melalui naik turunnya emosi yang pasti terjadi ketika suatu organisasi sedang mengalami perubahan besar-besaran [24].	C1, C20
3	Politik korporat dengan efek negatif pada proyek perangkat lunak.	Politik korporat yang berdampak negatif adalah penundaan / kegagalan proyek perangkat lunak. Oleh karena itu, penting bahwa organisasi TI memiliki dukungan penuh dari manajemen puncak untuk mengembangkan dan memiliki kebijakan di tempat dan tidak akan mentolerir dengan sedikit pun.	C1, C23, C22, C9, C7
4	Kurangnya Sumber Daya dan fasilitas referensi	Aritua menjelaskan pada penelitiannya bahwa fasilitas yang tidak mencukupi dan referensi yang merupakan risiko lain dalam fase pemeliharaan [25], pada penelitian ini dapat dikatakan kelengkapan dan kesiapan dokumentasi atau aset yang akan digunakan pada proses <i>development</i> akan mempengaruhi performa pengerjaan proyek.	C1, C23
5	Kurangnya komitmen dan dukungan manajemen puncak dan keterlibatan.	Dukungan manajemen puncak adalah kebutuhan mutlak untuk menangani perangkat lunak dan komitmen secara efektif. Oleh karena itu, faktor krusial utama adalah komitmen manajemen puncak dan kegagalan di banyak tempat yang disebabkan oleh kurangnya keterlibatan manajemen puncak yang dilaporkan [26].	C11, C3
6	Kekurangan dalam komponen yang diperlengkapi secara eksternal, <i>Commercially available Off-The-Shelf (COTS)</i> .	Komponen yang menciptakan risiko <i>maintenance</i> berasal dari faktor eksternal yang dapat menjadi sumber utama risiko dalam fase pemeliharaan seperti yang dikatakan oleh Galorath [27].	C2, C12
7	Pemanfaatan proyek	Faktor <i>maintenance</i> berasal dari warisan kode, <i>database</i> , atau desain	C3, C12, C14

No	Faktor Risiko Pemeliharaan Perangkat Lunak	Penjelasan	Teknik Manajemen risiko
	Perangkat Lunak Lama.	yang tidak ter <i>update</i> sehingga dapat mengakibatkan program tidak dapat dimanfaatkan kembali [28].	
8	Ketidaksesuaian proses akuisisi dan kontrak.	Risiko akuisisi dan ketidaksesuaian proses kontrak adalah salah satu risiko yang lebih tinggi dari pemeliharaan perangkat lunak. Ini karena proses pembuatan kontrak lebih kompleks multidimensi [29].	C8, C12, C5
9	Dokumentasi pengguna hilang atau tidak lengkap.	Dokumentasi yang tidak konsisten atau tidak lengkap adalah penyebab utama kesalahan dalam pengembangan dan pemeliharaan perangkat lunak [30].	C8, C24, C1
10	Tindakan persaingan yang merugikan.	Komunikasi yang baik antara tim perangkat lunak dan pelanggan akan memungkinkan mereka untuk memahami pilihan kompetitif dan strategis untuk sistem perangkat lunak [18].	C1, C9

C. Metode Penelitian Kuantitatif

Dengan pendekatan metode penelitian kuantitatif yang bersifat empiris, data penelitian diambil dari hasil pengisian kuesioner oleh manajer proyek. Data berkaitan dengan proyek yang sedang atau sudah selesai dikerjakan oleh manajer proyek. Hal ini sejalan dengan pengelompokkan

pada teknik manajemen risiko dengan menggunakan pendekatan *software maintenance risk factors* dalam Tabel I dan II.

Untuk membuat kuesioner dalam pengumpulan data dan identifikasi masalah maka akan disusunlah kumpulan pertanyaan sebagaimana ditampilkan dalam Tabel III.

TABEL III.

