

# PENGEMBANGAN SISTEM INFORMASI PENGELOLAAN KINERJA KARYAWAN (STUDI KASUS: MODENA STRATEGY SYSTEM)

Dinastutyasakti Bening Kinasih  
Jurusan Informatika  
Fakultas Teknologi Industri  
Universitas Islam Indonesia  
Yogyakarta, Indonesia  
17523136@students.uii.ac.id

## ABSTRAKSI

Berbagai cara dilakukan perusahaan untuk dapat memaksimalkan pengelolaan kinerja karyawan perusahaan. Pemanfaatan teknologi menjadi salah satu solusi yang ditempuh agar pengelolaan kinerja karyawan dapat terukur dengan baik. Artikel ini menyajikan pengembangan sistem informasi Modena Strategy System (MSS) dengan menggunakan metode pengembangan *agile* dengan kerangka kerja *scrum*. MSS ini dibuat karena tata kelola kinerja karyawan yang ada di perusahaan Modena selama ini tidak efektif. Proses pengelolaan evaluasi kinerja sebelumnya dilakukan secara manual dengan menggunakan file dengan format excel.xlsx sehingga para *reviewer*, *advisor*, dan *approver* memerlukan proses yang lama dalam penyelesaiannya. Implementasi pendekatan *scrum* untuk MSS terbukti dapat meningkatkan kecepatan dan fleksibilitas untuk pengembangan sistem informasi. Pendekatan *scrum* memerlukan banyak interaksi dengan klien terkait kebutuhan yang ingin ditambah maupun diubah dalam aplikasi serta bertujuan untuk mencapai proses eksekusi dari pemenuhan kebutuhan dan terlaksananya penyelesaian dari masalah yang dihadapi.

## Kata Kunci

Agile; Scrum; Sistem Informasi; Modena Strategy System;

## 1. PENDAHULUAN

Setiap organisasi atau perusahaan tentunya ingin mencapai sebuah kesuksesan bagi perusahaannya. Hal ini dapat dicapai dengan beberapa faktor pendukung dan salah satunya adalah kinerja karyawan yang baik dan berkualitas. Tidak dapat dipungkiri bahwa dalam tubuh sebuah perusahaan terdapat perbedaan kinerja pada setiap individu. Perbedaan kinerja tersebut dapat berupa perbedaan pada aspek disiplin, mandiri, kerja keras, memiliki kemampuan komunikasi, manajemen waktu yang baik dan kolaborasi antar tim. Dengan adanya perbedaan tersebut maka perlu dilakukan pemantauan agar tidak terjadi permasalahan yang nantinya dapat menghambat jalannya proses bisnis perusahaan.

MODENA merupakan sebuah perusahaan yang berdiri di Indonesia sejak tahun 1981 sebagai perusahaan yang menyediakan alat-alat kebutuhan rumah tangga. MODENA melakukan evaluasi kinerja karyawan menggunakan *Key Performance Indicator* (KPI) yang mengikat pada masing-masing individu maupun tim. KPI digunakan untuk mengukur seberapa baik perusahaan/organisasi maupun individu dalam mencapai tujuan strategis yang telah ditetapkan [15] [16]. KPI adalah salah satu model pengukuran kinerja karyawan yang termasuk dalam Performance Prism.

Performance Prism memiliki kelebihan yaitu dapat digunakan sebagai bahan pertimbangan untuk menentukan kebutuhan dan kepuasan dari seorang *stakeholder*. Selain itu, pengukuran kinerja dari Performance prism mampu mengukur seberapa banyak kontribusi dari *stakeholder* yang diberikan untuk perusahaan [1].

Pengukuran kinerja dilakukan untuk meningkatkan kemajuan suatu perusahaan/organisasi ke arah yang lebih baik dan berlaku bagi individu serta mengatasi masalah ketidakefektifan kinerja karyawan [2]. Di MODENA pengukuran kinerja KPI akan diterapkan dalam sebuah sistem informasi untuk pengelolaan kinerja karyawan. Sistem informasi pengelolaan kinerja karyawan yang dikembangkan bagi perusahaan ini bernama Modena Strategy System (MSS).

Dasar pengembangan MSS berawal dari ketidakefektifan evaluasi kinerja karyawan di perusahaan MODENA karena jumlah karyawan yang dimiliki banyak hingga ratusan pegawai. Selain itu, banyaknya divisi kerja yang tersebar dan terdapat di tiap cabang perusahaan MODENA dengan lokasi yang berbeda pula juga menjadi penyebab sangat diperlukannya penerapan pengembangan sistem informasi [14]. Selama ini langkah-langkah evaluasi kinerja karyawan yang dilakukan di MODENA membutuhkan proses yang lama dan memerlukan persetujuan dari pihak *reviewer*, *advisor* dan *approver*. Proses evaluasi kinerja yang dilakukan sebelumnya dilakukan secara manual dengan menggunakan fail berformat excel.xlsx. Dari permasalahan yang disebutkan oleh konsumen dapat disimpulkan bahwa semua permasalahan tersebut ingin diatasi dalam satu wadah yang sama.

