

Implementasi Scrum pada Pengembangan Modul Leadership Quality Feedback (LIQUID) (Studi Kasus: Pengembangan Aplikasi KOMANDO)

Safira Yuniar Putri Buana
Program Studi Sarjana Informatika
Universitas Islam Indonesia
Yogyakarta, Indonesia
17523020@students.uui.ac.id

Hanson Prihantoro Putro
Program Studi Sarjana Informatika
Universitas Islam Indonesia
Yogyakarta, Indonesia
hanson@uui.ac.id

Abstract—Scrum merupakan salah satu kerangka kerja yang mengimplementasikan agile development. Secara spesifik, scrum dirancang untuk memenuhi keinginan dan kebutuhan pelanggan yang berubah dengan cepat. Hanya saja, terkadang pengimplementasian kerangka kerja scrum tidak mengikuti panduan scrum dengan sempurna. Dalam penelitian ini, scrum diterapkan pada pengembangan modul Leadership Quality Feedback atau LIQUID. Penerapannya dibantu dengan tools manajemen proyek dan melakukan penambahan komponen dalam beberapa acara scrum. Modul LIQUID akan memfasilitasi kegiatan umpan balik dalam suatu organisasi agar lebih efisien, cepat, dan mudah dikelola. Dari hasil penelitian, diperoleh bahwa kemampuan scrum cocok dengan kebutuhan pengembangan modul LIQUID karena pengembangannya memerlukan banyak interaksi dengan klien berupa penambahan dan perubahan kebutuhan yang harus segera diimplementasikan dalam aplikasi. Pemenuhan kebutuhan dan penyelesaian masalah dapat berjalan secara efektif dan efisien.

Keywords—scrum, pengembangan aplikasi, modul Leadership Quality Feedback (LIQUID)

I. PENDAHULUAN

Kepemimpinan merupakan komponen yang sangat penting dalam suatu organisasi karena di dalamnya terdapat seseorang yang memiliki kemampuan untuk memengaruhi, memberi petunjuk dan mampu menentukan individu untuk mencapai tujuan organisasi. Maka, untuk mencapai tujuan tersebut dibutuhkan seorang pemimpin (*leader*) [1]. Peran seorang pemimpin sangat menentukan keberhasilan dalam mencapai tujuan yang telah ditetapkan bersama. Peran ini diterapkan juga pada salah satu perusahaan listrik negara yaitu PT PLN (Persero). Melalui pelaksanaan kegiatan *Leadership Quality Feedback* (LIQUID), para pemimpin diharapkan dapat meningkatkan kualitas kepemimpinan.

LIQUID merupakan kegiatan yang bertujuan untuk mendapatkan umpan balik atau *feedback* dari *user* untuk mengetahui kualitas kepemimpinan serta mendapatkan masukan dan saran yang akan menjadi bahan evaluasi bagi peningkatan dan perbaikan layanan [2]. Pemberian *feedback* dilakukan oleh bawahan kepada atasan, tujuannya agar atasan dapat mengenali perilaku atau kemampuan apa yang masih lemah dan dapat ditindaklanjuti sebagai upaya pengembangan diri. Awal mula kegiatan LIQUID dilaksanakan secara manual, dengan mengumpulkan peserta di suatu ruangan dan diberikan arahan untuk menuliskan *feedback* dengan *sticky notes* yang akan ditempelkan pada sebuah papan. Namun, cara manual seperti ini akan memakan banyak waktu ketika mengumpulkan peserta dan mengolah data. Oleh karena itu, proses LIQUID yang selama ini dilakukan secara manual, perlu di-*improve* dengan membuat sebuah modul yang masuk

dalam aplikasi Komando. Modul LIQUID akan memfasilitasi kegiatan umpan balik supaya lebih efisien, cepat, dan mudah dikelola.

Modul LIQUID adalah modul yang digunakan untuk memfasilitasi kegiatan umpan balik dari *user* yang selama ini masih dilakukan dengan cara manual, menjadi berbasis aplikasi yang lebih efisien, cepat, dan mudah dikelola. Modul LIQUID dikembangkan di dalam aplikasi Komando (Komunikasi Manajemen dan Budaya Organisasi) yang merupakan aplikasi untuk mengorganisir internalisasi pedoman perilaku [3]. Aplikasi ini digunakan sebagai pengumpulan pernyataan komitmen seluruh pegawai PLN bahwa telah menerima, membaca, dan memahami, serta melaksanakannya. Pengguna aplikasi Komando mencapai 50 ribu pengguna aktif dan terdapat lebih dari 50 unit daerah dari seluruh Indonesia.