PENJELASAN DAN PERTANYAAN IDENTIFIKASI MASALAH

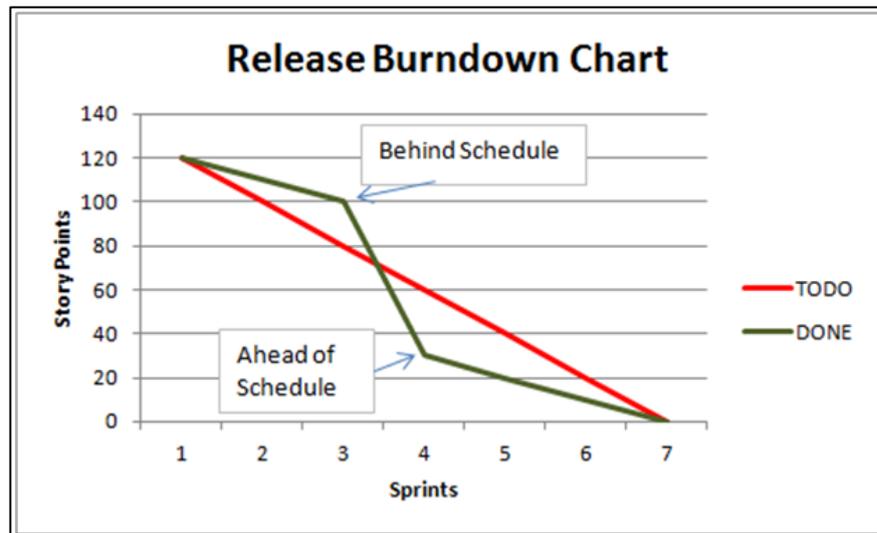
No	Kode	Motivasi	Pertanyaan
1	C1_K1	Komponen <i>risk management techniques</i> C1 maka dibuat suatu pertanyaan yang dapat melakukan identifikasi. faktor <i>inadequate knowledge/skills</i> memiliki fokus pada pengetahuan para anggota tim pembuatan perangkat lunak, sedangkan pada komponen <i>risk management techniques</i> . C1 memiliki fokus pada cara identifikasi fitur agar fitur yang dibuat sudah diidentifikasi agar pekerjaan yang diberikan pada setiap anggota tim adalah fitur yang dibuat sesuai kebutuhan, ditambah dengan pengertian faktor manajemen risiko maka pekerjaan yang dibuat haruslah sesuai dengan kemampuan dan kesesuaian pengetahuan, maka untuk dibuat identifikasi dibuat pertanyaan.	Apakah <i>product backlog</i> sesuai dengan yang dibuat <i>developer</i> ?
2	C1_K2	Pada komponen ini memiliki identifikasi teknik manajemen risiko yang sama dengan faktor <i>maintenance</i> perangkat lunak pengetahuan/keterampilan yang tidak memadai namun perlu dibedakan karena memiliki pendekatan yang berbeda sehingga pertanyaan yang dibuat perlu dibedakan menyesuaikan dengan faktor risiko <i>maintenance</i> untuk <i>inadequate change management</i> . Pada faktor <i>inadequate change management</i> memiliki fokus pada perubahan yang terjadi pada organisasi, perubahan bisa berdampak pada formasi tim development proyek maka perlu diidentifikasi apakah ada perubahan formasi tim tertentu yang mengakibatkan adanya pergantian anggota atau pengurangan, maka dibuat pertanyaan	Apakah <i>product backlog</i> dikerjakan oleh orang yang berbeda?
3	C20_K2	Pada analisis teknik ini perlu melakukan identifikasi yang menyangkut faktor manajemen risiko <i>maintenance</i> terhadap kualitas teknis pengembang perangkat lunak.	Adakah pergantian <i>developer</i> selama sprint ini berlangsung yang memerlukan data konfirmasi kelengkapan anggota saat <i>development</i> ?
4	C1_K3	Untuk melakukan identifikasi risiko dengan teknik C1, diperlukan pertanyaan yang menyangkut faktor risiko yang membicarakan tentang pengaruh pihak luar khususnya manajemen puncak atas performa <i>development</i> . Pada komponen C1 ini difokuskan pemilihan fitur yang akan dikerjakan agar performa penyelesaian bisa lebih cepat.	Adakah campur tangan pihak lain yang mempengaruhi hasil akhir?
5	C7_K3	Pada teknik manajemen C7 bertujuan untuk membuat tindakan antisipasi bila terjadi risiko saat berjalannya <i>sprint</i> , dihubungkan dengan faktor risiko yang perlu memperhitungkan kepentingan manajemen puncak, maka	Apakah sudah dipersiapkan <i>buffering</i> pada <i>sprint</i> saat ini?