Pengembangan MSS akan diimplementasikan dengan menggunakan sebuah metode pengembangan *agile* dengan kerangka kerja *scrum*. *Agile development* didefinisikan sebagai metodologi pengembangan perangkat lunak yang syarat dan solusi dari pemenuhan kebutuhan diperoleh melalui kolaborasi antar tim peran yang berbeda-beda dan *self-organizing* [13]. *Scrum* merupakan sebuah *framework* yang dapat digunakan untuk berbagai macam proses dan teknik dalam penerapannya. Kelebihan dari kerangka kerja ini memiliki sifat iteratif yakni setiap prosesnya terjadi secara berulang dan incremental yang memiliki arti guna memastikan versi produk yang berpotensi untuk digunakan selalu siap tersedia [3].

## 2. KAJIAN PUSTAKA

### 2.1 Penelitian Terdahulu

Terdapat beberapa penelitian yang telah dilakukan sebelumnya terkait implementasi kerangka kerja *scrum*. Pada penelitian yang

berjudul “Implementasi Metodologi *Scrum* dalam Pengembangan Sistem Pembayaran Elektronik Pada Usaha Mikro Kecil Menengah”, dijelaskan bahwa kerangka kerja *scrum* sangat sesuai untuk diterapkan sebab dapat mengantisipasi perubahan yang mungkin bisa terjadi didalam proses pengembangan suatu perangkat lunak atau sistem [4]. Selain itu, penelitian yang berjudul “Implementasi Kerangka Kerja *Scrum* pada Manajemen Pengembangan Sistem Informasi” menjelaskan bahwa metodologi ini tidak mengharuskan anggotanya untuk mengikuti aturan dari panduan *scrum* secara keseluruhan [5]. Perbedaannya terdapat pada penggunaan *burn down chart*. *Burn down chart* digunakan sebagai alat pengukuran visual yang menunjukkan pekerjaan selesai per hari terhadap proyek. Sedangkan penelitian yang dilakukan penulis, menggunakan *ganttt chart* sebagai monitoring proyek yang sedang berlangsung. Pada penelitian aplikasi monitoring perkembangan janin, dijelaskan bahwa metode *scrum* digunakan untuk mengelola pengembangan perangkat lunak yang tidak terlalu kompleks dan tidak terlalu membutuhkan tim yang besar [6].

Penelitian terkait sistem penilaian kinerja karyawan digunakan oleh PT. Tridharma Kencana. Perusahaan tersebut terus mengalami perubahan menuju arah perbaikan. Sebelumnya perusahaan tersebut menggunakan form penilaian sederhana, sekarang PT. Tridharma Kencana telah menggunakan sistem penilaian kinerja melalui sistem KPI. Sistem ini mengacu pada implementasi ISO 9001. Aspek-aspek utama dalam penilaian karyawan adalah; (1) *Value Acquisition Indicator*, (2) *General Performance Indicator*, (3) *Achievement Performance Indicator*, (4) *Personal Performance Indicator*. Aspek-aspek ini kemudian diturunkan dalam indikator-indikator kinerja karyawan dengan kriteria SMART, yaitu *Specific, Measurable, Achievable, Reasonable, Time-base* [7].

## 2.2 Scrum

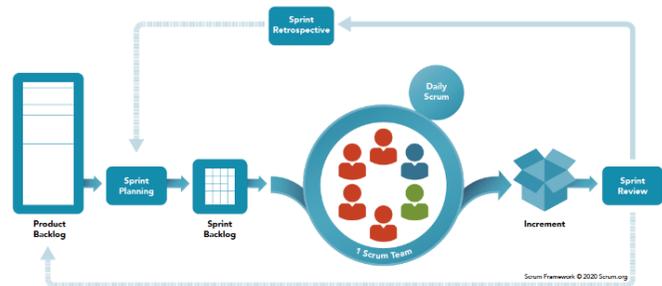
Konsep *scrum* adalah metode iterasi berkelanjutan yang dapat dilakukan untuk meningkatkan siklus hidup produk. *Scrum* selalu mengevaluasi dan meningkatkan hasil produk pengembangan dengan menyesuaikan proses agar lebih efektif [3]. Berdasarkan teori pengendalian empiris, *scrum* merupakan metodologi manajemen proyek yang berulang dan bertahap untuk mengendalikan risiko dan mengoptimalkan prediktabilitas proyek. Terdapat tiga faktor penting dalam proses *scrum* yaitu transparansi, inspeksi, dan adaptasi. Ulasan rinci faktor-faktor tersebut adalah sebagai berikut:

- Transparansi bersifat jelas, dapat dilihat oleh tim proyek pengembangan, serta mendefinisikan secara keseluruhan proses yang dikerjakan selanjutnya.
- Inspeksi pada *scrum* harus memiliki sifat kritis pada artefak untuk menemukan permasalahan.
- Adaptasi pada *scrum* memungkinkan berkembangnya proses pengembangan proyek menuju *sprint goal*.