Dalam pelaksanaan pengembangan modul LIQUID, akan terdapat banyak penambahan dan perubahan fitur yang diminta oleh klien. Maka, perlu diimplementasikan sebuah metode pengembangan agar pemenuhan kebutuhan dapat tercapai. Metode pengembangan yang diimplementasikan adalah metode pengembangan *agile* dengan kerangka kerja *scrum*. *Scrum* merupakan kerangka kerja yang cocok diimplementasikan dalam pengembangan modul LIQUID karena *scrum* digunakan untuk pekerjaan yang kompleks di mana permasalahan tidak dapat diprediksi [4]. Namun, terkadang praktik *scrum* bersifat dinamis atau berbeda-beda menyesuaikan dengan kondisi dan situasi dari sebuah perusahaan. Kondisi seperti itu disebut dengan istilah “*ScrumBut*” [5].

Penelitian ini dimaksudkan untuk mengetahui cara implementasi kerangka kerja *scrum* pada pengembangan modul LIQUID. Diharapkan penelitian ini dapat memberikan manfaat apabila ingin mengembangkan suatu sistem dengan metodologi yang serupa.

II. KAJIAN PUSTAKA

Terdapat beberapa penelitian yang telah dilakukan sebelumnya terkait implementasi kerangka kerja *scrum*. Dalam sebuah penelitian yang mengimplementasikan *scrum* pada proyek manajemen, dijelaskan bahwa kerangka kerja *scrum* sangat membantu dalam pengembangan perangkat lunak. Dengan pengimplementasian kerangka kerja *scrum*, akan memudahkan dalam *monitoring task* dan kerja tim secara *real time* [6]. Penelitian lain juga berkaitan dengan *scrum* yang dilakukan pada pengembangan aplikasi untuk mengelola data kegiatan organisasi. Hasil dari penelitian tersebut menunjukkan bahwa *scrum* dapat mengatasi

perubahan kebutuhan yang terjadi di tengah proses pengembangan [12]. Selain itu, penelitian tentang pengembangan sistem informasi seleksi masuk mahasiswa juga menerapkan *scrum* di dalamnya. Hasil dari penelitian tersebut *scrum* sangat tepat untuk pengembangan sistem yang membutuhkan kecepatan atau dengan kata lain waktu yang singkat [13].

Penelitian yang mengkaji peran *Trello* dalam adopsi *Agile Scrum* pada Pengembangan Sistem Informasi Kesehatan menjelaskan bagaimana pengimplementasian *scrum* dengan alat bantu *Trello*. Dalam penelitian ini, *Trello* sangat bermanfaat dalam implementasi *scrum* walaupun tidak semua artefak *scrum* diterapkan seperti pada *sprint review* dan ketiadaan *burn down chart*. Penelitian ini mendukung pengimplementasian *scrum* dengan praktik yang dinamis atau berbeda-beda [7].

Terkait kegiatan umpan balik, terdapat penelitian yang menjelaskan pengaruh umpan balik dalam suatu organisasi [8]. Dalam penelitian ini, keberadaan umpan balik dapat memengaruhi kinerja karyawan. Dengan adanya umpan balik, kesalahan dapat dikoreksi dan kinerja karyawan dapat ditingkatkan. Selain itu, penelitian [9] tentang kepemimpinan, menjelaskan bahwa peran pemimpin mampu memengaruhi kinerja karyawan.

Dari penelitian yang ada, telah dilakukan implementasi *scrum* pada pengembangan aplikasi dan proyek manajemen. Pada penelitian yang telah diuraikan, terdapat satu penelitian yang menjelaskan manfaat implementasi *scrum*, dan satu penelitian mengenai ketidaksempurnaan dalam implementasi *scrum*. Selain itu, terdapat penelitian mengenai kegiatan umpan balik dan kepemimpinan dalam suatu organisasi. Dalam penelitian tersebut, kegiatan umpan balik dan peran pemimpin dapat memengaruhi kinerja karyawan. Pada penelitian yang ada, belum ada yang melakukan penggabungan antara implementasi *scrum*, kegiatan umpan balik, dan peran kepemimpinan. Oleh karena itu, pada penelitian ini akan dilakukan implementasi *scrum* pada kegiatan umpan balik kepemimpinan dalam suatu organisasi.

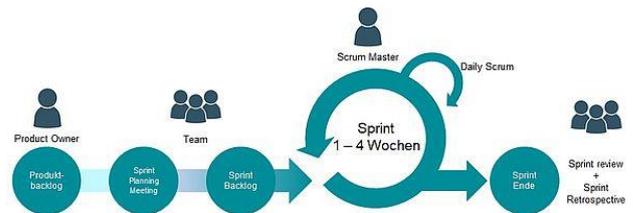
III. METODOLOGI

Data yang digunakan dalam penelitian ini dikumpulkan dengan dua tahapan yaitu wawancara dan dokumentasi. Selama penelitian berlangsung, wawancara dilakukan dengan *project manager* dengan mengajukan beberapa pertanyaan di antaranya sebagai berikut: (1) Bagaimana proses bisnis dari modul LIQUID? (2) Fitur apa saja yang dikembangkan dalam modul LIQUID? (3) Apa saja kendala yang dihadapi ketika mengembangkan modul LIQUID?