No	Kode	Motivasi	Pertanyaan
		diperlukan pendekatan untuk mengurangi risiko terhadap beban manajer proyek.	
6	C9_K3	Komponen C9 ini adalah teknik manajemen dengan mengadakan <i>meeting</i> . Pada kasus ini Agile sudah memiliki sesi <i>meeting</i> yang dinamakan <i>scrum meeting</i> . Namun pada teknik manajemen ini, pemegang keputusan tertinggi tidak mungkin selalu mengikuti sesi <i>meeting</i> setiap proyek pada satu waktu, maka peran proyek manajer sudah cukup untuk memimpin <i>meeting</i> , dan pemegang keputusan tertinggi hanya perlu memantau.	Apakah proses <i>scrum meeting</i> dilakukan setiap hari?
7	C22_K3	Pada teknik identifikasi C22 ini dilakukan pengawasan akan perubahan yang terjadi untuk dikombinasikan dengan <i>risk factor</i> .	Apakah setiap anggota pada proyek sudah memiliki perannya masing-masing?
8	C2_K3	Pada pertanyaan yang dibuat dengan teknik C23 menjelaskan bahwa edukasi kepada pengguna/klien terhadap perubahan yang dilakukan saat proses <i>development</i> sebelum <i>end sprint</i> akan mempengaruhi <i>timeline</i> .	Sudahkah memberitahu klien akan hal-hal yang dapat terjadi jika melakukan perubahan saat pengerjaan?
9	C1_K4	Dengan kombinasi teknik identifikasi risiko pertanyaan perlu dibuat untuk mewakili kebutuhan data apakah kebutuhan saat <i>development</i> sudah terpenuhi karena pada identifikasi C1 diperlukan data untuk dilakukan pengujian untuk memisahkan fitur yang diperlukan dan tidak.	Apakah setiap kebutuhan <i>development</i> sudah terpenuhi?
10	C23_K4	Pentingnya menjelaskan kembali akan layanan yang diberikan oleh perusahaan akan membantu manajemen proyek dalam mengurangi risiko terjadinya <i>maintenance</i> , terutama akan kesepakatan yang sudah disepakati oleh perusahaan dan klien, maka setiap adanya indikasi melencengnya kesepakatan maka proyek manajer berhak untuk mengingatkan klien, maka manajemen proyek harus mampu melakukan edukasi.	Apakah informasi pentingnya fasilitas dan sarana proyek akan memberi berpengaruh pada <i>development</i> ?
11	C3_K5	Dukungan manajemen puncak perlu mendengarkan tim <i>development</i> yang dimiliki terutama tim manajemen internal sendiri, pada saat manajemen sedang melakukan audit atau <i>sprint planing</i> manajer proyek harus melakukan evaluasi kembali kinerja dan melakukan perbandingan antara data <i>sprint planing</i> yang sudah disepakati dengan hasil akhir, di sini para manajemen puncak perlu mengerti situasi dan keadaan bila diperlukan perhitungan ulang untuk mengganti estimasi selesainya proyek, proyek manajer harus mampu membuat keputusan dengan membuat analisis.	Sudahkah melakukan perhitungan kembali untuk estimasi proyek?
12	C11_K5	Pada analisis identifikasi risiko ini proyek manajer perlu melakukan kajian ulang pada setiap fitur yang akan dibuat pada proyeknya, karena pada setiap komponen yang di ajukan klien masalah sebuah gambaran besar yang dibutuhkan, maka untuk mengurangi risiko perubahan estimasi saat proses <i>development</i> maka manajer proyek harus melakukan pemecahan pekerjaan pada <i>product backlog</i> .	Sudahkah menginformasikan solusi jika ada kendala pada saat <i>product backlog</i> ?
13	C2_K6	Permintaan dari pihak eksternal sangat mempengaruhi kinerja dan hitungan estimasi pada selesainya proyek, namun pada sisi lain permintaan haruslah di tanggapi dengan baik karena akan berpengaruh pada kepuasan pelanggan terhadap pelayanan pada perusahaan maka saat permintaan yang diajukan saat proses <i>development</i> haruslah dipertimbangkan dengan baik agar tidak memberikan risiko.	Sudahkah dipertimbangkan berbagai pengaruh terhadap penggabungan komponen-komponen baru?
14	C12_K6	Penggunaan kembali kode dan desain sangat membantu mengurangi risiko saat proses <i>development</i> , faktor eksternal yang mengganggu adalah bila kode yang dibuat tidak memiliki standar penulisan yang buruk sehingga akan memiliki dampak yang buruk pada pengguna selanjutnya, maka standarisasi kode dianjurkan untuk mengurangi risiko saat proses <i>development</i> , walaupun kode yang di terima ada risiko kurang dipahami, namun saat kode sudah diperbaiki dan digunakan untuk konsumsi tim <i>development</i> , setidaknya kode yang digunakan sudahlah diperbaiki.	Sudahkah menerapkan standarisasi kode ?
15	C3_K7	Risiko dari penggunaan <i>framework code</i> adalah menciptakan risiko baru dari memodifikasi kode, tidak cocok dan perbedaan fungsi yang dibuat akan membuat risiko juga.	Sudahkah melakukan <i>code review</i> ?
16	C12_K7	Pewarisan program atau <i>framework</i> akan mempercepat dan mengurangi perbaikan program. Untuk mengurangi risiko perlu adanya ketersediaan dokumentasi atas <i>framework</i> yang dimiliki agar dapat dimanfaatkan dengan baik.	Sudahkah mempersiapkan dokumentasi program?