Kerangka kerja *scrum* terdiri dari *scrum team*, acara (*event*), dan artefak (*artifact*). *Sprint* merupakan inti dari proses kerangka kerja *scrum*. Penyelesaian masalah dilakukan selama kurang dari satu bulan terhadap batasan yang ditetapkan sebuah *sprint*. *Sprint* memiliki durasi yang konsisten sepanjang daur hidup pengembangan produk [8]. Tahap-tahap *sprint* pada *scrum* yang dilakukan yaitu penentuan *Product Backlog*, *Sprint Planning*, *Sprint Backlog*, *Daily Scrum/Stand Up Meeting*, *Sprint Review/Demonstration*, *Sprint Retrospective*. [9]

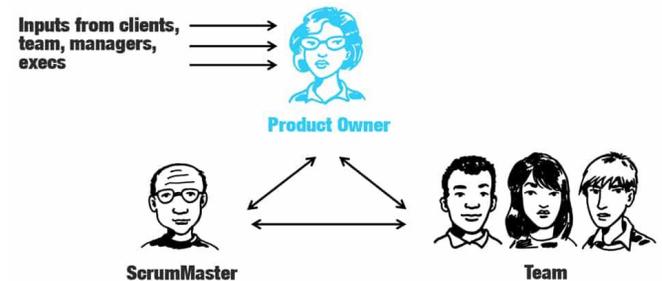
### 2.2.1 Alur Scrum

Penggambaran sistem merupakan langkah awal yang akan dikerjakan untuk penerapan metode *scrum*. Seorang *Product Owner* akan mendefinisikan rancangan rencana kerja dan dimasukkan ke *Product Backlog*. *Product Backlog* berisi rencana kebutuhan yang telah terdefinisi dan harus dikerjakan oleh tim yang terlibat. Selain itu, *Product backlog* perlu dikelola secara baik dan efektif guna memahami perencanaan produk [8]. Inti dari proses eksekusi kerangka kerja *scrum* adalah *sprint* dengan batas waktu yang berlangsung selama satu sampai empat minggu. *Sprint Planning* adalah tahap awal perencanaan untuk menentukan aktifitas yang termasuk dalam *sprint* yang akan dilakukan pada tahap berikutnya dan berfokus terhadap penyelesaian masalah [9]. Perencanaan yang telah disusun akan didiskusikan pada tahapan *Daily Scrum* yang diadakan setiap pagi pada waktu yang sama selama 15 menit. Tim proyek berdiskusi tentang “Hasil dari *Daily Scrum* kemarin, apa yang telah dilakukan?”, “Apa kendala yang ditemukan dan dihadapi selama proses pengembangan?”, dan “Langkah dan rencana apa yang akan dilaksanakan untuk *sprint* selanjutnya?” [9] [10]. Poin penting berupa tujuan, ruang lingkup dan hal-hal yang berkaitan dengan produk dipastikan kembali dan dapat dipahami dengan baik dan benar dalam *scrum team*. Hal ini merupakan tanggung jawab dari *scrum master*.



Gambar 1. Alur Scrum [11].

### 2.2.2 Tim Scrum



Gambar 2. Tim Scrum [12].

Dalam tim *scrum*, terbagi atas tiga peran: *Product Owner* (pemilik produk), *Scrum Master* dan *Development Team* (tim pengembang). *Product Owner* memiliki tanggung jawab dalam menentukan serta memaksimalkan nilai bisnis dari suatu produk. [9] *Product Owner* menyusun kebutuhan-kebutuhan proyek yang memiliki prioritas tinggi dan harus segera dikerjakan oleh Tim Kebutuhan- [9] kebutuhan proyek yang telah disusun dimasukkan ke dalam *Product Backlog*. *Development Team* adalah orang yang memiliki peran untuk pengerjaan proses pengembangan produk, seperti *System Analyst*, *Programmer*, *Software Tester*, dan *Quality Assurance*. Isi dari *Product Backlog* merupakan tanggung jawab dari setiap peran yang ada pada *Development Team* Peran. [9] *Scrum Master* adalah melakukan manajemen dari tiap proses *scrum*

saat proses pengembangan sedang berlangsung. *Scrum Master* akan menjelaskan proses dan tahapan kerja *scrum* kepada tim yang terlibat serta menegaskan apakah penerapan *scrum* dapat dipahami oleh setiap individu .[9]

Di dalam tim itu sendiri, terdapat beberapa posisi:

- a. *Project Manager*, mengelola kemajuan proyek dan menyesuaikan pekerjaan sesuai kebutuhan, memastikan proyek memenuhi tenggat waktu, mengelola hubungan dengan klien dan pemangku kepentingan serta merancang dan menandatangani kontrak. Posisi pada *scrum* adalah sebagai *product owner*.
- b. *System Analyst*, bertanggung jawab untuk keberhasilan proyek seperti *requirement gathering* (mengumpulkan kebutuhan), memproduksi spesifikasi untuk sistem baru atau yang dimodifikasi, dan berkomunikasi dengan programmer untuk menghasilkan sistem baru. Posisi *system analyst* termasuk sebagai *development team*.
- c. *Programmer*, bertanggung jawab untuk mengeksekusi dan implementasi perancangan kebutuhan seperti kegiatan *coding* dan *debugging*, mengelola sistem *database*, memelihara sistem operasi, mengedit kode sumber (*source code*), dan menganalisis algoritma.
- d. *Quality Assurance (QA)*, bertanggung jawab untuk memastikan bahwa produk akhir memenuhi standar kualitas perusahaan dan berjalan sesuai dengan kesepakatan. Posisi pada *scrum* adalah anggota *scrum team* sebagai *development team*.
- e. *Software Tester*, bertanggung jawab menyiapkan skenario pengujian, menjalankan tes pada perangkat lunak, menganalisis kesalahan atau *bug*, dan menyiapkan laporan hasil pengujian untuk diserahkan kepada *programmer*.

### 3. METODOLOGI

Penelitian yang disajikan dalam artikel ini dimulai dari tahap persiapan sebelum sistem MSS dikembangkan. Prosesnya secara umum mengikuti alur yang terdapat pada pendekatan *scrum*. *Scrum* adalah salah satu metodologi tangkas yang dirancang untuk memandu suatu perusahaan dalam memberikan hasil berupa solusi dari permasalahan yang terbilang rumit. Fokusnya adalah pada penggunaan proses empiris yang memungkinkan tim merespons perubahan dengan cepat, efisien, dan efektif. Metodologi *scrum* dapat dijadikan sebagai pendekatan dalam manajemen proyek. Berikut adalah tahap-tahap yang dilakukan saat pengembangan MSS.

#### 3.1 Penentuan Product Backlog

*Product Backlog* berisikan daftar rancangan dari tugas-tugas yang harus diselesaikan. Penambahan fitur-fitur baru, perbaikan, maupun perubahan bagi pengembangan sebuah produk yang diminta oleh klien termasuk dalam *product backlog*. *Product Owner* yang bertanggung jawab untuk menjaga *product backlog* tetap *up-to-date* pada *scrum board*. Dalam penelitian ini, metode *scrum* yang diterapkan membuat sebuah *scrum board* berupa *task list* yang digunakan untuk mendefinisikan pembagian tugas-tugas yang telah dianalisis oleh *analyst*. *Task list* tersebut dapat dilihat pada Tabel 1.

Setelah menentukan pembagian *task list*, daftar kebutuhan tersebut akan dikategorikan berdasarkan level tingkatan dengan label untuk dijadikan prioritas dalam pengerjaannya yang dapat dilihat pada Tabel 2.

Di Tabel 2, terdapat *label issue*. Label ini digunakan untuk melaporkan adanya *bugs* dalam pengembangan sistem dan harus

Tabel 1. Task List

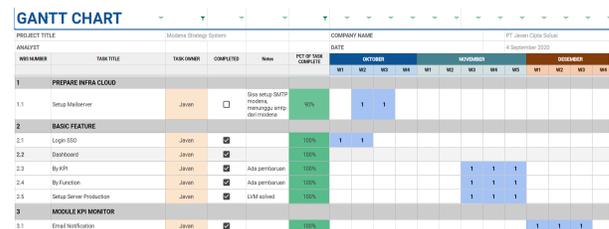
Assigned	Task List	Keterangan
<i>Analyst</i>	<i>Backlog</i>	Daftar kebutuhan proyek yang harus dianalisis oleh <i>analyst</i> agar bisa dikerjakan oleh <i>Programmer</i> .
<i>Programmer</i>	<i>To Do</i>	Tugas yang siap dikerjakan oleh tim.
	<i>In Progress</i>	Tugas yang sedang dikerjakan oleh anggota tim.
	<i>Feedback</i>	Berisi tugas yang diperoleh dari <i>issue/bug</i> yang ditemukan oleh tester atau dari <i>code reviewer</i> . <i>Task list</i> ini akan menjadi prioritas untuk segera diselesaikan.
<i>Code Reviewer</i>	<i>Code Review</i>	Melakukan <i>code review</i> terhadap <i>merge request</i> yang diajukan oleh tim.
<i>Tester</i>	<i>Ready to Test</i>	Tugas yang sudah dikerjakan oleh tim dan siap diuji oleh tester.

Tabel 2. Product Backlog

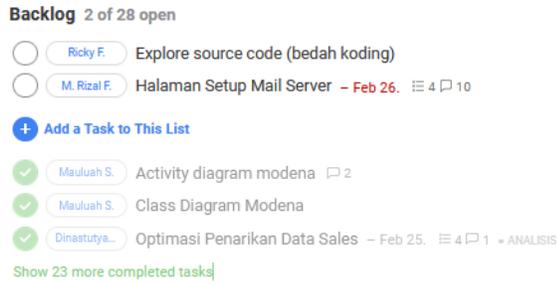
No	Jenis Aktifitas	Label Prioritas
1	Halaman tidak responsif ketika menampilkan data yang dipilih pada menu <i>Category KPI</i>	<i>Issue</i>
2	<i>Error</i> klik button “ <i>Ready to Evaluate</i> ” di menu <i>Project Progress</i>	<i>Issue</i>
3	Tidak dapat melakukan submit <i>KPI</i> yang sudah di <i>checklist</i>	<i>Issue</i>
4	Proses bisnis menu <i>Failure Management</i>	<i>Rabbit Holes</i>

segera dikerjakan oleh tim. Kemudian terdapat label *rabbit holes* yang berfungsi untuk menandakan bahwa daftar kebutuhan belum memiliki kejelasan dan kepastian. Di penelitian ini terdapat beberapa label lain yang dimanfaatkan pada *task list* untuk *product backlog* adalah sebagai berikut:

- a. Tingkat kesulitan task yaitu *Easy*, *Medium*, dan *Hard*.
- b. Tipe pekerjaan seperti *Server*, *Analyst*, *Coding*, *Testing*, *Documentation*, *Issue Code*, *Review*, dan *Change Request*.
- c. Status kesulitan seperti *Blocking* dan *Confirmation*.
- d. Sumber *task* yaitu *Sentry*, *Sonar*, *UI/UX*, *Performance*, dan *Security*.



Gambar 4. Gantt Chart.



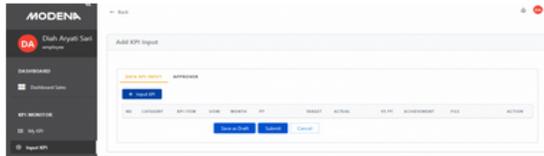
Gambar 5. Sprint Backlog.

Deskripsi:

- Tidak bisa submit KPI yang sudah di centang
- KPI yang dicentang tidak tampil
- Ketika dicoba input lagi, button input tidak berfungsi
- Ketika mau coba input lagi, perlu refresh, namun inputan tetap tidak bisa

URL:

<https://mss.cloud.javan.co.id/modules/kpi-input/create>



Skenario:

1. Login sebagai diah
2. Isi KPI di My KPI
3. Input KPI di tahun 2019

Gambar 6. Representasi Task.

### 3.2 Sprint Planning

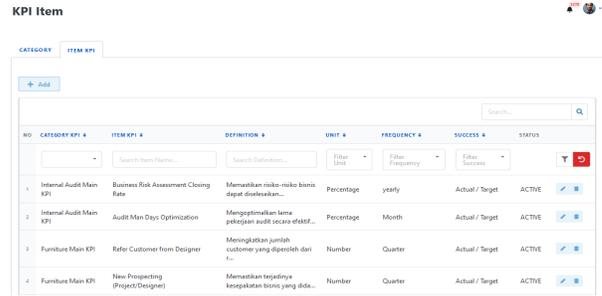
*Sprint planning* merupakan tahapan perencanaan dalam kerangka kerja *scrum*. Tahap ini dijalankan secara bersama-sama/kolaboratif oleh anggota tim *scrum*. [6] Penerapan *sprint planning* biasanya dilakukan untuk menentukan fitur yang dikerjakan, jumlah persentase hasil yang telah dibuat, *task owner* untuk tim serta durasi *sprint*. Hasil ini, dipresentasikan dengan *ganttt chart*. *Gantt chart* berfungsi untuk membantu *project manager* dan *system analyst* memastikan bahwa semua kegiatan yang berkaitan dengan proyek telah direncanakan, urutan pembagian kerja, dan durasi waktu kegiatan yang telah diprediksi.

Gambar 4 merupakan salah satu representasi *ganttt chart* milik proyek pengembangan MSS. Bagian di sebelah kiri merupakan daftar kegiatan atau bisa disebut dengan Work Breakdown Structure (WBS). Isi dari WBS berupa fitur-fitur kebutuhan proyek yang termasuk dalam *Product Backlog/Scope of Work (SOW)*. Bagian sebelah kanan adalah durasi waktu proyek yang sedang berlangsung (tahap pengembangan).

### 3.3 Sprint Backlog

*Sprint backlog* berisi semua tugas yang harus diselesaikan. Tugas-tugas diambil dari dari *product backlog* yang telah terdefinisi, diprioritaskan dalam *sprint*, dan ditugaskan ke *development team* untuk diselesaikan selama *sprint*. Pada penelitian yang dilakukan, penyusunan *sprint backlog* menggunakan aplikasi management project yaitu Active Collab yang dapat dilihat pada Gambar 5.

Gambar 5 menampilkan tugas-tugas yang diberikan untuk dikerjakan oleh *development team*. Tugas-tugas tersebut dibagi



Gambar 7. Category KPI.

dengan tim sesuai dengan peran. Tugas yang berkaitan dengan proses bisnis MSS dikerjakan oleh *system analyst*. Sedangkan untuk *programmer* mengerjakan tugas yang berkaitan dengan pengembangan proyek.

Gambar 6 menampilkan salah satu contoh rincian dari *task* yang terdapat pada *sprint backlog*. *Sprint backlog* yang telah disusun akan diberikan penjelasan secara rinci untuk tiap *task* seperti deskripsi, url, gambar sistem, skenario, ekspektasi, serta *role* yang digunakan untuk *login* pada sistem.