Selain dengan wawancara, untuk mendapatkan informasi mengenai modul LIQUID dilakukan dengan tahapan dokumentasi. Informasi didapatkan melalui dokumen-dokumen yang telah diberikan oleh *stakeholder* sebelumnya, contohnya seperti dokumen teknis sistem. Hal ini sangat membantu dalam proses *mapping* fitur dan mengetahui spesifikasi teknis aplikasi Komando.

Dalam penelitian ini, tidak semua panduan *scrum* diterapkan secara sempurna. Secara garis besar langkah-langkah mengimplementasikan kerangka kerja *scrum* [10] dalam pengembangan modul LIQUID yaitu: (1) Menentukan *product backlog*, (2) Membuat *Sprint Planning*, (3) Membuat *Sprint Backlog*, (4) Melakukan *Daily Scrum*, (5) Melakukan

Sprint Review, (6) Melakukan *Sprint Retrospective*. Untuk langkah yang lebih detail dapat dilihat pada Gambar 1.



Gambar 1. Alur Kerangka Kerja Scrum [11].

A. Product Backlog

Product backlog merupakan tahapan pertama *scrum*. Pada tahap ini, daftar kebutuhan *user* akan diurutkan berdasarkan prioritas yang harus dikerjakan dalam satu sprint atau lebih. Prioritas kebutuhan akan ditentukan oleh *Product Owner* setelah berdiskusi dengan *stakeholder*. Prioritas ditentukan berdasarkan tingkat kebutuhan dari fitur yang diminta atau kebutuhan yang akan segera digunakan. *Product backlog* akan berevolusi seiring dengan perkembangan produk dan lingkungan di mana produk tersebut dikembangkan [10].

B. Sprint Planning

Tahap selanjutnya, tim akan melakukan analisis pekerjaan yang akan dikerjakan dalam waktu sprint. Task atau pekerjaan dipilih oleh tim *scrum* dari *product backlog* sebelumnya. Dalam *sprint planning*, tim akan melakukan *breakdown* pada *product backlog* yang telah dipilih menjadi beberapa task kecil. Pada tahap ini, *scrum master* harus memastikan seluruh anggota tim memahami tujuannya [10].

C. Sprint Backlog

Sprint backlog berisi seluruh daftar pekerjaan yang telah disepakati pada *sprint planning*. Ketika ada task baru yang diperlukan, akan ditambahkan pada *sprint backlog*. Dalam *sprint backlog*, *development team* dapat memantau keadaan dari sisa pekerjaan yang telah direncanakan sebelumnya [10].

D. Daily Scrum

Tahap ini dilakukan apabila sprint telah berjalan di mana setiap individu tim akan mengikuti *meeting* yang dilakukan setiap hari selama 15 menit. Di tahapan ini, setiap individu akan memaparkan apa yang akan dikerjakan dalam waktu 24 jam ke depan. *Daily scrum* bertujuan untuk mengoptimalkan kolaborasi dan performa tim dengan melakukan inspeksi pada pekerjaan yang dilakukan semenjak *daily scrum* sebelumnya [10].

E. Sprint Review

Tahapan ini dilakukan setelah durasi sprint telah selesai. *Sprint* dianggap selesai apabila telah mencapai *definition of done* (DoD) yang telah ditetapkan dan disepakati sebelumnya. Pada tahap *sprint review*, tim akan mendemonstrasikan apa yang telah dikerjakan selama sprint yang telah berlangsung kepada *stakeholder*. *Product owner* juga akan menjelaskan mengenai pekerjaan apa yang sudah dan belum selesai. Dalam *sprint review*, hadirin berkolaborasi untuk menentukan pekerjaan pada *sprint planning* berikutnya [10].

F. Sprint Retrospective

Tahap ini dilakukan setelah *sprint review* terlaksana. Pada tahap ini *scrum team* akan memilih pekerjaan yang akan dilakukan di *sprint planning* berikutnya. Pekerjaan yang akan

dikerjakan di *sprint* berikutnya mengacu pada *scope of work* yang sudah disepakati pada awal proyek.

G. Weekly Report

Weekly report merupakan laporan kegiatan proyek yang dikirimkan kepada manajemen perusahaan dan klien setiap minggu. *Weekly report* mencakup capaian dan progress proyek yang terdiri dari kendala, solusi, *summary progress*, *sprint next week*, dan *burn down chart* dalam bentuk *line chart*. *Weekly report* akan dibuat dan dikirimkan oleh *system analyst*. Tujuan dari pembuatan *weekly report* yaitu agar manajemen perusahaan dan klien dapat mengetahui sejauh mana *progress* dari pengembangan proyek yang dikerjakan.