No	Kode	Motivasi	Pertanyaan
17	C14_K7	Penggunaan struktur database dan data yang sama dengan yang akan <i>hand offer</i> akan mengurangi risiko <i>maintenance</i> pada sektor basis data.	Apakah menggunakan struktur <i>database</i> dan data yang sama saat <i>development</i> ?
18	C26_K7	Untuk melakukan analisis dengan kode atau <i>framework</i> yang sudah ada dibutuhkan pemecahan fitur. Dengan demikian dapat ditentukan fitur yang dapat digunakan kembali atau dilakukan pengembangan maka diperlukan pemecahan pengerjaan pada bagian kecil.	Apakah dilakukan pemecahan fitur untuk analisis?
19	C5_K8	Saat mengerjakan proyek setiap anggota tim memiliki peranya masing-masing dan pekerjaan didistribusikan oleh manajer proyek. Manajer proyek menyusun pekerjaan dan diberikan pada anggota namun ada faktor eksternal yang membuat munculnya tugas yang memberikan beban pada tim dan pasti akan memberikan risiko <i>maintenance</i> karena mengganggu <i>timeline</i> proyek maka kepastian ini perlu disadari oleh manajemen proyek.	Apakah sebagian besar anggota tim dalam proyek telah menyelesaikan tugas sesuai rancangan proyek?
20	C8_K8	Untuk mengurangi risiko <i>maintenance</i> setiap anggota tim <i>development</i> harus memiliki peran masing-masing yang jelas, tugas yang diberikan juga sesuai kemampuan dan peran agar setiap anggota tim dapat bekerja secara fokus.	Memberikan tugas-tugas telah didistribusikan merata sesuai rancangan dan kemampuan anggota tim?
21	C12_K8	Dengan teknik ini manajer proyek perlu melakukan analisis fitur aplikasi yang tidak digunakan pada proyek agar tim dapat terfokus pada pengerjaan fitur yang sesuai saja.	Apakah terdapat proses analisis dan telah dilakukan penyesuaian ruang lingkup pada komponen yang tidak sesuai kontrak?
22	C1_K9	Dengan teknik ini manajemen proyek perlu memastikan dokumentasi yang dimiliki mengenai kebutuhan aplikasi/sistem yang akan dibuat apakah sudah lengkap agar dapat dianalisis spesifikasi yang dibutuhkan sesuai dengan kebutuhan program yang dibuat.	Apakah telah memastikan dokumentasi lengkap sesuai spesifikasi program yang dibuat?
23	C8_K9	Setelah dokumentasi lengkap, pembagiannya harus dilakukan secara merata dan sesuai dengan peranya masing-masing. Pembagian yang benar akan membantu setiap anggota tim yang memiliki perannya dalam fokus dalam membaca dan mempelajari dokumentasi.	Apakah sudah memastikan dokumentasi yang diberikan kepada <i>developer</i> sesuai dengan pembagian peran?
24	C24_K9	Memastikan dokumentasi yang diberikan kepada tim sudah lengkap dan memiliki kualitas yang baik seperti memastikan sumber dokumentasi yang sudah pada versi yang paling terbaru.	Apakah sudah memastikan kualitas dokumentasi agar pekerjaan menjadi mudah dan jelas?
25	C1_K10	Pada <i>risk factor</i> ini manajer proyek mampu menengahi setiap topik pembahasan dan pengerjaan pada setiap orang dari faktor internal dan eksternal. Untuk manajemen teknik C1, pihak pengelola proyek harus memastikan setiap pekerjaan yang di berikan pada tim internal harus sesuai dengan porsi yang adil.	Apakah telah memberikan batasan pada tugas, peran dan ruang lingkungannya agar tidak berlebih dan kurang?
26	C9_K10	Dengan menggunakan SDLC Agile, sesi <i>team building</i> sudah difasilitasi dengan adanya <i>scum meating</i> setiap harinya.	Apakah seluruh anggota tim telah mengikuti <i>sprint planning</i> dan <i>international meeting</i> (pihak eksternal)?