### 3.4 Daily Scrum/Stand Up Meeting

*Daily Scrum* diadakan untuk membicarakan tentang pekerjaan yang telah dilakukan di hari sebelumnya, mendiskusikan penutupan, dan menentukan pekerjaan apa yang akan diselesaikan hari itu. Setiap anggota tim memperbarui grup tentang apa yang telah mereka kerjakan dan mengemukakan masalah atau pertanyaan apa pun. Tahapan ini dilaksanakan selama 15 menit. Aplikasi yang digunakan seperti Zoom, Google Meet dan Whereby.

### 3.5 Spring Review/Demonstration

Pada penelitian ini, *sprint review* berlangsung di akhir *sprint* dengan tujuan untuk melihat kemajuan dari pengembangan proyek MSS yang telah dibuat. Tim akan mendemonstrasikan fitur-fitur yang sudah dikerjakan. Demo produk dilaksanakan oleh tim-tim yang terlibat pada proyek. Sebelum tahapan ini dilakukan, tim akan melakukan uji coba terlebih dahulu untuk memastikan tidak ditemukannya *bug/error* saat demo dilaksanakan. Apabila ditemukan bug saat dilakukan pengujian, QA dan *Software Tester* akan mencatat pada dokumen yang telah disediakan.

### 3.6 Sprint Retrospective

Tahapan pada *sprint retrospective*, tim *scrum* mendiskusikan daftar-daftar pekerjaan yang memiliki kendala, daftar pekerjaan yang dapat dan tidak dapat dikerjakan di tahap berikutnya. Cakupan tugas ini berdasar *task list* dan SOW yang telah terdefinisi.

## 4. HASIL & PEMBAHASAN

Berikut adalah beberapa hasil penerapan *Scrum* dalam pengembangan MSS:

- a. Halaman tidak responsif ketika menampilkan data yang dipilih pada menu Category KPI. Gambar 7 menampilkan halaman di menu Category KPI. Di halaman ini, user disuguhkan dengan kategori-kategori pada KPI dengan masing-masing itemnya. Di halaman ini sebelumnya, terdapat beberapa tampilan yang tidak sesuai seperti penomoran yang tidak teratur pada Category KPI, status ACTIVE yang menunjukkan bahwa KPI tersebut telah siap digunakan namun tidak tampil ketika user mengakses halaman ini. Kemudian, jarak antar Category KPI yang terlalu rapat. Perubahan-

**Tabel 3. Isu Penerapan Scrum**

No	Isu	Keterangan
1	Kualitas Kode	<i>Development Team</i> seperti <i>Programmer</i> memiliki tenggat waktu yang pendek untuk menghasilkan kualitas kode yang baik bagi pengembangan sistem. Sehingga ditemukan banyak <i>bug</i> yang terdeteksi pada aplikasi <i>SonarQube</i> .
2	Durasi <i>Sprint</i>	Durasi <i>sprint</i> memainkan peran penting dalam efektivitas <i>scrum</i> . Durasi <i>sprint</i> minimum yang diterapkan memiliki durasi satu minggu dan terdapat masalah yang teridentifikasi dari penerapan durasi ini.
3	Proses Rilis	Pada metode <i>scrum</i> adalah proses rilis/proses <i>deployment</i> dari sistem yang telah dikembangkan. Masalah yang ditemukan adalah pengembangan proyek yang tidak selesai dengan tepat waktu. Hal ini berkaitan dengan permintaan klien mengenai kebutuhan proyek. Fitur yang sudah dikembangkan oleh <i>Development Team</i> harus dirubah kembali karena dirasa belum sesuai dengan apa yang diminta oleh klien.
4	Kurangnya Pelatihan <i>Scrum</i>	Anggota tim tidak memiliki pelatihan dan pengenalan terkait <i>scrum</i> dan tidak mengetahui proses <i>scrum</i> (khususnya apabila ada pegawai baru/magang).
5	Gantt Chart	<i>Gantt chart</i> digunakan untuk untuk membantu <i>Project Manager</i> dan <i>Analyst</i> memastikan bahwa semua kegiatan yang berkaitan dengan proyek telah direncanakan, urutan kinerja telah diperhitungkan, perkiraan waktu kegiatan telah tercatat serta keseluruhan waktu proyek telah dibuat. Namun, <i>gant chart</i> ini sering diabaikan, tidak dianalisis dan tidak di <i>update</i> .
6	Komunikasi	Komunikasi yang semakin meningkat dari penerapan metode ini terdapat beberapa anggota tim merasakan gangguan dalam pekerjaannya. Tim tidak bisa berkonsentrasi dengan baik. Komunikasi yang efektif juga menambah seringnya pertemuan. Beberapa anggota tim menganggap menghadiri semua rapat tidak perlu, sehingga tim tidak menunjukkan minat dalam rapat yang tidak terkait langsung dengan mereka.

perubahan yang diminta oleh *user* telah disesuaikan berdasar *sprint review* yang telah dilaksanakan.

- b. Error klik button “Ready to Evaluate” di menu Project Progress. Gambar 8 menampilkan halaman di menu Project Evaluation. Di halaman ini, *user* khususnya *role Admin, Reviewer Work Plan* dapat melakukan evaluasi untuk proyek-

proyek. Untuk bisa mengakses halaman di menu Project Evaluation, *employee* melakukan pengisian data terlebih dahulu di halaman Project Progress. Apabila proyek yang didaftarkan siap untuk dilakukan evaluasi, maka di halaman tersebut *user* dapat menekan tombol Ready to Evaluate.

- c. *Rabbit holes* yang terdapat pada proses bisnis menu Failure Management. Gambar 9 menampilkan hasil analisis untuk proses bisnis bagi menu Failure Management. Hasil analisis tersebut berupa data yang dibutuhkan hingga fungsi-fungsi untuk memperjelas penggunaan fitur di menu Failure Management dan dicatat pada aplikasi manajemen proyek yaitu Active Collab. *Rabbit holes* yang telah terdefinisi dipindahkan pada *Product Backlog* dan akan dijelaskan secara rinci serta diberi label sesuai dengan *task list* sehingga pekerjaan yang dilakukan sesuai dengan prioritas yang telah ditentukan.

#### 4.1 Dampak Penerapan Scrum

Konsep *scrum* mengevaluasi dan meningkatkan hasil produk pengembangan dengan menyesuaikan proses agar lebih efektif. Hal ini dapat berdampak untuk meningkatkan siklus hidup produk. Pendekatan *scrum* dilakukan dengan mengakomodasi perubahan karena tim *scrum* bekerja dalam *sprint*, dan setiap perubahan yang perlu dilakukan dapat diimplementasikan dengan mudah di *sprint* berikutnya. Namun, terdapat kendala yang ditemukan dalam penerapan *scrum* pada penelitian ini yakni berupa isu-isu penerapan metode *scrum* yang telah dicatat di Tabel 3.

#### 4.2 Solusi Isu Scrum

Solusi yang diusulkan untuk identifikasi isu yang diperoleh pada Tabel 3 dalam penerapan *scrum* terdapat pada Tabel 4.

**Tabel 4. Solusi penerapan Scrum**

No	Isu	Solusi
1	Kualitas Kode	Untuk programmer, tugas dapat dibagi menjadi beberapa tugas.
		Memindahkan sub-bagian dari tugas ke <i>sprint</i> berikutnya.
2	Durasi <i>Sprint</i>	Menambah jumlah <i>sprint</i> .
		Memprioritaskan pengerjaan <i>task</i> . Sehingga hal ini dapat digunakan untuk mengukur kemajuan dan kecepatan tim dalam pekerjaannya.
3	Proses Rilis	Menerapkan manajemen rilis/ <i>deployment</i> dalam tim sesuai dengan kebijakan <i>scrum</i> .
4	Kurangnya Pelatihan <i>Scrum</i>	Pelatihan <i>scrum</i> membantu mengenalkan tim tentang cara kerja <i>scrum</i> dan dalam meningkatkan kesadaran tentang peran-peran tim sehingga pada akhirnya tim akan tahu cara menjalankan dan mengekstrak manfaat <i>scrum</i> untuk tim dan produk mereka.
5	Gantt Chart	Melakukan <i>update</i> dan <i>monitoring</i> terhadap keberlangsungan dari proyek.
6	Komunikasi	Melakukan rapat resolusi dengan tujuan untuk penyelesaian masalah.

		Setelah resolusi dilakukan, anggota tim diberikan informasi tentang kesimpulan dari diskusi atau rapat. Resolusi juga harus ditambahkan dalam pembuatan <i>task</i> untuk membantu anggota tim untuk memahami persyaratan di <i>sprint</i> selanjutnya.
--	--	---

## 5. PENUTUP

Berdasarkan hasil pembahasan pada bagian sebelumnya maka terdapat beberapa kesimpulan sebagai berikut:

- a. MSS memiliki menu pengelolaan KPI yang bertujuan untuk dapat menentukan KPI tiap karyawannya. Selain itu, dapat menunjukkan hasil pencapaian dan tidaknya pada halaman menu Failure Management dari KPI yang telah diinput dan melalui tahapan *review*, *advise* dan *approve*.
- b. MSS sangat membantu pada penilaian karyawan dikarenakan berbagai tahapan telah dilakukan secara digital dan memudahkan dalam pertukaran data penilaian karyawan tersebut. Pada tahapan *review*, *advice*, dan *approve* KPI menjadi lebih cepat karena tahapan tersebut tidak perlu menyerahkan dokumen secara konvensional.
- c. Produktivitas tim lebih tinggi dalam pengembangan proyek.
- d. Dengan *scrum*, proyek yang berada dalam tahap pengembangan mampu beradaptasi. Perubahan dapat didukung dan diintegrasikan ke dalam proyek yang sedang berlangsung.
- e. Dengan *scrum*, tugas-tugas diprioritaskan berdasarkan urutan kepentingan seperti label yang terdapat pada *task list*. Sehingga, kebutuhan fitur dari proyek harus diselesaikan terlebih dahulu berdasar label yang telah ditentukan.
- f. *Scrum* memastikan penggunaan waktu yang efektif. Apabila pengembangan proyek besar dan kompleks dapat dibagi menjadi sprint yang mudah dikelola.
- g. Perkembangan proyek diberi label dan diuji selama *sprint review*.

Masih terdapat berbagai kekurangan dalam pelaksanaan pengembangan MSS dengan metode *scrum*. Pada penerapannya *scrum* telah banyak dimodifikasi pada tiap tahap-tahapnya. Kurangnya dokumentasi terhadap perubahan dari *scrum* tersebut membuat beberapa karyawan yang menggunakan metode ini kurang memahami proses implementasinya. Peneliti juga merasa kesulitan untuk memahami tahapan yang telah diubah. Kurangnya kajian terhadap literatur yang berkaitan dengan metode *scrum* menjadi salah satu kendala dalam memahami metode ini.

## 6. REFERENSI

- [1] Claudia Indriya Ningrum. 2021. Pengukuran Kinerja Perusahaan Depo Peti Kemas dengan Menggunakan Metode Performance Prism dan Analytical Hierarchy. 191.
- [2] Mulyadi, Y. 2017. Pengaruh Penempatan Kerja dan Pengembangan Karir Karyawan terhadap Kinerja Karyawan pada PT. Dhanar Mas Concern di Kabupaten Bandung. 18.
- [3] Krunal Bhavsar, Vrutik Shah, and Samir Gopalan. 2020. *Scrum: An Agile Process Reengineering in Software Engineering*. *International Journal of Innovative Technology*

and *Exploring Engineering* 9, 3: 840–848. DOI=<https://doi.org/10.35940/ijitee.c8545.019320>

- [4] Taufik Rizaldi. 2017. Implementasi Metodologi *SCRUM* dalam Pengembangan Sistem Pembayaran Elektronik Pada Usaha Mikro Kecil Menengah. *Cerebral cortex (New York, N.Y. : 1991)* 27, 1: 485–495. Retrieved from <https://publikasi.polije.ac.id/index.php/prosiding/article/view/236>
- [5] Firdaus, M. A. 2017. Implementasi Kerangka Kerja *Scrum* Pada Manajemen Pengembangan Sistem Informasi. *Seminar Nasional Teknologi Informasi dan Multimedia 2017* 1, 2: 283–288.
- [6] Imaduddin, Z., Saptono, H., Fauziah, S.T., Tawakal, H.A. and Hamzah, D. 2019. APLIKASI MONITORING PERKEMBANGAN JANIN (ANTENATAL CARE) DENGAN METODE *SCRUM* BERBASIS PERANGKAT MOBILE. *Jurnal Teknologi Terpadu*. 5, 1 (2019), 8.
- [7] Setiobudi, E. 2017. Analisis Sistem Penilaian Kinerja Karyawan Studi pada PT. Tridharma Kencana. *JABE (Journal of Applied Business and Economic)*. 3, 3 (2017), 170. DOI=<https://doi.org/10.30998/jabe.v3i3.1768>.
- [8] Rawsthorne, D. and Shimp, D. 2019. *Scrum handbook: Single team scrum* (STS). 3Back LLC
- [9] Schwaber, K. and Sutherland, J. 2017. *Scrum guides*
- [10] *Scrum*. D. Claire. Retrieved May 24, 2021, from: <https://www.atlassian.com/agile/scrum/>
- [11] *scrum.org*, What is *scrum*?. Diakses pada: 1 Juni 2021. [Online]. <https://www.scrum.org/resources/what-is-scrum>
- [12] *nutcache.com*, What is *Scrum*? Methodology and Project Management. Diakses pada: 1 Juni 2021. [Online]. Tersedia: <https://www.nutcache.com/blog/what-is-scrum-methodology-and-project-management/>
- [13] Shastri, Y., Hoda, R. and Amor, R. 2021. The Role of The Project Manager in Agile Software Development Projects. *Journal of Systems and Software*. 173, (2021), 110871. DOI=<https://doi.org/10.1016/j.jss.2020.110871>.
- [14] Rainer, R. K., Prince, B., Splettstoesser-Hogeterp, I., Sanchez-Rodriguez, C., & Ebrahimi, S. 2020. Introduction to information systems. John Wiley & Sons
- [15] Scott, R.T. and de Ziegler, D. 2020. Introduction: Key Performance Indicators in Assisted Reproductive Technologies. *Fertility and Sterility*. 114, 1 (2020), 4–5. DOI=<https://doi.org/10.1016/j.fertnstert.2020.04.057>.
- [16] Putri, A.R.T., Priyandari, Y. and Liquiddanu, E. 2019. Design of E-commerce Competency Improvement Program for Batik SMEs in Surakarta. *IOP Conference Series: Materials Science and Engineering*. 598, 1 (2019). DOI=<https://doi.org/10.1088/1757-899X/598/1/012067>.
- [17] Westland, J. 2000. *The Project Management Life Cycle: A Complete Step-by-step Methodology for Initiating Planning Executing and Closing the Project*. Kogan Page Publishers.