IV. HASIL DAN PEMBAHASAN

Pada tahapan ini dilakukan penerapan pengembangan modul LIQUID dengan *scrum* mulai dari menentukan *product backlog* hingga beberapa acara-acara *scrum* yang mendukung. Sebelum masuk ke tahap menentukan *product backlog*, implementasi kerangka kerja *scrum* pada pengembangan modul LIQUID terbagi atas peran-peran sebagai berikut:

- *Project Manager* (PM), sebagai pemimpin proyek yang bertanggung jawab atas pengelolaan produk secara keseluruhan.
- *Business Analyst*, bertanggung jawab untuk menganalisis bisnis yang akan dikembangkan.
- *System Analyst*, bertanggung jawab pada keberhasilan suatu proyek dengan mengumpulkan kebutuhan user dan mencari solusinya.
- *Programmer*, terdiri dari dua level yaitu junior dan senior, keduanya bertanggung jawab atas eksekusi dan implementasi perancangan kebutuhan dalam bentuk kode program.
- *Quality Assurance* (QA), bertanggung jawab untuk melakukan pengujian terhadap produk yang dikembangkan.
- *Technical Document*, bertanggung jawab pada dokumen-dokumen yang akan dikirim pada klien.

A. Product Backlog

Product Backlog merupakan sebuah dokumen yang berisi daftar prioritas kebutuhan yang akan dikerjakan pada *sprint*. Pada penelitian yang telah dilakukan sebagaimana bisa dilihat pada tabel 1, *product backlog* berisi kolom *scope of work* (SOW), fitur dan prioritas. Kolom SOW merupakan ruang lingkup pekerjaan akan dikerjakan. Kolom fitur berisi jenis fitur apa saja yang akan dikerjakan untuk mencapai SOW. Pada penamaan fitur akan disertakan menu atau *submenu* tempat fitur akan dikerjakan dan judul singkat fitur. Kolom prioritas berisi tingkat kebutuhan fitur yang dikerjakan, apabila pengguna ada yang melaporkan *bugs*, maka pada prioritas akan tertulis *Bug*. Dalam penelitian ini terdapat tiga level tingkatan di antaranya sebagai berikut:

- *Must to have*, fitur yang termasuk dalam prioritas ini berarti harus dikerjakan, apabila tidak dikerjakan akan berpengaruh pada hasil akhir produk.
- *Good to have*, fitur yang termasuk dalam prioritas ini akan memberikan nilai tambah bagi klien apabila dikerjakan.

- *Nice to have*, fitur yang termasuk dalam prioritas ini hanyalah tambahan pemanis dan fitur ini dapat dikerjakan atau tidak dikerjakan.

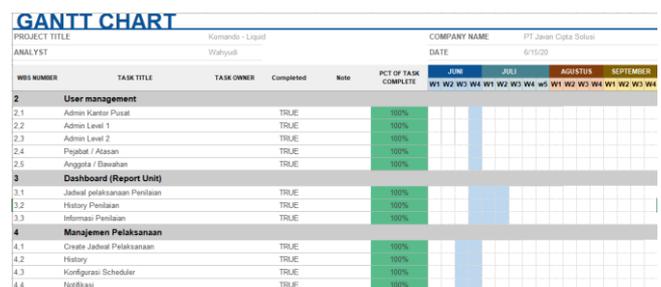
Setiap fitur akan ditentukan prioritas berdasarkan tingkatannya oleh *Project Manager* dan *Business Analyst*.

TABEL I. PRODUCT BACKLOG

SOW : Review Workshop Batch #2		
No	Fitur	Prioritas
1	[Dashboard] Status Terakhir masih belum sesuai	Bug
2	[Dashboard] Grup Peserta Liquid berdasarkan Jenjang Jabatan	Must to have
3	[Dashboard] Add Bawahan langsung dari daftar peserta Liquid	Must to have
..
6	[Dashboard] Fitur Send Notification untuk Peserta Liquid + Permission-nya	Good to have
7	[Dashboard/Detail History Penilaian] Urutkan Kelebihan dan Kekurangan DESC dan tambahkan kolom Nomor	Nice to have

B. Sprint Planning

Tahap *sprint planning* dapat digambarkan sebagai tahap analisis pada masalah yang akan diselesaikan. Analisis yang dilakukan meliputi: penjelasan alur kerja aplikasi, merinci *item product backlog*, menentukan tugas untuk masing-masing anggota, menentukan durasi *sprint* dan menentukan *sprint goal*. Pada penelitian ini, *sprint planning* dibuat berdasarkan grafik *gantt chart* yang dibuat dengan *excel* (lihat Gambar 2). Meskipun metode yang dipakai adalah *scrum*, namun penggunaan *gantt chart* tetap masih digunakan. *Gantt Chart* akan memaparkan waktu pengerjaan pada setiap ruang lingkup atau SOW yang ada. Dengan adanya *gantt chart* proses *monitoring* ruang lingkup atau SOW proyek yang dikerjakan akan lebih mudah. *System analyst* akan meng-*update* setiap *progress* di dalam *gantt chart*. Selain penerapan *gantt chart*, pembagian *task* yang telah dipilih dari *product backlog* untuk *sprint* ditentukan oleh *system analyst*. Hal ini berbeda dengan panduan *scrum* dimana *development team* akan mengambil keputusan sepenuhnya terkait *task* yang akan dikerjakan [10].