D. Klasifikasi Data

Pengelompokan data dibagi ke dalam tiga kelas, berdasarkan penggambaran *burndown chart* yang dapat dilihat pada Gambar 1.



Gambar 1 Burndown chart

Pada Gambar 1 terlihat grafik yang dimulai dengan nilai *scrum of task estimates* sebesar tiga puluh hari dan *iteration time* sebesar nol lalu diakhiri dengan *sum of task estimates* sebesar nol dan *iteration time* sebesar dua puluh, garis sumbu berwarna biru menggambarkan nilai ideal *tasks* dan pada garis sumbu berwarna merah menggambarkan contoh *Actual task*. *Burn down chart* menggambarkan garis sumbu berwarna merah yang menyerupai sumbu grafik berwarna biru atau secara data semakin bertambahnya *sprint* maka semakin sedikit pekerjaan atau *task* yang dimiliki.

Untuk melakukan pengelompokan akan digunakan data jumlah *sprint* yang sedang dilaksanakan dan total estimasi *sprint* yang dijalankan. Klasifikasi ini bertujuan untuk melakukan pengelompokan status *sprint* menjadi: awal *sprint*, pertengahan *sprint*, dan akhir *sprint*. Tujuannya untuk melakukan pengelompokan agar saat didapatkan hasil *Optimized Forest* pengelompokan data akan membantu memberikan informasi pada fase *sprint* yang manakah *sprint* proyek dapat dikatakan sedang bermasalah.

Setelah ditentukan klasifikasi data maka perlu ditentukan rentang nilainya. Pada penelitian ini, digunakan perhitungan persentase berdasarkan jumlah *sprint* dalam proyek. Persentase dimulai dari 0% - 33 % menunjukkan klasifikasi data fase awal *sprint*, 34%-67% menunjukkan klasifikasi data fase pertengahan *sprint*, dan 68-100% menunjukkan data memasuki fase akhir *sprint*.