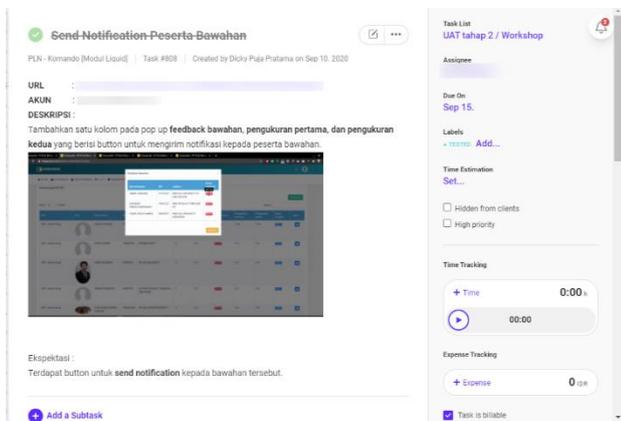
Gambar 2. Tampilan Gantt Chart.

C. Sprint Backlog

Sprint backlog berisi daftar pekerjaan hasil kesepakatan dari *sprint planning* yang akan dikerjakan selama satu sprint. Dalam penelitian ini, pembuatan *sprint backlog* dibantu dengan aplikasi proyek manajemen yaitu *ActiveCollab*. Task pada *list product backlog* yang telah disepakati pada *sprint planning* akan ditarik ke *list sprint backlog* seperti pada Gambar 3. Selain itu, *system analyst* juga akan memberikan ulasan rinci pada setiap *task* yang ada pada *list sprint backlog* seperti deskripsi, ekspektasi, skenario, *url* halaman, dan akun login yang dipakai. Gambar 4 merupakan representasi 1 *task* yang ada pada *list sprint backlog*.



Gambar 3. List Sprint Backlog



Gambar 4. Representasi 1 task collab

D. Daily Scrum

Selama sprint berlangsung, akan diadakan *daily scrum* yang umumnya berdurasi 15 menit untuk setiap pertemuannya. Pada praktiknya, durasi bergantung pada jumlah anggota dan jenis pembahasannya. Komponen pembahasan dalam *daily scrum* di antaranya: *task* apa saja yang sudah dikerjakan kemarin, kendala apa yang ditemui selama pengerjaan, dan *task* yang akan dikerjakan berikutnya.

Setelah *system analyst* menguraikan *task* yang ada pada *list sprint backlog* seperti pada Gambar 4, project manager akan melakukan proses monitor dan kontrol melalui *ActiveCollab*. Pada Collab setiap *task* yang sedang dieksekusi akan diberi label *In Progress* (lihat Gambar 3). Ketika *task* diberi label *Ready To Test*, maka *task* tersebut sudah bisa diuji oleh QA dan *task* akan diberi label *Tested* seperti pada Gambar 5.



Gambar 5. Contoh task siap di testing.

E. Sprint Review

Sprint review dilaksanakan pada akhir masa sprint. Peserta dalam acara ini adalah *scrum team* dan *stakeholder*. *Project Manager* menjelaskan *product backlog* yang telah dan belum diselesaikan, kemudian *system analyst* akan mendemokan hasil pekerjaan yang telah diselesaikan. Pada praktiknya, *sprint review* dilaksanakan setiap dua minggu sekali oleh tim *scrum* dan *stakeholder*. Pada *sprint* terakhir, *system analyst* akan mengacu dokumen *User Acceptance Test (UAT)* ketika mendemonstrasikan hasil pekerjaan. Dokumen UAT berisi *test case scenario* dari semua *product backlog* yang telah dikelompokkan berdasarkan ruang lingkup pekerjaannya seperti pada Gambar 6. *Stakeholder* mengisi status dan catatan (apabila ada) yang sudah disediakan dalam dokumen UAT.

Dalam tahap ini, tanggapan dan masukan akan ditampung dan dicatat oleh *system analyst*. Apabila perlu perubahan pada fitur yang telah rilis akan masuk pada *sprint* berikutnya. Namun, perubahan-perubahan yang diminta oleh *stakeholder* akan ditinjau pada tahap *sprint retrospective*.

Menu	Sub Menu	ID Testing	Prosedur Pengujian	Skenario	Hasil Yang Diharapkan	Hasil Aktual	Status	Tanggal Validasi (dd/mm/yy)
Tanggal Input	Create Liquid - Tanggal Input	B1	Passive Test		Berdapat ke halaman url sepe	Berdapat ke halaman url sepe	OK	-
		B2	Negative Test		Tanggal input dan sepepe akan otomatis sama dengan sepepe	Tanggal input dan sepepe akan otomatis sama dengan sepepe	OK	-
		B3	Negative Test		Muncul notifikasi "Tanggal Mula Pengukuran" pada saat input dan sepepe dengan "tanggal sepepe" "sepepe"	Muncul notifikasi "Tanggal Mula Pengukuran" pada saat input dan sepepe dengan "tanggal sepepe" "sepepe"	OK	-
		B4	Negative Test		Muncul notifikasi "Tanggal Mula Pengukuran" pada saat input dan sepepe dengan "tanggal sepepe" "sepepe"	Muncul notifikasi "Tanggal Mula Pengukuran" pada saat input dan sepepe dengan "tanggal sepepe" "sepepe"	OK	-

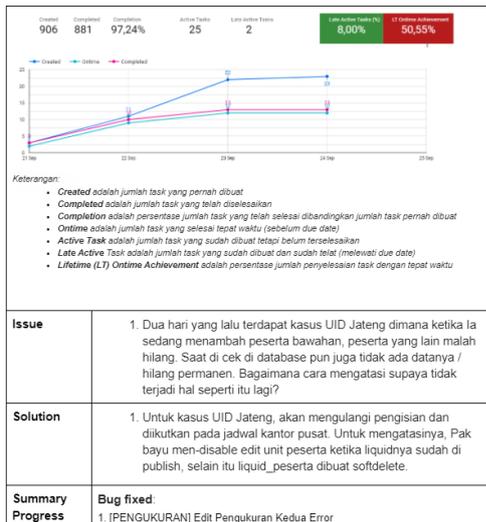
Gambar 6. Contoh Dokumen UAT

F. Sprint Retrospective

Sprint retrospective diadakan setelah dilaksanakannya *sprint review*. Pada tahap ini PM, *Business Analyst*, dan *System Analyst* meninjau perubahan-perubahan yang diminta oleh *stakeholder* dalam *sprint review*. Perubahan-perubahan ini ditinjau apakah masih tercakup dalam SOW atau tidak. Apabila perubahan tidak tercakup dalam SOW, maka perubahan tersebut tidak akan dikerjakan di sprint berikutnya. Namun, dalam praktiknya terdapat banyak perubahan yang disampaikan oleh *user* setelah terselenggara UAT di mana perubahan masih mencakup SOW. Perubahan yang banyak disampaikan berupa penambahan fitur dan perbaikan *bugs*, sedangkan waktu yang tersisa dalam kontrak tidak banyak. Sehingga, proses pengerjaannya melewati waktu batas kontrak. Hal ini bisa menjadi catatan bagi tim *scrum* untuk membatasi permintaan *user* pada pengembangan selanjutnya.

G. Weekly Report

Setiap hari Kamis, *system analyst* akan mengirimkan *weekly report* kepada manajemen perusahaan dan klien via email. Dalam *weekly report* mencakup kendala apa saja yang ditemukan pada 1 minggu, solusi apa yang dilakukan untuk menyelesaikan kendala tersebut dan pekerjaan apa saja yang dikerjakan dalam 1 minggu seperti pada Gambar 7. Selain itu, dalam *weekly report* juga mencantumkan *burn down chart* dan mencantumkan *sprint next week*.



Gambar 7. Contoh *Weekly Report*.

H. Diskusi

Hasil dari penelitian menunjukkan bahwa tidak semua acara *scrum* diterapkan secara sempurna. Hal ini dapat dilihat pada Tabel II yang memaparkan perbandingan antara panduan *scrum* atau *scrum guide* dengan *scrum* yang diterapkan dalam proyek ini. Pada kolom *scrum guide* menjelaskan implementasi *scrum* yang sesuai dengan panduan *scrum*, sedangkan kolom *scrumbut* menjelaskan implementasi *scrum* dalam penelitian ini.