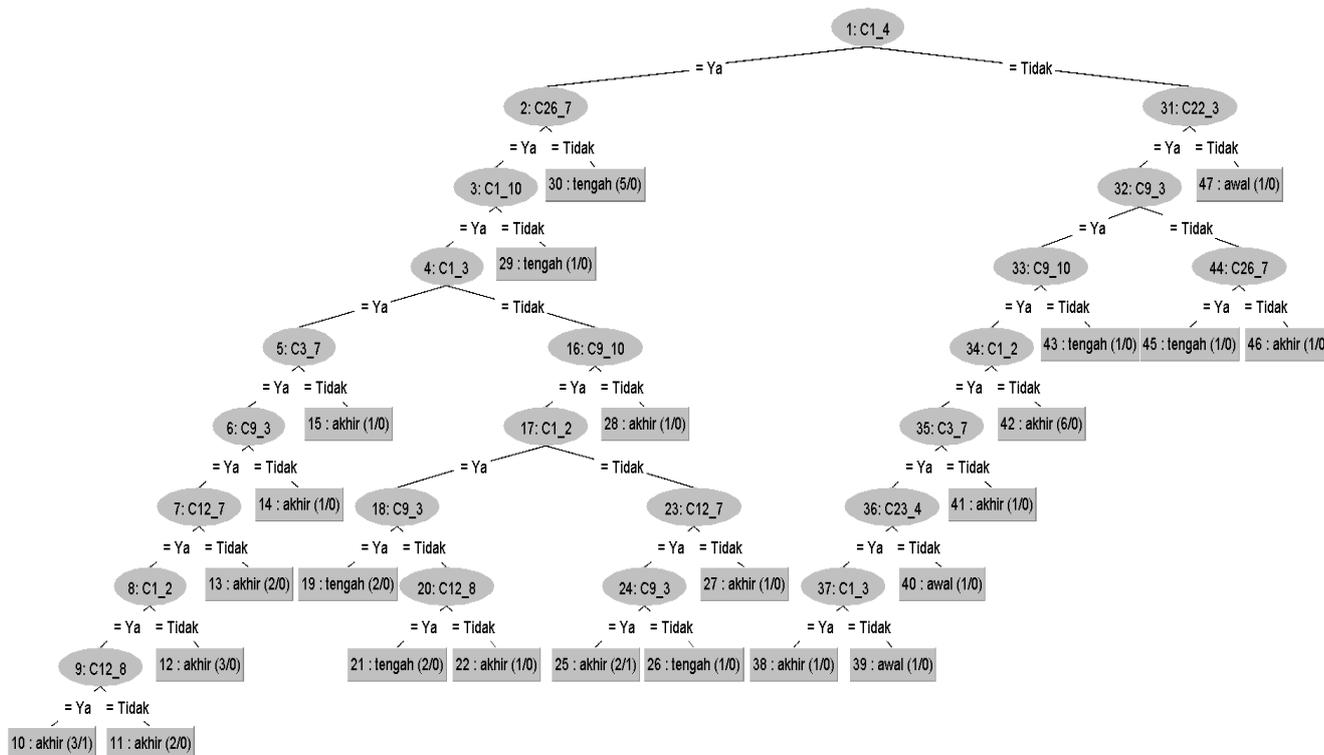
E. Optimized Forest

Setelah ditentukannya pengelompokan data langkah selanjutnya adalah mencari atribut yang akan menentukan komponen yang memiliki pengaruh besar pada risiko pemeliharaan, dengan menggunakan algoritma *Optimized Random Tree*. keluaran dari algoritma ini menyarankan peningkatan posisi optimal dari sekumpulan atribut dalam *dataset* [31].

Sebelum memasuki tahap optimasi akan dilakukan *attribute selection* untuk menentukan atribut yang memiliki bobot dan ragam data yang dapat di masukan pada optimasi, data hasil survei akan dilakukan pembersihan dengan mengubah nilai atribut menjadi kode C untuk teknik manajemen risiko, dan kode K sebagai faktor manajemen risiko. Setelah dilakukan pembersihan data dan melakukan pengelompokan data maka *attribute selection* dapat di jalankan. Sebagai batasan akan dilakukan penyaringan nilai atribut dengan nilai *information gain* minimal adalah 0,03, maka didapatkan kumpulan atribut pada Tabel III.

TABEL IV
RANKING ATRIBUT

No.	Atribut	Information Gain Atribut
1	C1_K4	0.17058
2	C1_K2	0.08848
3	C26_K7	0.08559
4	C22_K3	0.0844
5	C1_K3	0.08007
6	C8_K8	0.06781
7	C24_K9	0.06461
8	C12_K7	0.0599
9	C9_K3	0.04916
10	C23_K4	0.04301
11	C8_K9	0.04301
12	C11_K5	0.04202
13	C9_K10	0.04061
14	C1_K10	0.03635
15	C3_K7	0.03222
16	C12_K8	0.03133



Gambar 2 Visualisasi Pohon

Setelah dilakukan sortir pada atribut yang akan digunakan, proses selanjutnya yang akan dilakukan adalah dengan melakukan klasifikasi dengan *Random Tree* dengan menggunakan data yang sudah di sortir berdasarkan hasil nilai atribut yang lebih besar dari 0,03 dan didapatkan hasil visualisasi pohon pada Gambar 2.

Berdasarkan hasil visualisasi pada Gambar 2, dapat diambil kesimpulan atribut yang memiliki pengaruh besar terhadap risiko *maintenance* yang memiliki risiko pada data yang memberikan data “Tidak” pada setiap tiga kelompok data yang disimpulkan pada Tabel V

Kelompok Sprint.

TABEL V
KELOMPOK SPRINT

Kelompok Sprint	Atribut
Awal	C23_K4
Tengah	C9_K3
Akhir	C1_K2, C12_K8

IV. DISKUSI HASIL

Dari hasil akhir klasifikasi, menunjukkan bahwa setiap kelompok *sprint* memiliki atribut yang perlu diperhatikan saat berlangsungnya proyek berdasarkan pengaruh dari setiap komponen risiko.

TABEL VI
KELOMPOK SPRINT AWAL

Kelompok Sprint Awal	
Komponen	Pertanyaan
C23_K4	Apakah informasi pentingnya fasilitas dan sarana proyek akan memberi berpengaruh pada <i>development</i> ?

Pada fase awal sprint pentingnya informasi fasilitas dan sarana yang dapat dipakai tim sangat penting untuk memberikan kelancaran selama sprint ke depan, sehingga pada fase ini perlu dipersiapkan seluruh kebutuhan agar pada fase sprint awal ini risiko ini tidak menjadi kendala yang menjadikan estimasi meleset. yang dapat dilihat pada Tabel VI.

TABEL VII
KELOMPOK SPRINT TENGAH

Kelompok Sprint Tengah	
Komponen	Pertanyaan
C9_K3	Apakah proses <i>scrum meeting</i> dilakukan setiap hari?

Pada Tabel VII menunjukkan pada pertengahan sprint memiliki faktor risiko yang berfokus pada komponen C9

Apakah *Scrum* dilakukan setiap hari ?, pada pertanyaan ini memiliki arti di setiap harinya apakah ada terdapat komunikasi terhadap tim dan manajemen proyek karena pada fase pertengahan ini pekerjaan yang dimiliki tim sudah berkurang, dan pada fase pertengahan ini dapat melakukan penggambaran hasil akhir, sehingga pada komponen ini perlu diperhatikan agar komunikasi dapat berjalan dengan baik agar dapat dilakukan perhitungan estimasi kembali agar tidak terjadi *maintenance*.

TABEL VIII
KELOMPOK SPRINT AKHIR

Kelompok Sprint Akhir	
Komponen	Pertanyaan
C1_K2	Apakah <i>product backlog</i> dikerjakan oleh orang yang berbeda?
C12_K8	Apakah terdapat proses analisis dan telah dilakukan penyesuaian ruang lingkup pada komponen yang tidak sesuai kontrak?

Pada fase akhir yang dapat dilihat pada Tabel VIII sprint komponen C1_K1 memiliki fokus identifikasi risiko terhadap campur tangan pihak lain yang mempengaruhi pekerjaan. Pada fase akhir ini pihak eksternal perlu diberikan pengarahan serta edukasi. Bila terjadi perubahan akan mempengaruhi estimasi karena hasil pekerjaan pada fase sebelumnya dapat terusik dan dapat menambah pekerjaan dan mengganggu estimasi.

Pada komponen C12_K8 memiliki keterkaitan dengan komponen C1_K2. Pada komponen C12_K8 perlu ditegaskan dan melakukan analisis kembali pekerjaan yang dapat mempengaruhi estimasi, karena pada fase ini proyek memasuki tahap akhir dan sebaiknya tidak terjadi penambahan pekerjaan yang mempengaruhi hasil.

V. SIMPULAN

Dalam melakukan identifikasi risiko terjadinya pemeliharaan program dalam pengembangan perangkat lunak berbasis Agile dapat dilakukan dengan menggunakan Pohon Klasifikasi teknik ini sudah dapat dibuktikan dapat melakukan pengelompokan sehingga dapat membantu dalam proses identifikasi terutama pada klasifikasi yang bermasalah dalam mengurangi dan identifikasi dini terhadap risiko *maintenance*.. Dengan metode *random forest* dan sampel data yang dimiliki sudah dilakukan pengujian yang menghasilkan komponen risiko terkuat yang membuat risiko *maintenance* muncul saat tahap rentan pada tiga klasifikasi data.

Sebagai rencana selanjutnya melakukan analisis terhadap setiap komponen sprint adakah solusi dalam mengurangi risiko pada setiap komponen kelompok data, untuk dibuatnya penelitian ini diperlukan data set yang lebih

banyak agar tingkat keakuratan dapat menjadi lebih baik terutama dalam keanekaragaman data.

DAFTAR PUSTAKA

- [1] Kulkarnil, "Investigating Agile Adaptation for Project Development," *International Journal of Electrical and Computer Engineering (IJECE)*, vol. 7, no. 3, p. 1281, 2017.
- [2] Alain, "What you Need to Know about Software Maintenance," *The Maintenance and Asset Management Journal*, vol. 20(2), pp. 6-3, 2005.
- [3] Rather, "A Comparative Study Of Software," *International Journal of Application or Innovation in Engineering & Management (IJAEM)*, vol. 4, no. 10, p. 23, 2015.
- [4] Sharma, "Agile Processes and Methodologies: A," *International Journal on Computer Science and Engineering (IJCSE)*, vol. 4, no. 5, p. 892, 2021.
- [5] Rilyani, "Analisis Risiko Teknologi Informasi Berbasis Risk Management Menggunakan Iso 31000 (studi Kasus : I-gracias Telkom University)," *eProceedings of Engineering*, vol. 2, no. 2, p. 6202, 2015.
- [6] Istiningrum, "implementasi penilaian risiko dalam menunjang pencapaian tujuan instansi pendidikan," *Jurnal Pendidikan Akuntansi Indonesia*, vol. IX, no. 2, p. 19, 2011.
- [7] Dalton, *Multi-Domain Master Data Management*, Amsterdam: Elsevier, 2015.
- [8] Linoff, *Data Mining Techniques for Marketing, Sales and Customer Relationship Management*, Wiley Publishing, 2004.
- [9] Kulkarni, "Efficient learning of random forest classifier using disjoint partitioning approach," *Lecture Notes in Engineering and Computer Science*, vol. 2, 2013.
- [10] A. Primajaya, "Random Forest Algorithm for Prediction of Precipitation," *Indonesian Journal of Artificial Intelligence and Data Mining*, vol. 1, p. 29, 2018.
- [11] McConnell, *Rapid development: Taming wild software schedules*: Microsoft Press, Microsoft Press, 1996.
- [12] Kwan-SikNa, "Software development risk and project performance measurement: Evidence in Korea," *Journal of Systems and Software* 80(6), vol. 80, p. 596-605, 2007.
- [13] Ropponen, "Components of software development risk: How to address them?," *IEEE TRANSACTIONS ON SOFTWARE ENGINEERING*, vol. 26, no. 2, 2000.
- [14] Addison, "Controlling Software Project Risks—an Empirical Study of Methods Used by Experienced Project Managers," *Proceedings of SAICSIT*, 2002.
- [15] Severin, "Risks and Controls in the Implementation of ERP Systems," *The International Journal of Digital Accounting Research*, vol. 1, 2001.
- [16] Jiang, "Reducing user-related risks during and prior to system development," *International Journal of Project Management* 20, 2002.
- [17] Jones, *Applied software measurement global analysis of productivity and quality* 3rd ed, McGraw-Hill Education; 2008.
- [18] Kasurinen, "Supporting the curation of biological databases with reusable text," *Genome Informatics*, vol. 16, pp. 32-44, 2005.
- [19] Dyba, "Empirical studies of agile software development: A systematic review,," *Information and Software Technology*, vol. 50, p. 833-859, 2008.
- [20] Fairley, *Managing and Leading Software Projects*, New Jersey: Wiley, 2008.
- [21] Persohn, "Interactive Real-Time Embedded Systems Education," *2011 35th IEEE Annual Computer Software and Applications Conference*, 2011.

- [22] Bavani, "Global software engineering: Challenges in customer value creation," *presented at the 2010 International Conference on Global Software Engineering*, 2010.
- [23] Brandon, "Project management for modern information system," *Project Management for Modern Information Systems*, 2006.
- [24] Elzamly, "Mitigating Software Maintenance Project Risks with Stepwise Regression," *Journal of Modern Mathematics Frontier*, vol. 3, no. 2, p. 41, 2014.
- [25] Keil, "Pulling the Plug: Software Project Management and the Problem of Project Escalation. *MIS Quarterly*," vol. Vol 19, pp. 421-441, 1995.
- [26] Linda, *Managing Business with SAP: Planning, Implementation and Evaluation*, Hershey: Idea Group Publishing, 2005.
- [27] Aritua, "What Risks Are Common to or Amplified in," *International Journal of Project*, vol. 12, p. 303, 2011.
- [28] Kerzner, *A Systems Approach to Planning, Scheduling, and Controlling* 10th ed, New Jersey: wiley, 2009.
- [29] Galorath, *Software Sizing, Estimation, and Risk Management: When Performance is Measured Performance Improves* 1st Edition, Auerbach Publications, 2006.
- [30] Wong, "Critical success factors for implementing knowledge management in small and medium enterprises," *Journal of the Australian Institute of Project Management* 24, vol. 105, p. 28-31, 2004.
- [31] Boehm, "USC CSSE Workshop Overview: Top 3 Software-Intensive Systems Risk Items.," dalam *carer for systems and software engineering*, 2007.
- [32] Binder, *Testing Object-Oriented Systems: Models, Patterns, and Tools (ARP/AOD)*, Boston: Addison-Wesley, 1999.
- [33] Orujpour, "Multi-modal forest optimization algorithm," *Neural Computing and Applications*, vol. 32, 5 3 2019.