TABEL II. PERBANDINGAN IMPLEMENTASI SCRUM DI SCRUM GUIDE DAN SCRUMBUT

No	Penerapan	Scrum Guide	Scrumbut
1	<i>Scrum Team</i>	Terdiri dari <i>Product Owner</i> , <i>Development Team</i> , dan <i>Scrum Master</i>	<i>Scrum</i> team terdiri dari 1 tim proyek dengan peran yang berbeda dari panduan <i>scrum</i> .
2	<i>Product Backlog</i>	Menentukan daftar prioritas kebutuhan yang ditentukan oleh <i>Product Owner</i> .	<i>Project Manager</i> menentukan tingkat prioritas kebutuhan sebelum masuk pada tahap <i>sprint planning</i> .
3	<i>Sprint Planning</i>	<i>Scrum team</i> akan menganalisis pekerjaan yang akan dikerjakan dalam waktu <i>sprint</i> . Komponen analisis meliputi: • Jumlah pekerjaan yang dipilih dari <i>Product Backlog</i> untuk <i>Sprint</i> merupakan keputusan <i>Development Team</i> sepenuhnya • Menentukan <i>sprint goal</i>	Tim <i>scrum</i> akan menganalisis terkait: <i>flow</i> aplikasi, merinci item <i>product backlog</i> , menentukan tugas untuk masing-masing anggota, menentukan durasi <i>sprint</i> dan menentukan <i>sprint goal</i> . <i>Product backlog</i> untuk <i>sprint</i> akan ditentukan oleh <i>system analyst</i> . Selain itu, terdapat <i>gantt chart</i> untuk proses <i>monitoring</i> pada ruang lingkup pekerjaan.
4	<i>Sprint Backlog</i>	<i>Sprint backlog</i> berisi seluruh daftar pekerjaan yang telah disepakati pada <i>sprint planning</i> . Ketika ada <i>task</i> baru yang diperlukan, akan ditambahkan pada <i>sprint backlog</i> oleh <i>development team</i> .	Ketika ada <i>task</i> baru, <i>system analyst</i> akan memberikan ulasan rinci dan menambahkan <i>task</i> tersebut ke dalam <i>sprint backlog</i> .

5	<i>Sprint Review</i>	<i>Sprint review</i> dilaksanakan diakhir <i>sprint</i> untuk meninjau apa saja pekerjaan yang telah dan belum diselesaikan. <i>Sprint review</i> dilakukan oleh tim <i>scrum</i> dan <i>stakeholder</i> .	Tim <i>scrum</i> melaksanakan <i>sprint review</i> setiap dua minggu sekali bersama <i>stakeholder</i> . Namun, pada <i>sprint</i> terakhir, pelaksanaan <i>sprint review</i> dilaksanakan bersamaan dengan UAT.
6	<i>Sprint Retrospective</i>	<i>Sprint retrospective</i> terselenggara setelah <i>sprint review</i> dan sebelum <i>sprint planning</i> berikutnya. Dalam <i>sprint retrospective</i> akan meninjau pekerjaan yang dilakukan dan yang tidak dilakukan untuk <i>sprint</i> selanjutnya.	Pada <i>sprint retrospective</i> , PM, <i>business analyst</i> , dan <i>system analyst</i> akan meninjau pekerjaan yang akan dikerjakan atau tidak berdasarkan SOW yang sudah ditetapkan sebelumnya.

Berdasarkan perbandingan di atas, terdapat perbedaan antara panduan *scrum* dan *scrum* yang diterapkan. Salah satu perbedaan yang menonjol adalah Peran *scrum*. Dalam penelitian ini, peran *scrum* tidak sama dengan yang ada di panduan *scrum*. Selain itu, terdapat beberapa acara *scrum*, seperti pada *sprint planning* ketika pembagian *task product backlog* untuk *sprint* dipilih oleh *system analyst*, *sprint review* yang terselenggara bersamaan dengan UAT, dan *sprint retrospective* tidak dilakukan oleh seluruh tim.

Meskipun demikian, tim *scrum* tetap menerapkan kerangka kerja *scrum* dengan menambah beberapa komponen pada beberapa acaranya, seperti penggunaan *gantt chart* pada *sprint planning*, penggunaan *Activecollab* untuk proyek manajemen, dan pembuatan *weekly report*. Penggunaan *gantt chart* sangat memudahkan dalam *monitoring* fitur mana yang telah dan belum diselesaikan. *Activecollab* sangat membantu dalam *monitoring* dan kontroling *task*. Pada *weekly report* menggambarkan capaian progress yang telah dilakukan setiap minggunya dan terdapat *burn down chart* yang berguna untuk melacak total sisa pekerjaan.

Implementasi *scrum* yang telah diterapkan juga efektif dan efisien. Dapat dikatakan efektif, karena dalam penerapan *scrum* terdapat *sprint* yang merupakan jantung dari *scrum*. Dengan adanya *sprint*, manajemen *task* menjadi lebih mudah dan cepat. Selain itu, implementasi *scrum* dapat dikatakan efisien karena dalam penerapannya menekankan kolaborasi tim. Tim dapat menentukan prioritas yang harus segera dikerjakan untuk hasil yang maksimal.

V. KESIMPULAN DAN SARAN

A. Kesimpulan

Implementasi *scrum* pada pengembangan proyek modul LIQUID berhasil diterapkan secara efektif dan efisien, walaupun melewati batas waktu yang sudah ditentukan. Dalam praktiknya, pengembangan modul LIQUID tidak semua mengikuti panduan *scrum* dengan sempurna. Namun, penerapan *scrum* dibantu dengan *tools* manajemen proyek yaitu *ActiveCollab* dan melakukan penambahan komponen dalam beberapa acara *scrum*, seperti adanya *weekly report* dan penggunaan *gantt chart* pada *sprint planning*. Selain itu, dalam pengembangan modul LIQUID memerlukan banyak interaksi dengan *klien*, sehingga banyak perubahan dan

penambahan kebutuhan yang harus segera diimplementasikan dalam aplikasi. Oleh karena itu, kemampuan *scrum* dapat dikatakan cocok dengan kebutuhan pengembangan modul LIQUID ini.

B. Saran

Dalam implementasi *scrum* pada pengembangan modul LIQUID diberikan saran demi kesempurnaan ke depannya. Saran yang diberikan seperti ketika *stakeholder* meminta penambahan fitur menjelang berakhirnya kontrak. Hal tersebut perlu dibatasi supaya target waktu dapat tercapai. Selain itu, pada *sprint retrospective* sebelumnya dilakukan oleh PM, *Business Analyst*, dan *System Analyst* yang menyebabkan perbedaan pendapat dengan *Programmer* terkait *task* yang diseleksi. Maka, saran untuk ke depannya *sprint retrospective* dilakukan oleh seluruh tim *scrum* supaya tidak ada *miscommunication*.

<http://repository.unjani.ac.id/repository/9ec3721ead4436680675214b6fc56247.pdf>.

- [13] U. Ependi, "Implementasi Model Scrum pada Sistem Informasi Seleksi Masuk Mahasiswa Politeknik Pariwisata Palembang," *Jurnal Informatika: Jurnal Pengembangan IT (JPIT)*, vol. 3, pp. 49-55, Januari 2018.

REFERENCES

- [1] M. Fahmi, S. Agung, and R. Rachmatullaily, "Gaya Kepemimpinan Dan Budaya Organisasi Terhadap Kinerja Karyawan," *Inovator*, vol. 7, no. 1, p. 90, 2018, doi: 10.32832/inovator.v7i1.1462.
- [2] Redaksi, "PLN Aceh Gelar Liquid Forum," 2017. <https://harianrakyataceh.com/2017/04/26/pln-aceh-gelar-liquid-forum/> (accessed Nov. 21, 2020).
- [3] PLN, "Sistem Komunikasi dan Budaya Organisasi PT. PLN." <https://budaya.pln.co.id/auth/login> (accessed Nov. 21, 2020).
- [4] H. Petter and K. Nils, "International Journal of Managing Projects in Business Article information :," *Int. J. Manag. Proj. Bus.*, vol. 7, no. 1, pp. 133-143, 2014.
- [5] V. P. Eloranta, K. Koskimies, and T. Mikkonen, "Exploring ScrumBut - An empirical study of Scrum anti-patterns," *Inf. Softw. Technol.*, 2016, doi: 10.1016/j.infsof.2015.12.003.
- [6] P. Adi, "Scrum Method Implementation in a Software Development Project Management," *Int. J. Adv. Comput. Sci. Appl.*, vol. 6, no. 9, pp. 198-204, 2015, doi: 10.14569/ijacsa.2015.060927.
- [7] T. Tohirin and S. R. Widiyanto, "Peran Trello dalam Adopsi Agile Scrum Pada Pengembangan Sistem Informasi Kesehatan," *Multinetics*, vol. 6, no. 1, pp. 32-39, 2020, doi: 10.32722/multinetics.v6i1.2763.
- [8] I. I. Wan, "Umpan Balik Sistem Pakar Sebagai Penilaian Kinerja Karyawan Pada PT. Multistrada Arah Sarana, Tbk," *BMAJ Bus. Manag. Anal. J.*, vol. 2, no. 1, pp. 52-67, 2019, doi: 10.24176/bmaj.v2i1.3211.
- [9] M. Nunung Ghoniyah, "PENINGKATAN KINERJA KARYAWAN MELALUI KEPEMIMPINAN,," *Jurnal Dinamika Manajemen*, vol. 2, pp. 118-129, 201
- [10] K. S. dan J. S., *Panduan Scrum*, 2017.
- [11] Vit, "Agile working," 2020. [Online]. Available: <https://www.vit.de/en/it/software-development/we-work-agil-scrum/> . [Diakses 22 November 2020].
- [12] A. Sholahudin, E. Paulus, and U. Padjadjaran Jl Raya Bandung-Sumedang Km, "Analisis dan Desain Pengembangan Program Aplikasi DMS dengan Metodologi Scrum Framework," no. September, pp. 47-52, 2017, [Online]. Available: