

SKRIPSI

IMPLEMENTASI *SCRUM* PADA PENGEMBANGAN MODUL LEADERSHIP QUALITY FEEDBACK (LIQUID) (STUDI KASUS: PENGEMBANGAN APLIKASI KOMANDO)



Disusun Oleh:

N a m a : Safira Yuniar Putri Buana

NIM : 17523020

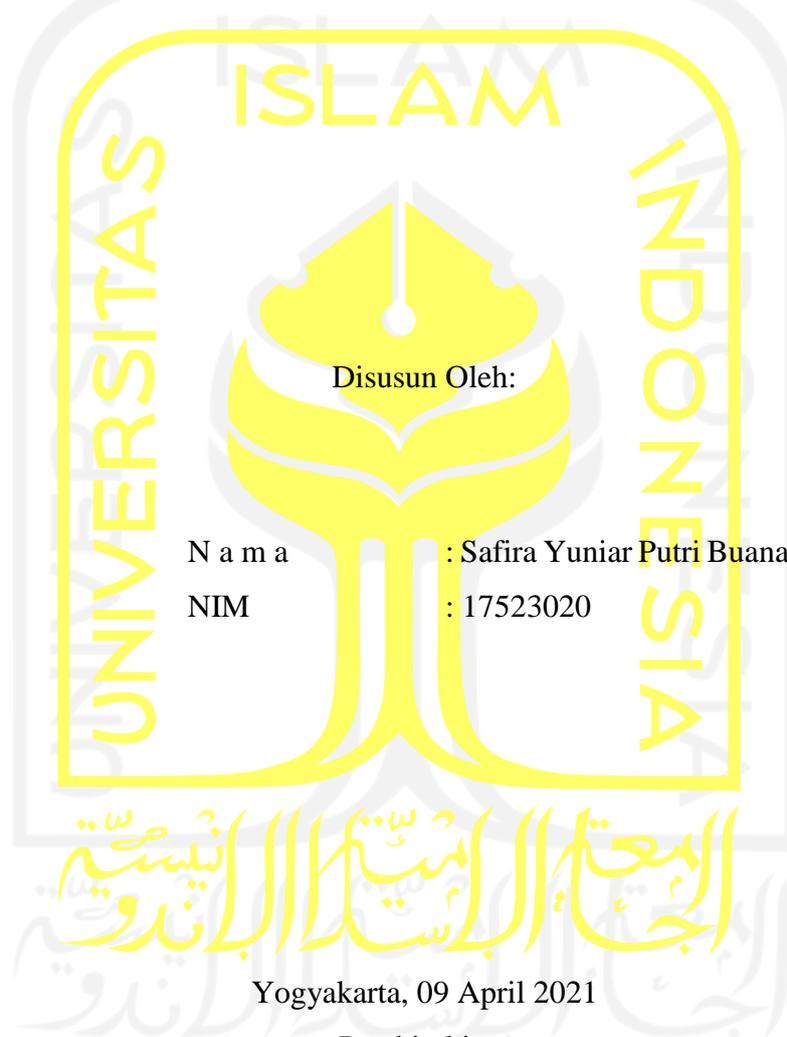
PROGRAM STUDI INFORMATIKA – PROGRAM SARJANA
FAKULTAS TEKNOLOGI INDUSTRI
UNIVERSITAS ISLAM INDONESIA

2021

HALAMAN PENGESAHAN DOSEN PEMBIMBING

**IMPLEMENTASI SCRUM PADA PENGEMBANGAN MODUL
LEADERSHIP QUALITY FEEDBACK (LIQUID) (STUDI
KASUS: PENGEMBANGAN APLIKASI KOMANDO)**

TUGAS AKHIR JALUR MAGANG



Pembimbing,

(Hanson Prihantoro Putro, S.T, M.T)

HALAMAN PENGESAHAN DOSEN PENGUJI

**IMPLEMENTASI SCRUM PADA PENGEMBANGAN MODUL
LEADERSHIP QUALITY FEEDBACK (LIQUID) (STUDI
KASUS: PENGEMBANGAN APLIKASI KOMANDO)****TUGAS AKHIR JALUR MAGANG**

Telah dipertahankan di depan sidang pengujian sebagai salah satu syarat untuk memperoleh gelar Sarjana Komputer dari Program Studi Informatika di Fakultas Teknologi Industri Universitas Islam Indonesia

Yogyakarta, 05 Mei 2021

Tim Penguji

Ketua Penguji

Hanson Prihantoro Putro, S.T., M.T.

Anggota 1

Kholid Haryono, S.T., M.Kom.

Anggota 2

Beni Suranto, S.T., M.Soft.Eng.

Mengetahui,

Ketua Program Studi Informatika – Program Sarjana

Fakultas Teknologi Industri

Universitas Islam Indonesia



(Dr. Raden Teduh Dirgahayu, S.T., M.Sc.)

HALAMAN PERNYATAAN KEASLIAN TUGAS AKHIR

Yang bertanda tangan di bawah ini:

Nama : Safira Yuniar Putri Buana

NIM : 17523020

Tugas akhir dengan judul:

**IMPLEMENTASI SCRUM PADA PENGEMBANGAN MODUL
LEADERSHIP QUALITY FEEDBACK (LIQUID) (STUDI
KASUS: PENGEMBANGAN APLIKASI KOMANDO)**

Menyatakan bahwa seluruh komponen dan isi dalam tugas akhir ini adalah hasil karya saya sendiri. Apabila dikemudian hari terbukti ada beberapa bagian dari karya ini adalah bukan hasil karya sendiri, tugas akhir yang diajukan sebagai hasil karya sendiri ini siap ditarik kembali dan siap menanggung resiko dan konsekuensi apapun.

Demikian surat pernyataan ini dibuat, semoga dapat dipergunakan sebagaimana mestinya.

Yogyakarta, 09 April 2021



(Safira Yuniar Putri Buana)



HALAMAN PERSEMBAHAN

Laporan Tugas Akhir ini saya persembahkan untuk diri saya, kedua orang tua, dan teman-teman yang selalu memberikan motivasi, dorongan, dan dukungan kepada penulis, sehingga atas berkat dukungan dari semua pihak dapat saya jadikan motivasi untuk menyelesaikan Tugas Akhir ini.



HALAMAN MOTO

“The one who falls and gets up, is so much stronger than one who never fell.”

– La La Anthony

“Pengalaman tidak bisa dipelajari tetapi harus dilalui.”

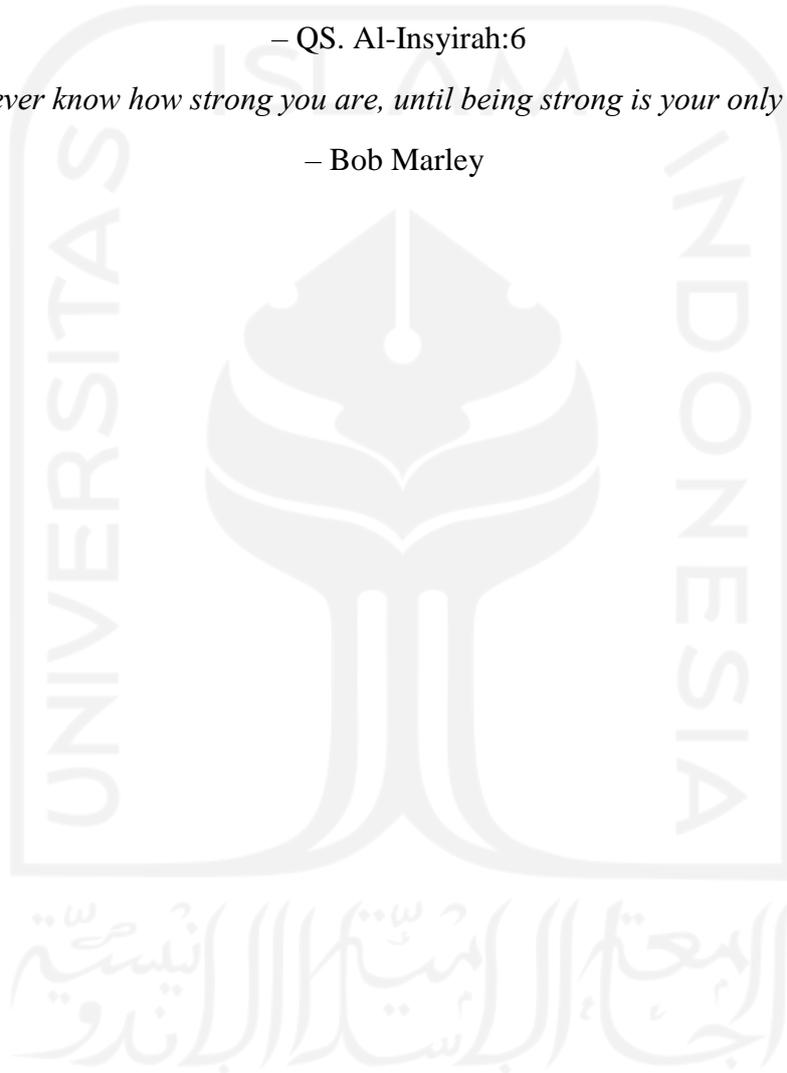
– BJ. Habibie

“Sesungguhnya sesudah kesulitan itu ada kemudahan”

– QS. Al-Insyirah:6

“You never know how strong you are, until being strong is your only choice”

– Bob Marley



KATA PENGANTAR

Assalamualaikum Wr. Wb.

Alhamdulillah, puji dan syukur kepada Allah SWT atas rahmat dan hidayah-Nya, sehingga penulis dapat menyelesaikan Laporan Tugas Akhir yang berjudul “Implementasi Scrum Pada Pengembangan Modul *Leadership Quality Feedback* (Studi Kasus: Pengembangan Aplikasi Komando)” ini. Laporan ini disusun sebagai bukti pelaksanaan magang serta untuk memenuhi kewajiban penulis dalam memperoleh gelar sarjana pada Program Studi Informatika, Fakultas Teknologi Industri, Universitas Islam Indonesia.

Dalam pelaksanaan magang di PT Javan Cipta Solusi, banyak hambatan dan rintangan yang telah dihadapi, namun pada akhirnya dapat terselesaikan dengan baik karena adanya bantuan dan dukungan dari orang-orang di sekitar saya. Untuk itu, penulis menyampaikan terimakasih yang sebesar-besarnya kepada:

1. Allah SWT yang telah memberikan kesehatan, keselamatan, dan kelancaran.
2. Kedua orang tua penulis yang telah memberikan doa, nasehat, dorongan, serta semangat yang luar biasa.
3. Bapak Hanson Prihantoro, S.T., M.T. selaku dosen pembimbing yang telah memberikan waktu, ilmu, arahan, dan bimbingan kepada penulis dalam menyelesaikan tugas akhir ini.
4. Bapak Wisnu, Bapak Dicky, Bapak Bayu, Mba Nurul, Mba Erikha, Mba Lulu, dan karyawan lain di PT Javan Cipta Solusi yang telah memberikan kesempatan, ilmu, dan bantuan kepada penulis selama proses magang berlangsung.
5. Segenap dosen informatika yang telah memberikan ilmunya dengan baik selama masa perkuliahan.
6. Lintang Digdoyo yang selalu memberikan waktu, nasehat, dukungan, dan bantuan kepada penulis untuk dapat menyelesaikan Tugas Akhir.
7. Shinta Dewi Kusumaningrum dan Syarifah Elza Ramadhania selaku teman dekat penulis yang selalu memberikan semangat, serta mendengarkan keluh kesah penulis dengan sabar.
8. Rekan rekan seperjuangan magang (Lia, Rei, Abyanda, dan Haqi) yang telah membantu, memberi dukungan, membagi pengetahuan, serta mewarnai kehidupan magang penulis.
9. Teman-teman PIXEL yang telah memberikan bantuan, dukungan, serta mewarnai kehidupan penulis pada masa perkuliahan.

10. Seluruh pihak yang telah membantu dalam menyelesaikan Tugas Akhir.

Atas bantuan dari berbagai pihak, penulis dapat menyelesaikan Laporan Tugas Akhir dengan sebaik mungkin. Namun, penulis menyadari bahwa laporan ini masih jauh dari kata sempurna, sehingga penulis sangat membutuhkan kritik dan saran yang membangun demi kesempurnaan Laporan Tugas Akhir ini. Akhir kata, semoga laporan ini bermanfaat bagi kita semua.

Wassalamualaikum Wr. Wb.

Yogyakarta, 09 April 2021



(Safira Yuniar Putri Buana)



SARI

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. Hal ini disebabkan oleh adanya peraturan dan budaya organisasi yang sudah ada dan tidak sejalan dengan aturan *scrum*. Maka, perlu dilakukan pembahasan terkait perbedaan *scrum* yang telah diterapkan dengan buku panduan *scrum* atau *Scrum Guide*.

Dalam penelitian ini, *scrum* diterapkan pada pengembangan modul *Leadership Quality Feedback* atau LIQUID. Penerapannya dibantu dengan beberapa *tools* manajemen proyek pada 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.

Kata kunci: *Scrum*, pengembangan aplikasi, modul *Leadership Quality Feedback* (LIQUID), *ScrumBut*

GLOSARIUM

<i>Agile</i>	metodologi dalam pengembangan perangkat lunak
<i>Burn-up Chart</i>	grafik yang menunjukkan perkembangan proyek secara berkala
<i>Change Request</i>	dokumen yang berisi permintaan penyesuaian sistem
<i>Gantt Chart</i>	diagram batang yang menggambarkan jadwal proyek
<i>Increment</i>	hasil pekerjaan yang dikerjakan oleh development team yang telah diselesaikan dalam sprint
Iterasi	proses yang dilakukan secara berulang-ulang
<i>Scrum</i>	salah satu kerangka kerja yang mengikuti prinsip <i>agile development</i>
<i>ScrumBut</i>	praktik ketidakaturan dalam scrum



DAFTAR ISI

HALAMAN PENGESAHAN DOSEN PEMBIMBING	ii
HALAMAN PENGESAHAN DOSEN PENGUJI	iii
HALAMAN PERNYATAAN KEASLIAN TUGAS AKHIR.....	iv
HALAMAN PERSEMBAHAN	v
HALAMAN MOTO	vi
KATA PENGANTAR.....	vii
SARI	ix
GLOSARIUM	x
DAFTAR ISI	xi
DAFTAR TABEL	xiii
DAFTAR GAMBAR.....	xiv
BAB I PENDAHULUAN	1
1.1 Latar Belakang	1
1.2 Ruang Lingkup Magang.....	2
1.3 Tujuan	4
1.4 Manfaat	4
1.5 Sistematika Penulisan	4
BAB II KAJIAN PUSTAKA	6
2.1 <i>Agile Development</i>	6
2.2 <i>Scrum</i>	6
2.2.1 <i>Tim Scrum</i>	7
2.2.2 <i>Artefak Scrum</i>	8
2.2.3 <i>Acara Scrum</i>	8
2.3 <i>ScrumBut</i>	10
2.4 Penelitian Serupa.....	11
BAB III PELAKSANAAN MAGANG	13
3.1 Manajemen Proyek	13
3.1.1 Pendefinisian Proyek	13
3.1.2 Inisialisasi Proyek.....	13
3.1.3 Perencanaan Proyek	14
3.1.4 Pelaksanaan Proyek.....	15
3.1.5 Pemantauan dan Pengendalian Proyek.....	22
3.1.6 Penutupan Proyek.....	24
3.2 Implementasi <i>Scrum</i> pada Pengembangan Modul LIQUID	24
3.2.1 <i>Sprint planning</i>	25
3.2.2 <i>Sprint backlog</i>	26
3.2.3 <i>Daily scrum</i>	27
3.2.4 <i>Sprint review</i>	27
3.2.5 <i>Sprint retrospective</i>	28
3.2.6 <i>Weekly Report</i>	28
3.3 Praktik Non-Scrum pada Pengembangan modul LIQUID	29
BAB IV REFLEKSI PELAKSANAAN MAGANG	31
4.1 Teknis.....	31
4.1.1 Bentuk Implementasi <i>Scrum</i>	31
4.1.2 Bentuk Implementasi Non- <i>Scrum</i>	33
4.1.3 Proyek Lain	37
4.2 Non Teknis.....	38

4.2.1	Komunikasi	38
4.2.2	Knowledge.....	38
4.2.3	Self-Management	39
4.2.4	Menemukan <i>Passion</i>	39
BAB V KESIMPULAN DAN SARAN		40
5.1	Kesimpulan	40
5.2	Saran.....	41
DAFTAR PUSTAKA.....		42
LAMPIRAN		44



DAFTAR TABEL

Tabel 3.1 Spesifikasi Teknologi	13
Tabel 3.2 Daftar prioritas <i>feedback</i> setelah dilakukan sosialisasi.....	16
Tabel 3.3 <i>Product backlog</i> pengembangan modul LIQUID.....	25
Tabel 4.1 Perbandingan implementasi <i>scrum</i> antara <i>Scrum Guide</i> dan modul LIQUID.....	31
Tabel 4.2 Perbedaan implementasi <i>scrum</i> berdasarkan peran	33
Tabel 4.3 Perbedaan implementasi <i>scrum</i> berdasarkan acara.....	35



DAFTAR GAMBAR

Gambar 1.1 Lokasi PT Javan Cipta Solusi dari Google Maps	3
Gambar 2.1 Alur kerangka kerja <i>scrum</i> (sumber: Fakhri, 2019).....	9
Gambar 3.1 Grafik <i>Gantt Chart</i>	15
Gambar 3.2 Tampilan <i>mockup</i> fitur memajukan jadwal menggunakan Balsamiq	18
Gambar 3.3 Hasil mockup fitur memajukan jadwal yang telah diimplementasi	18
Gambar 3.4 Query view “Rekapitulasi Liquid”	20
Gambar 3.5 Perbandingan <i>query</i> sebelum dan sesudah dilakukan optimasi	20
Gambar 3.6 Contoh dokumen <i>test case scenario</i> modul LIQUID.....	21
Gambar 3.7 Tampilan tabel daftar peserta Liquid	21
Gambar 3.8 Tampilan fitur notifikasi untuk peserta bawahan.....	22
Gambar 3.9 Fitur label <i>task</i> pada <i>ActiveCollab</i> untuk menunjukkan status pengerjaan.....	23
Gambar 3.10 Tampilan daftar <i>issue sentry</i> pada pengembangan modul LIQUID	24
Gambar 3.11 Daftar <i>sprint backlog</i>	26
Gambar 3.12 Representasi 1 <i>task ActiveCollab</i>	27
Gambar 3.13 Contoh task dengan label <i>Ready to Test</i>	27
Gambar 3.14 Contoh <i>burn-up chart</i> pada <i>weekly report</i> modul LIQUID	29

BAB I

PENDAHULUAN

1.1 Latar Belakang

Kepemimpinan merupakan komponen yang sangat penting dalam suatu organisasi karena di dalamnya terdapat seseorang yang memiliki kemampuan untuk mempengaruhi, memberi petunjuk dan mampu menentukan individu untuk mencapai tujuan organisasi. Supaya organisasi dapat mencapai tujuannya, maka suatu organisasi harus digerakkan oleh seorang pemimpin (*leader*). 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 untuk mencapai kinerja yang unggul.

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 (Redaksi, 2017). 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 pada setiap unit. Kegiatan ini dilakukan dengan cara mengumpulkan peserta disuatu tempat kemudian peserta diberikan arahan untuk menuliskan *feedback* dengan *sticky notes* yang akan ditempelkan pada sebuah papan. *Feedback* yang diberikan oleh peserta berupa kelebihan, kelemahan, harapan, dan saran untuk atasannya. Dari hasil *feedback* tersebut disusun 3 komitmen perbaikan atau resolusi. Resolusi ditentukan dari *feedback* kelemahan yang dianggap *urgent* kemudian disusun menjadi kalimat positif. Kegiatan LIQUID akan dilaksanakan rutin setiap 6 bulan sekali.

Seiring berjalannya waktu, kegiatan LIQUID dengan cara manual dianggap tidak efektif karena memakan banyak waktu ketika mengumpulkan peserta dan mengolah data *feedback* yang telah diberikan peserta. Oleh karena itu untuk mengatasi permasalahan tersebut, kegiatan LIQUID yang selama ini dilaksanakan secara manual akan di-*improve* dengan mengembangkan sebuah modul yaitu modul LIQUID. 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 (PLN, n.d.). 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.

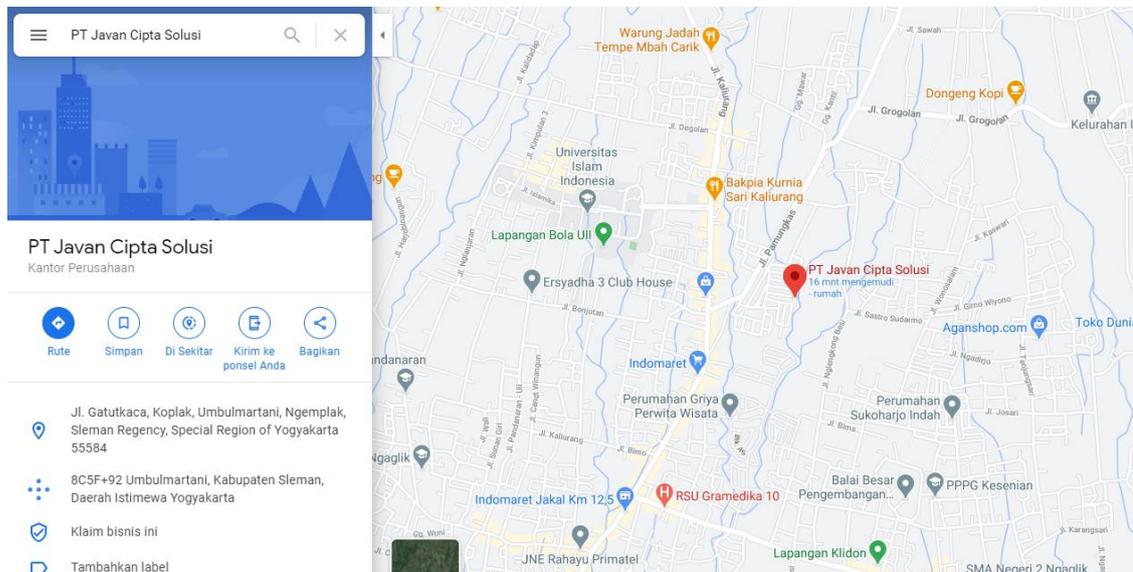
Pengembangan modul LIQUID dilakukan dengan metode pengembangan *agile* dengan kerangka kerja *scrum*. *Scrum* merupakan kerangka kerja yang dapat mengatasi masalah kompleks adaptif atau memungkinkan penyelesaian masalah yang kompleks dan berubah-ubah secara bersamaan (Ken Schwaber dan Jeff Sutherland, 2017). Selain itu, *scrum* bertumpu pada kekuatan kolaborasi tim, inkremental produk, dan proses iterasi untuk mewujudkan hasil akhir yang bernilai tinggi (Ken Schwaber dan Jeff Sutherland, 2017). Berdasarkan penjelasan tersebut, *scrum* merupakan kerangka kerja yang cocok diimplementasikan dalam pengembangan modul LIQUID karena dalam pengembangannya terdapat banyak penambahan dan perubahan fitur yang disampaikan oleh klien secara berkala. Namun, pada proses pengembangan modul LIQUID, praktik *scrum* tidak menerapkan seluruh aturan yang ada di *Scrum Guide* atau disebut dengan *ScrumBut* (Schwaber, n.d.). Hal ini bisa menjadi salah satu kegagalan penerapan *scrum* karena tidak mengikuti standar aturan yang ada (Krishna & Basu, 2012). Oleh karena itu, untuk mengatasi masalah tersebut penerapan *scrum* dibantu dengan *tools* manajemen proyek yang diterapkan pada acara *scrum* agar praktiknya dapat memberikan hasil yang lebih baik.

Laporan ini bertujuan untuk mengetahui bagaimana implementasi kerangka kerja *scrum* pada pengembangan modul LIQUID. Adapun manfaat yang diharapkan adalah penerapan *scrum* dapat membantu proses pemenuhan kebutuhan dan penyelesaian masalah yang kompleks secara efektif dan efisien. Selain itu, *scrum* juga diharapkan membantu dalam hal komunikasi dan pemantauan sisa pekerjaan dengan mudah. Sehingga, apabila ada kendala pada suatu pekerjaan akan terlihat dengan jelas.

1.2 Ruang Lingkup Magang

Pelaksanaan magang di PT Javan Cipta Solusi berlangsung selama tujuh bulan dengan periode Februari 2020 sampai dengan Agustus 2020. PT Javan Cipta Solusi atau Javan

merupakan sebuah *Software House* yang fokus dalam penyediaan solusi untuk optimasi proses bisnis IT di perusahaan swasta, perbankan, dan bahkan lembaga pemerintahan. PT Javan Cipta Solusi memiliki dua cabang kantor yang terletak di Yogyakarta dan Bandung. Kantor cabang Yogyakarta berlokasi di Jl. Gatotkaca, Koplak, Umbulmartani, Ngemplak, Kabupaten Sleman, Daerah Istimewa Yogyakarta. Lokasi lebih jelas dapat dilihat pada Gambar 1.1



Gambar 1.1 Lokasi PT Javan Cipta Solusi dari Google Maps

Selama magang di PT Javan Cipta Solusi, penulis telah berpartisipasi dalam beberapa proyek salah satunya adalah pengembangan modul LIQUID dalam aplikasi Komando. Modul LIQUID mulai dikembangkan sejak bulan Juni 2020 hingga Agustus 2020 ditambah masa retribusi satu bulan. Adapun aktivitas yang dilakukan selama magang secara keseluruhan yaitu:

- a. Mempersiapkan *tools* untuk Magang
- b. Membuat *wireframe website* untuk proyek PPSDM
- c. Membuat laporan dengan *JasperReport* dalam proyek SIMPEL
- d. Memperbaiki *website* Indowig
- e. Melakukan *testing* dalam proyek SMEP
- f. Menganalisis proyek *Dashboard* Bappeda
- g. Melakukan *testing* dalam proyek E-Jawatan & E-HR
- h. Menganalisis proyek E-Jawatan & E-HR
- i. Menganalisis proyek Modul LIQUID
- j. Membuat *query view* pada proyek Modul LIQUID
- k. Mendampingi sosialisasi pada proyek Modul LIQUID

1.3 Tujuan

Tujuan dari laporan tugas akhir ini adalah untuk mengimplementasikan *scrum* pada pengembangan modul LIQUID. Tujuan ini perlu dicapai untuk mengetahui hal-hal apa saja yang berbeda dari penerapan *scrum* pada modul LIQUID dengan *Scrum Guide*. Selain itu, dengan adanya pekerjaan ini, dampak yang dihasilkan dari implementasi *scrum* dapat didapatkan dan dijadikan referensi untuk pengembangan sistem informasi lainnya.

1.4 Manfaat

Manfaat yang diperoleh dari implementasi kerangka kerja *scrum* dalam pengembangan modul LIQUID adalah sebagai berikut:

- a. Memudahkan proses inspeksi dan peninjauan pekerjaan yang dilakukan oleh setiap individu.
- b. Mengetahui perbedaan implementasi *scrum* dalam suatu proyek dengan *scrum* secara teori berdasarkan *Scrum Guide*.
- c. Setiap masalah yang timbul dapat diidentifikasi dan diselesaikan dengan cepat melalui *daily scrum*.
- d. Dengan adanya *sprint*, *scrum* dapat mudah mengatasi setiap perubahan yang terjadi.

1.5 Sistematika Penulisan

Sistematika penulisan laporan ini bertujuan untuk memudahkan pembaca dalam memahami isi laporan secara menyeluruh. Adapun susunan dari laporan ini adalah sebagai berikut.

BAB I PENDAHULUAN

Bab ini membahas tentang latar belakang pengembangan modul LIQUID dan implementasi *scrum* dalam proses pengembangannya, ruang lingkup magang, tujuan implementasi *scrum*, manfaat implementasi *scrum*, dan sistematika penulisan.

BAB II KAJIAN PUSTAKA

Bab ini membahas tentang beberapa teori dan penelitian serupa yang berkaitan dengan implementasi *scrum* pada pengembangan modul LIQUID.

BAB III PELAKSANAAN MAGANG

Bab ini terdiri dari 2 sub bab, yaitu manajemen proyek dan implementasi *scrum* dalam pengembangan modul LIQUID. Manajemen proyek berisi tentang bagaimana modul LIQUID

dikelola agar mencapai tujuan proyek. Sedangkan, implementasi *scrum* berisi tentang bagaimana penerapan *scrum* pada pengembangan modul LIQUID.

BAB IV REFLEKSI PELAKSANAAN MAGANG

Bab ini membahas tentang refleksi dari pelaksanaan magang di PT Javan Cipta Solusi baik dari sisi teknis maupun non-teknis.

BAB V KESIMPULAN DAN SARAN

Bab ini berisi kesimpulan dari implementasi *scrum* dalam pengembangan modul LIQUID, serta saran yang diberikan untuk pihak-pihak yang terkait.



BAB II KAJIAN PUSTAKA

2.1 *Agile Development*

Agile development merupakan salah satu metodologi dalam pengembangan perangkat lunak. Metode pengembangan *agile* dirumuskan pada tahun 2001 melalui *Agile Manifesto* atau nilai dasar *agile* yang merupakan hasil dari pertemuan 17 praktisi pengembangan perangkat lunak dengan tujuan memenuhi kebutuhan industri akan metode pengembangan perangkat lunak yang cepat (Blankenship et al., 2011). *Agile development* memungkinkan organisasi untuk beradaptasi dengan pengembangan yang lebih fleksibel (Cockburn & Highsmith, 2001).

Metode *agile* menerapkan konsep iterasi dalam meningkatkan produknya. Setiap iterasi akan menghasilkan versi produk terbaru yang dibuat setiap dua atau tiga minggu dan didistribusikan kepada pelanggan. *Agile* juga melibatkan pelanggan secara langsung dalam proses pengembangannya agar mereka mendapatkan *feedback* dari para pelanggan. *Agile* menganggap proses desain dan implementasi sebagai inti dari semua aktivitas dalam proses pengembangan perangkat lunak, sehingga proses dokumentasi diminimalisasi dan cenderung memilih jenis komunikasi secara informal daripada pertemuan formal dan dokumentasi tertulis (Sommerville, 2010).

2.2 *Scrum*

Scrum merupakan salah satu dari berbagai kerangka kerja *agile* karena menerapkan dan mengikuti nilai dan prinsip yang ada pada *Agile Manifesto* (Sommerville, 2010). Pada tahun 1986, Hirotaka Takeuchi dan Ikujiro Nonaka memperkenalkan *scrum* pertama kali melalui artikelnya yang berjudul "*The New New Product Development Game*". *Scrum* adalah kerangka kerja yang bersifat iteratif dan inkremental untuk mengendalikan risiko dan mengoptimalkan kemampuan prediksi proyek (Lei et al., 2017). Kekuatan tim memegang peran penting dalam berjalannya *scrum*. Hal ini memungkinkan orang-orang untuk bekerja sama secara efektif, dan dengan demikian, mereka dapat menghasilkan produk yang bernilai tinggi dan canggih (Schwaber & Beedle, 2001).

Scrum terdiri dari beberapa elemen utama yaitu peran, acara, artefak, dan aturan-aturan *scrum*. Setiap komponen elemen memiliki tujuan tertentu dan sangat berpengaruh pada keberhasilan *scrum*, terutama pada elemen peran dan acara *scrum*. Selain itu, aturan *scrum* juga

akan mengikat peran, acara, dan artefak, serta memelihara hubungan dan interaksi antara komponen-komponen tersebut. *Scrum* memiliki tiga pilar yang memperkuat setiap implementasi dari proses kontrol empiris di antaranya adalah transparansi, inspeksi, dan adaptasi. Transparansi berarti proyek dapat terlihat oleh semua orang yang bertanggung jawab atas hasil proyek tersebut. Inspeksi bertujuan agar aplikasi sesuai hasil yang diharapkan atau sesuai dengan *sprint goal*. Biasanya, proses inspeksi dilakukan pada *sprint review* dengan melakukan demo aplikasi. Sedangkan, adaptasi merupakan proses yang dapat disesuaikan apabila ditemukan proses yang menyimpang diluar ambang batas (Schwaber & Sutherland, 2017).

2.2.1 Tim Scrum

Peran *scrum* atau tim *scrum* merupakan salah satu komponen utama dalam penerapan *scrum*. Tim *scrum* terdiri dari tiga peran yaitu *Product Owner*, *Development Team*, dan *Scrum Master*. Tim *scrum* bersifat swakelola (*self-organizing*) dan lintas fungsi (*cross-functional*). Maksud dari swakelola adalah tim dapat memilih cara terbaik dalam menyelesaikan pekerjaan mereka tanpa harus diperintahkan oleh seseorang diluar tim. Sedangkan lintas fungsi adalah tim memiliki kemampuan untuk menyelesaikan pekerjaan mereka tanpa bergantung pada orang lain diluar tim. Tim akan menghantarkan produk secara iteratif dan inkremental untuk memaksimalkan umpan balik yang diterima. Berikut penjelasan tentang peran dari tim *scrum*:

a. *Product Owner*

Product owner bertanggung jawab untuk mengelola *product backlog* dan memaksimalkan nilai produk. *Product owner* akan menjelaskan dan memastikan kepada tim tentang isi dan tujuan dari *product backlog*. Oleh karena itu, *product owner* akan memprioritaskan item *product backlog*, serta memastikan item tersebut transparan dan jelas bagi semua orang (Schwaber & Sutherland, 2017).

b. *Scrum Master*

Scrum Master akan memfasilitasi penggunaan *scrum* sesuai dengan praktik dan tata nilai *scrum*, serta menghilangkan hambatan yang memperlambat perkembangan pekerjaan *development team*. *Scrum master* akan memimpin dan mengatur keberlangsungan kerangka kerja *scrum*. Maka dari itu, *scrum master* memastikan bahwa tujuan, ruang lingkup, dan ranah produk dipahami oleh tim *scrum* (Schwaber & Sutherland, 2017).

c. *Development Team*

Development team bertanggung jawab dalam membuat dan mengirimkan produk yang dapat dirilis pada setiap akhir “*sprint*”, yang merupakan batasan waktu untuk meningkatkan

produk yang dapat digunakan. Selain itu, *development team* juga bertanggung jawab atas *product backlog* yang sudah disampaikan oleh *product owner*, mereka akan menyusun dan mengelola pekerjaan mereka sendiri (Schwaber & Sutherland, 2017).

2.2.2 Artefak Scrum

Artefak *scrum* merupakan representasi dari pekerjaan atau nilai bisnis guna menciptakan transparansi dan kesempatan untuk menginspeksi dan mengadaptasi. Artefak *scrum* terdiri dari *product backlog*, *sprint backlog*, dan *increment*.

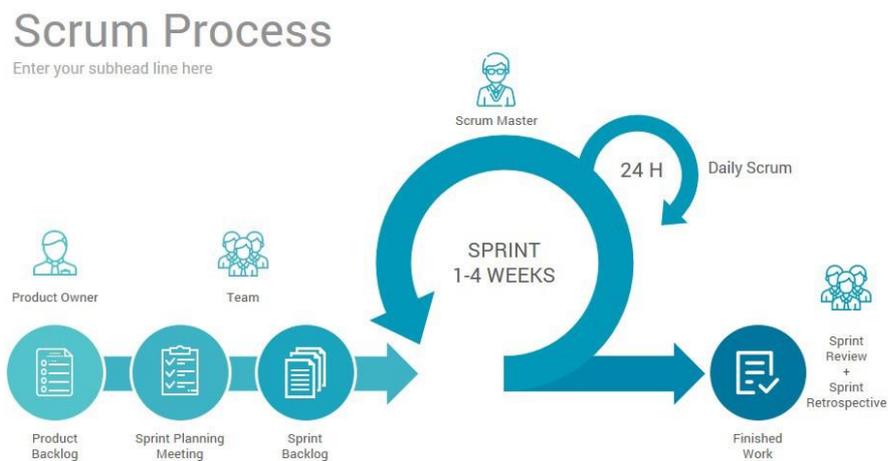
Product backlog berisi daftar kebutuhan user yang akan diurutkan berdasarkan prioritas. *Product owner* bertanggung jawab atas *product backlog*. Prioritas ditentukan berdasarkan tingkat kebutuhan dari fitur yang diminta atau kebutuhan yang akan segera digunakan. *Product backlog* akan selalu ada selama produk masih ada. *Product backlog* akan berevolusi seiring dengan perkembangan produk dan lingkungan di mana produk tersebut dikembangkan.

Sprint backlog berisi seluruh daftar pekerjaan yang telah disepakati pada *sprint planning*. Ketika ada task baru yang diperlukan, maka akan ditambahkan pada *sprint backlog*. *Development team* mengubah *sprint backlog* sepanjang *sprint*, dan *sprint backlog* muncul saat *sprint* berlangsung. Dalam *sprint backlog*, *development team* dapat memantau keadaan dari sisa pekerjaan yang telah direncanakan sebelumnya.

Increment adalah wujud dari *product backlog* item yang diselesaikan dalam *sprint*. Pada akhir *sprint*, *increment* terbaru harus memenuhi kondisi yang dapat digunakan dan sesuai dengan definisi “Selesai” yang telah dibuat oleh tim *scrum* (Schwaber & Sutherland, 2017).

2.2.3 Acara Scrum

Acara *scrum* merupakan komponen utama yang harus ada dalam penerapan *scrum*. Beberapa acara *scrum* diselenggarakan dengan tujuan untuk menciptakan kerutinan dan mengurangi kemunculan acara lain yang tidak termasuk dalam *scrum*. Setiap acara *scrum* memiliki batasan waktu atau durasi waktu maksimum. Ketika tujuan acara telah tercapai, acara-acara selain *sprint* dapat diakhiri kapan saja agar memastikan waktu digunakan secukupnya dan tidak terbuang sia-sia sepanjang proses (Schwaber & Sutherland, 2017). *Scrum* memiliki lima acara yang terdiri dari *sprint*, *sprint planning*, *daily scrum*, *sprint review*, dan *sprint retrospective*.



Gambar 2.1 Alur kerangka kerja *scrum* (sumber: Fakhri, 2019)

a. *Sprint*

Sprint merupakan jantung dari *scrum*, yang merupakan batasan waktu dengan durasi maksimal satu bulan. *Sprint* memiliki durasi yang konsisten, apabila *sprint* sebelumnya telah selesai maka *sprint* baru akan langsung dimulai. Jika durasi *sprint* terlalu panjang, maka definisi yang akan dibangun dapat berubah, kompleksitas meningkat, dan resiko meningkat. *Sprint* berisi dan terdiri atas *sprint planning*, *daily scrum*, *sprint review*, dan *sprint retrospective* yang alurnya dapat dilihat pada Gambar 2.1

b. *Sprint planning*

Tim *scrum* secara kolaboratif akan melakukan analisis pekerjaan yang akan dikerjakan dalam waktu *sprint*. Dalam *sprint planning*, task atau pekerjaan akan dipilih oleh tim *scrum* dari *product backlog* sebelumnya. Pada tahap ini, *scrum* master harus memastikan seluruh anggota tim memahami tujuannya. *Scrum* master juga mendukung tim *scrum* untuk melaksanakannya dalam batasan waktu (Schwaber & Sutherland, 2017).

Terdapat beberapa pertanyaan yang harus terjawab pada saat *sprint planning*, yaitu:

- Apa yang dapat dihantarkan ke dalam *increment* dari *Sprint* ini?
- Bagaimana mereka bisa menyelesaikan pekerjaan yang dibutuhkan untuk menghantarkan *increment*?

c. *Daily scrum*

Daily scrum dapat dilaksanakan apabila *sprint* telah berjalan di mana setiap individu akan mengikuti meeting yang dilakukan setiap hari selama 15 menit. Dalam *daily scrum*, setiap individu akan memaparkan apa yang akan dikerjakan dalam waktu 24 jam ke depan. *Daily scrum* bertujuan untuk mengoptimalkan kolaborasi dan kinerja tim dengan

melakukan inspeksi pada pekerjaan yang telah diselesaikan sejak *daily scrum* sebelumnya. *Scrum* master memastikan bahwa development team melakukan *daily scrum* setiap hari (Schwaber & Sutherland, 2017).

Dalam *daily scrum*, development team akan berdiskusi terkait tiga pertanyaan berikut:

- Apa yang telah saya kerjakan kemarin?
- Apa yang akan saya kerjakan hari ini?
- Apakah ada hambatan yang menghalangi pekerjaan saya?

d. *Sprint review*

Acara ini dilaksanakan pada akhir *sprint* yang bertujuan untuk melakukan inspeksi dan perbaikan *product backlog* jika diperlukan. *Sprint review* dihadiri oleh tim *scrum* dan stakeholder yang akan berkolaborasi untuk meninjau apa yang sudah dikerjakan dalam *sprint* yang telah usai dan menentukan apa yang dapat dikerjakan pada *sprint* berikutnya. Hasil dari *sprint review* adalah perbaikan dari *product backlog* yang menjabarkan kemungkinan item *product backlog* untuk *sprint* berikutnya (Schwaber & Sutherland, 2017).

e. *Sprint retrospective*

Sprint retrospective dilakukan setelah *sprint review* dan sebelum *sprint planning* terlaksana. Acara ini dihadiri oleh seluruh tim *scrum*. Pada tahap ini tim *scrum* mendapat kesempatan untuk meninjau dirinya sendiri dan membuat rencana peningkatan pada *sprint* berikutnya. Adapun tujuan dari *sprint retrospective* sebagai berikut (Schwaber & Sutherland, 2017):

- Menginspeksi bagaimana jalannya *Sprint* terakhir yang terkait dengan orang-orang, hubungan antar mereka, proses, dan alat-alat yang digunakan;
- Mengidentifikasi dan mengurutkan hal utama yang berjalan dengan baik dan peningkatan yang berpotensi untuk dilakukan; dan,
- Membuat perencanaan untuk implementasi peningkatan cara kerja *Scrum* Team

Pada akhir *sprint retrospective*, tim *scrum* harus menyetujui peningkatan yang akan dicapai pada *sprint* berikutnya. Hal ini menjadi bentuk inspeksi dan adaptasi terhadap tim *scrum* itu sendiri.

2.3 *ScrumBut*

Dalam panduan *scrum* sudah ditetapkan bahwa peran, artefak, acara, dan aturan *scrum* tidak dapat diubah dan walaupun hanya menerapkan sebagian dari *scrum*, maka itu tidak bisa

disebut dengan *scrum*. Namun, jika dilihat lebih detail, beberapa perusahaan biasanya tidak menerapkan *scrum* secara keseluruhan. Kondisi ini sering disebut sebagai "*ScrumBut*" (Hassani-Alaoui et al., 2020).

ScrumBut adalah alasan mengapa tim dan organisasi tidak dapat mengambil keuntungan *scrum* sepenuhnya dan tidak menyadari manfaat penggunaan *scrum* dalam proses pengembangan lunak mereka. Hal ini bisa disebabkan oleh berbagai alasan, seperti pemahaman dan keinginan yang rendah terhadap *scrum*, atau regulasi, budaya, dan birokrasi perusahaan yang telah ada tidak sejalan dengan nilai dan prinsip *scrum* yang selanjutnya diselesaikan dengan mencari solusi diluar nilai dan prinsip yang ada (Fuadi, 2020).

2.4 Penelitian Serupa

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 (Adi, 2015). 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 (Sholahudin et al., 2017). 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 (Ependi, 2018).

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* pengguna sistem tidak ikut serta dan ketiadaan *burndown chart* (Tohirin & Widiyanto, 2020). Selain itu, pada penelitian (Akif & Majeed, 2012) tentang masalah dan tantangan dalam implementasi *scrum*, menjelaskan bagaimana penerapan *scrum* dalam dua organisasi. Hasil dari penelitian tersebut penerapan *scrum* tidak sepenuhnya mengikuti panduan *scrum*, namun dilakukan modifikasi dengan memberikan beberapa solusi terhadap kekurangan *scrum* yang diterapkan. Penelitian ini mendukung pengimplementasian *scrum* dengan praktik yang dinamis atau berbeda-beda.

Terkait kegiatan umpan balik, terdapat penelitian yang menjelaskan pengaruh umpan balik dalam suatu organisasi (Wan, 2019). Dalam penelitian ini, keberadaan umpan balik dapat mempengaruhi kinerja karyawan. Dengan adanya umpan balik, kesalahan dapat dikoreksi dan kinerja karyawan dapat ditingkatkan. Selain itu, penelitian tentang pengaruh perilaku pemimpin terhadap kinerja karyawan dengan perhitungan koefisien korelasi menyatakan bahwa hubungan antara variabel perilaku pemimpin dengan kinerja karyawan dikategorikan kuat dan cukup tinggi. Hal ini menunjukkan bahwa adanya hubungan yang nyata antara perilaku pemimpin dan kinerja karyawan (Anzhari et al., 2016).

Dari penelitian yang ada, telah dilakukan implementasi *scrum* pada pengembangan aplikasi dan proyek manajemen. Pada penelitian yang telah diuraikan, terdapat empat penelitian yang menjelaskan manfaat implementasi *scrum*, dan dua 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 mempengaruhi 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.

BAB III

PELAKSANAAN MAGANG

3.1 Manajemen Proyek

Dalam pengembangan modul LIQUID, diterapkan manajemen proyek yang terdiri dari beberapa tahap seperti: pendefinisian proyek, inisialisasi proyek, perencanaan proyek, pemantauan dan pengendalian proyek, dan penutupan proyek. Tahapan tersebut dilakukan sebagai bentuk persiapan untuk mengerjakan sebuah proyek agar terarah dan terorganisir.

3.1.1 Pendefinisian Proyek

Modul LIQUID merupakan sebuah modul yang dikembangkan dalam aplikasi Komando. Modul LIQUID dikembangkan untuk memfasilitasi kegiatan pemberian umpan balik atau *feedback* yang dilakukan oleh bawahan kepada atasannya untuk mengetahui kualitas kepemimpinan. Modul ini akan digunakan oleh seluruh pegawai PT PLN (Persero). Dengan adanya modul ini, kegiatan *feedback* yang selama ini dilakukan secara manual, akan menjadi berbasis aplikasi yang lebih efisien, cepat, dan mudah dikelola.

3.1.2 Inisialisasi Proyek

Pada tahap inisialisasi Modul LIQUID, ditentukan spesifikasi kebutuhan sistem guna mendukung pengembangan modul agar sesuai dengan harapan. Beberapa teknologi yang digunakan dalam pengembangan modul sudah ditentukan oleh pihak perusahaan karena menyesuaikan dengan spesifikasi yang digunakan dalam aplikasi Komando seperti pada Tabel 3.1

Tabel 3.1 Spesifikasi Teknologi

Aspek	Spesifikasi
Bahasa Pemrograman	PHP
Framework	Laravel
Database	Oracle 11g
Cache Database	Redis
Container Platform	Docker
Tools	Project Management : Active Collab Prototype Design : Balsamiq Source Code Repository : GIT Automated Testing : Selenium Analytical and Monitoring Tools : Sentry

Selain itu, proses pengembangan modul LIQUID juga terbagi dalam sejumlah peran di antaranya:

- a. *Project Manager* (PM), sebagai pemimpin proyek yang bertanggung jawab atas pengelolaan produk secara keseluruhan. PM akan mengelola keseluruhan sumber daya proyek untuk mencapai *goal* proyek.
- b. *Business Analyst*, bertanggung jawab untuk menganalisa dan memvalidasi persyaratan untuk perubahan proses bisnis. *Business Analyst* berperan dalam menggerakkan organisasi kearah efisiensi, produktivitas, dan profitabilitas.
- c. *System analyst*, bertanggung jawab pada keberhasilan suatu proyek dengan mengumpulkan kebutuhan user dan mencari solusinya. *System analyst* akan melakukan analisis terhadap kebutuhan user untuk menjadi task yang dapat dikerjakan oleh tim.
- d. *Programmer*, terdiri dari dua level yaitu junior dan senior, keduanya bertanggung jawab atas eksekusi dan implementasi perancangan kebutuhan dalam bentuk kode program.
- e. *Quality Assurance* (QA), bertanggung jawab untuk melakukan pengujian terhadap produk yang dikembangkan.
- f. *Technical Document*, bertanggung jawab pada dokumen-dokumen yang akan dikirim pada klien.

3.1.3 Perencanaan Proyek

Pada tahap perencanaan proyek meliputi tiga hal yaitu perencanaan biaya, ruang lingkup, dan waktu. Dalam laporan ini, perencanaan biaya tidak akan dibahas karena tidak ada kewenangan yang dimiliki. Maka dari itu, dalam laporan ini hanya akan membahas ruang lingkup dan waktu.

Dalam pengembangan modul LIQUID, ruang lingkup pekerjaan atau *Scope of Work* (SOW) diuraikan dengan menggunakan grafik *gant chart* yang dapat dilihat pada Gambar 3.1. *Gantt chart* memaparkan ruang lingkup pekerjaan serta waktu pengerjaannya. Dengan adanya *gant chart* proses *monitoring* ruang lingkup pekerjaan menjadi lebih mudah karena *gant chart* akan selalu di-*update* apabila ada *progress* atau pekerjaan yang telah selesai. Proses pengerjaan modul LIQUID dijadwalkan selama 4 bulan (sudah termasuk masa retensi).

yang bisa dilihat pada Tabel 3.2. Untuk *feedback* yang berupa *bug* akan ditindaklanjuti oleh *programmer* yang memiliki kewenangan memperbaiki *bug*. Prioritas dibagi menjadi tiga 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.

Tabel 3.2 Daftar prioritas *feedback* setelah dilakukan sosialisasi

No	Feedback	Deskripsi	Prioritas
1	[Dashboard] Add Bawahan langsung dari daftar peserta Liquid	Tambahkan fitur untuk menambahkan bawahan langsung dari daftar peserta liquid.	<i>Must to have</i>
2	[Dashboard] Fitur Send Notification untuk Peserta Liquid + Permission-nya	Tambahkan satu kolom pada pop up <i>feedback</i> bawahan, pengukuran pertama, dan pengukuran kedua yang berisi button untuk mengirim notifikasi kepada peserta bawahan.	<i>Good to have</i>
3	[Dashboard] Status Terakhir masih belum sesuai	Status pada daftar peserta liquid masih belum sesuai dengan status terakhirnya.	<i>BUG</i>
4	[Dashboard] Grup Peserta Liquid berdasarkan Jenjang Jabatan	Urutkan peserta liquid berdasarkan jenjang jabatan seperti pada saat <i>create</i> liquid	<i>Must to have</i>
5	[Dashboard/Detail History Penilaian] Urutkan Kelebihan dan Kekurangan DESC dan tambahkan kolom Nomor	Urutkan kelebihan dan kekurangan mulai dari angka terbesar-terkecil. Tambahkan satu kolom untuk penomoran.	<i>Nice to have</i>

b. Membuat fitur untuk memajukan jadwal pengukuran kedua

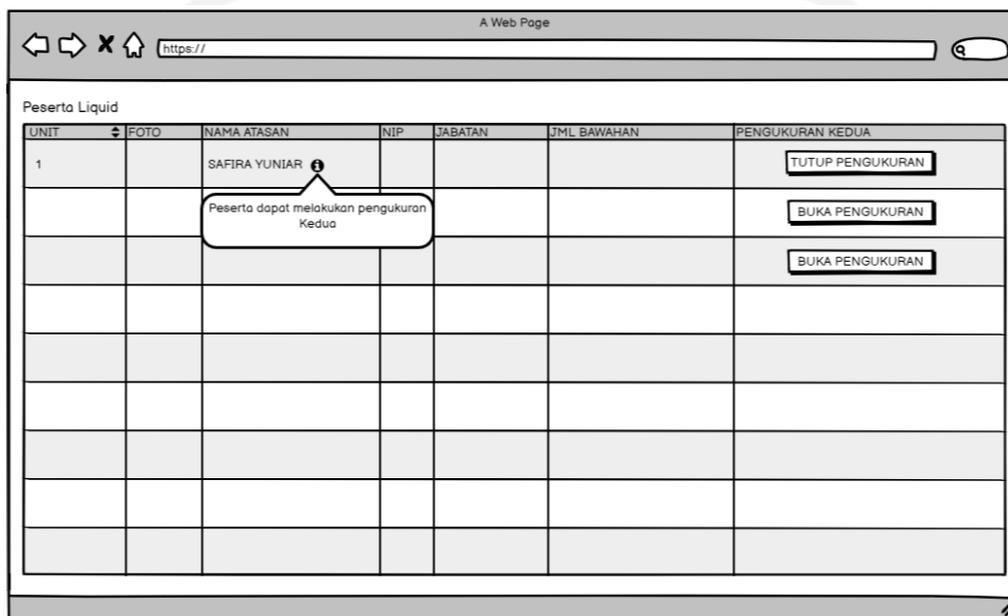
Pelaksanaan kegiatan LIQUID akan dilakukan oleh pegawai atasan dan bawahan. Kegiatan tersebut akan dilaksanakan dengan melalui beberapa tahapan di antaranya sebagai berikut:

- Pemberian *Feedback*, tahap ini akan dilakukan oleh bawahan ketika jadwal LIQUID sudah dibuat oleh admin pelaksana. Pada tahap ini bawahan akan memberikan *feedback* terhadap atasan berupa daftar kelebihan dan kekurangan. Daftar kelebihan dan kekurangan yang disediakan mengacu pada Pedoman Perilaku dan Etika Bisnis PT PLN (Persero). Pemberian *feedback* dilaksanakan sesuai jadwal LIQUID yang sudah ditentukan.
- Penyelarasan, tahap ini hanya dapat dilakukan oleh atasan. Pada tahap ini atasan dapat melihat resolusi atau perubahan yang akan dicapai. Resolusi akan ditentukan oleh sistem berdasarkan data kekurangan yang telah diberikan oleh bawahan. Selanjutnya, atasan dapat memberikan aksi nyata atau aksi yang akan dilakukan untuk mencapai resolusi yang telah ditentukan.
- Pengukuran Pertama, tahap ini dilakukan oleh bawahan setelah atasan melakukan penyelarasan. Pada tahap ini bawahan akan memberikan penilaian berdasarkan resolusi atasan. Penilaian dilakukan dengan memberi skor dari 1-10 pada setiap resolusi.
- Pengisian *Activity Logbook*, tahap ini dilaksanakan oleh atasan setelah tahap pengukuran pertama dan sebelum tahap pengukuran kedua dimulai. *Activity Log* adalah daftar aktivitas yang diisi oleh atasan guna meningkatkan kapasitas diri sesuai dengan *feedback* yang didapatkan.
- Pengukuran Kedua, tahap ini dilakukan oleh bawahan ketika sudah memasuki jadwal pengukuran kedua. Penilaian ini sama seperti tahap pengukuran pertama, namun bawahan dapat melihat bukti atasan telah melakukan resolusi yang dapat dilihat dari *activity logbook*.

Setiap tahapan kegiatan sudah ditentukan jadwalnya, sehingga kegiatan tersebut tidak bisa dilaksanakan diluar jadwal LIQUID. Pada selesainya tahap pengukuran pertama ke pengukuran kedua membutuhkan waktu sekitar 3-6 bulan. Karena waktu yang lama seringkali terjadi mutasi karyawan atasan sebelum memasuki jadwal pengukuan kedua.

Oleh karena itu, dibuatlah fitur untuk memajukan jadwal pengukuran kedua, sehingga pengukuran kedua dapat dilaksanakan lebih awal.

Untuk merancang fitur memajukan jadwal, perlu dibuat visualisasi konsep desain untuk mempermudah *programmer* dalam mengembangkan fitur tersebut. Konsep desain dibuat menggunakan *Balsamiq* yang merupakan sebuah tool untuk merancang *wireframe*. Hasil dari visualisasi yang telah dibuat dapat dilihat pada Gambar 3.2, sedangkan hasil visualisasi yang telah diimplementasikan dalam aplikasi dapat dilihat pada Gambar 3.3.



Gambar 3.2 Tampilan *mockup* fitur memajukan jadwal menggunakan Balsamiq

Show 10 entries Search:

Nama Atasan	NIP	Jabatan	Jumlah Bawahan	Feedback Bawahan	Penyelarasan	Pengukuran Pertama	Activity Log	Pengukuran Kedua	Aktifkan Pengukuran Kedua
		VP DAL LAYANAN TI	5	2/5	Not Yet	0/5	0	0/5	Buka Pengukuran
		VP OPRS SIS DAN TI SUM	6	0/6	Not Yet	0/6	0	0/6	Tutup Pengukuran
		VP REN DAN STRATEGI TI	4	4/4	Done	2/4	2	1/4	Buka Pengukuran
		VP OPRS SIS DAN TI KAL SUL, MALUKU, PAP	7	1/7	Not Yet	0/7	0	0/7	Buka Pengukuran
	7	VP LOLA INFRASTRUKTUR TI	4	1/4	Not Yet	0/4	0	0/4	Buka Pengukuran

Gambar 3.3 Hasil *mockup* fitur memajukan jadwal yang telah diimplementasi

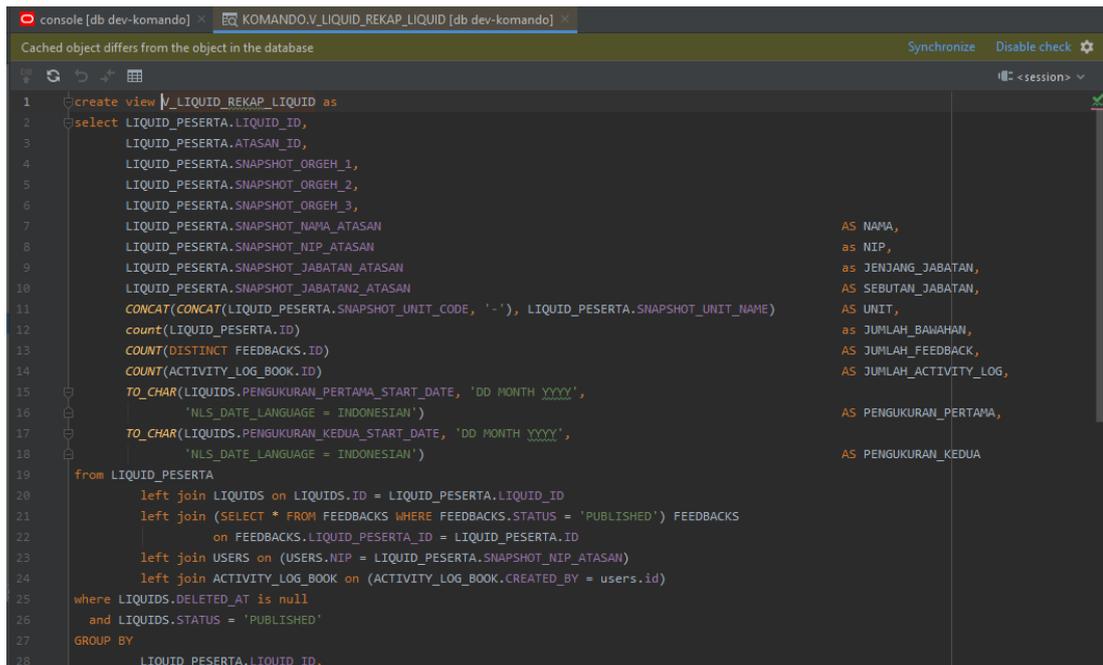
c. Optimasi *query* pada fitur *report* menggunakan *view*

Berdasarkan ruang lingkup pekerjaan modul LIQUID, dikembangkan fitur *report* yang menampilkan hasil rekapitulasi dari kegiatan LIQUID. Fitur *report* dibuat untuk keperluan

manajemen perusahaan PLN terhadap pelaksanaan kegiatan LIQUID. Fitur *report* terbagi menjadi 5 rekapitulasi yang jenisnya berbeda-beda, di antaranya:

- Rekapitulasi LIQUID, fitur *report* ini menampilkan rekapitulasi kegiatan liquid berupa penilaian yang diberikan oleh bawahan kepada atasan.
- Rekapitulasi Top 5 Kelebihan, fitur *report* ini menampilkan jumlah dan persentase *feedback* kelebihan 5 terbanyak dalam bentuk tabel dan grafik *bar chart*.
- Rekapitulasi Top 5 Kekurangan, fitur *report* ini akan menampilkan jumlah dan persentase *feedback* kekurangan 5 terbanyak dalam bentuk tabel dan grafik *bar chart*.
- Rekapitulasi Feedback Lainnya, dalam *report* ini menampilkan *feedback* kelebihan, kekurangan, harapan dan saran yang diberikan oleh bawahan. Namun, pada kelebihan dan kekurangan hanya menampilkan *feedback* yang ditulis langsung oleh bawahan atau diluar Pedoman Perilaku dan Etika Bisnis PT PLN (Persero).
- Rekapitulasi Partisipan, dalam *report* ini menampilkan jumlah dan persentase atasan yang sudah diberi *feedback* oleh bawahan per unit daerah.

System analyst membuat *query* pada setiap *report* yang akan dikembangkan. Hal ini dilakukan untuk mempermudah *programmer* dalam mengintegrasikan *query* ke aplikasi. Namun, pada proses pengaksesan fitur *report* rekapitulasi liquid dan dan rekapitulasi partisipan membutuhkan waktu yang cukup lama. Hal ini disebabkan *query* yang melibatkan banyak *join*, sehingga proses pengolahan data yang berskala besar akan membutuhkan waktu yang cukup lama. Maka dari itu, dilakukan optimasi *query* dengan membuat *view*, sehingga proses pengaksesan dan pengolahan data menjadi lebih cepat. Salah satu contoh *query* pembuatan *view* untuk menampilkan data “Rekapitulasi Liquid” dapat dilihat pada Gambar 3.4.



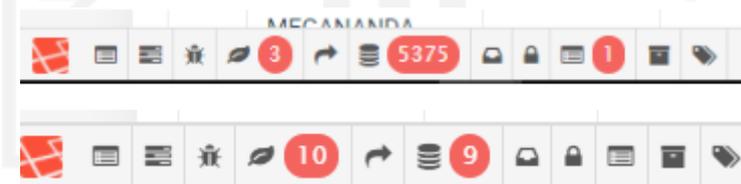
```

1 create view v_LIQUID_REKAP_LIQUID as
2 select LIQUID_PESERTA.LIQUID_ID,
3        LIQUID_PESERTA.ATASAN_ID,
4        LIQUID_PESERTA.SNAPSHOT_ORGEH_1,
5        LIQUID_PESERTA.SNAPSHOT_ORGEH_2,
6        LIQUID_PESERTA.SNAPSHOT_ORGEH_3,
7        LIQUID_PESERTA.SNAPSHOT_NAMA_ATASAN AS NAMA,
8        LIQUID_PESERTA.SNAPSHOT_NIP_ATASAN as NIP,
9        LIQUID_PESERTA.SNAPSHOT_JABATAN_ATASAN as JENJANG_JABATAN,
10       LIQUID_PESERTA.SNAPSHOT_JABATAN2_ATASAN AS SEBUTAN_JABATAN,
11       CONCAT(CONCAT(LIQUID_PESERTA.SNAPSHOT_UNIT_CODE, '-'), LIQUID_PESERTA.SNAPSHOT_UNIT_NAME) AS UNIT,
12       count(LIQUID_PESERTA.ID) as JUMLAH_BAWAHAN,
13       COUNT(DISTINCT FEEDBACKS.ID) AS JUMLAH_FEEDBACK,
14       COUNT(ACTIVITY_LOG_BOOK.ID) AS JUMLAH_ACTIVITY_LOG,
15       TO_CHAR(LIQUIDS.PENGUKURAN_PERTAMA_START_DATE, 'DD MONTH YYYY',
16              'NLS_DATE_LANGUAGE = INDOONESIAN') AS PENGUKURAN_PERTAMA,
17       TO_CHAR(LIQUIDS.PENGUKURAN_KEDUA_START_DATE, 'DD MONTH YYYY',
18              'NLS_DATE_LANGUAGE = INDOONESIAN') AS PENGUKURAN_KEDUA
19 from LIQUID_PESERTA
20      left join LIQUIDS on LIQUIDS.ID = LIQUID_PESERTA.LIQUID_ID
21      left join (SELECT * FROM FEEDBACKS WHERE FEEDBACKS.STATUS = 'PUBLISHED') FEEDBACKS
22              on FEEDBACKS.LIQUID_PESERTA_ID = LIQUID_PESERTA.ID
23      left join USERS on (USERS.NIP = LIQUID_PESERTA.SNAPSHOT_NIP_ATASAN)
24      left join ACTIVITY_LOG_BOOK on (ACTIVITY_LOG_BOOK.CREATED_BY = users.id)
25 where LIQUIDS.DELETED_AT is null
26      and LIQUIDS.STATUS = 'PUBLISHED'
27 group by
28      LIQUID_PESERTA.LIQUID_ID,

```

Gambar 3.4 Query view “Rekapitulasi Liquid”

Hasil eksekusi dari *view* yang telah dibuat menunjukkan bahwa proses pengolahan data lebih cepat daripada cara sebelumnya. Hal ini bisa dilihat pada Gambar 3.5 yang menunjukkan perbandingan jumlah *query* “Rekapitulasi Liquid” sebelum dioptimasi dan sesudah dioptimasi. Sebelum dilakukan optimasi terdapat 5375 *query* yang tereksekusi, sedangkan setelah dioptimasi menampilkan 9 *query* yang tereksekusi. Dari hasil tersebut membuktikan bahwa optimasi *query* telah berhasil dilakukan.



Gambar 3.5 Perbandingan *query* sebelum dan sesudah dilakukan optimasi

d. Melakukan *User Acceptance Test* (UAT) modul LIQUID

User Acceptance Test (UAT) merupakan suatu proses pengujian yang melibatkan stakeholder aplikasi untuk menguji dan memverifikasi bahwa perangkat lunak yang dibangun sudah berjalan sesuai dengan harapan. Pelaksanaan UAT pada pengembangan modul LIQUID dilaksanakan dua kali. UAT pertama dilaksanakan sebelum diadakan sosialisasi modul LIQUID, sedangkan UAT kedua dilaksanakan setelah *feedback* yang disampaikan dari sosialisasi modul LIQUID telah diselesaikan. Sebelum dilakukan UAT, *system analyst* akan membuat dokumen *test case scenario* terlebih dahulu yang bisa dilihat pada Gambar 3.6. Pada pelaksanaan UAT, klien akan mengisi status pada dokumen *test*

case scenario yang sudah disediakan. Apabila terdapat *bug*, maka klien akan menuliskan catatan dalam dokumen tersebut dan akan segera ditindaklanjuti oleh *programmer*.

Test Case Scenario : Create Liquid												
Menu	Sub Menu	ID Testing	Prosedu Pengujian	Objektif	Skenario	Prekondisi	Hasil Yang Diharapkan	Hasil Aktual	Status	Tanggal Testing [dd/mm/yy yy]	Catatan	Automated Test
Tanggal liquid	Create Liquid - Tanggal Liquid	B1	Positive Test	Isi semua data dengan benar	1. Login 2. Buka menu liquid->create liquid 3. Isi dengan benar (tanggal mulai sebelum tanggal selesai), field: - Jadwal feedback - Jadwal penyeselaran - Jadwal pengukuran pertama - Jadwal pengukuran kedua 4. Klik Remender Aksi Resolusi 5. Klik next	Loading	Redirect ke halaman unit kerja	Redirect ke halaman unit kerja	OK			b1_can_create_liquid_with_normal_input
		B2	Negative Test	Tanggal dibuat terbalik	1. Login 2. Buka menu liquid->create liquid 3. Isi tanggal mulai sesudah tanggal selesai), field: - Jadwal feedback - Jadwal penyeselaran - Jadwal pengukuran pertama - Jadwal pengukuran kedua 4. Klik Remender Aksi Resolusi 5. Klik next		Tanggal mulai dan selesai akan otomatis sama apabila terbalik	Tanggal mulai dan selesai akan otomatis sama apabila terbalik	OK			b2_cannot_create_liquid_with_invaid_date
		B3	Negative Test	Input tanggal mulai penyeselaran sama dengan tanggal mulai feedback	1. Login 2. Buka menu liquid->create liquid 3. Isi tanggal mulai sesudah tanggal selesai), field: - Jadwal feedback - Jadwal penyeselaran (input tanggal mulai = tanggal mulai feedback)		Muncul notifikasi "Tanggal Mulai Penyeselaran harus lebih besar atau sama dengan Tanggal Berakhir Feedback"	Muncul notifikasi "Tanggal Mulai Penyeselaran harus lebih besar atau sama dengan Tanggal Berakhir Feedback"	OK			

Gambar 3.6 Contoh dokumen *test case scenario* modul LIQUID

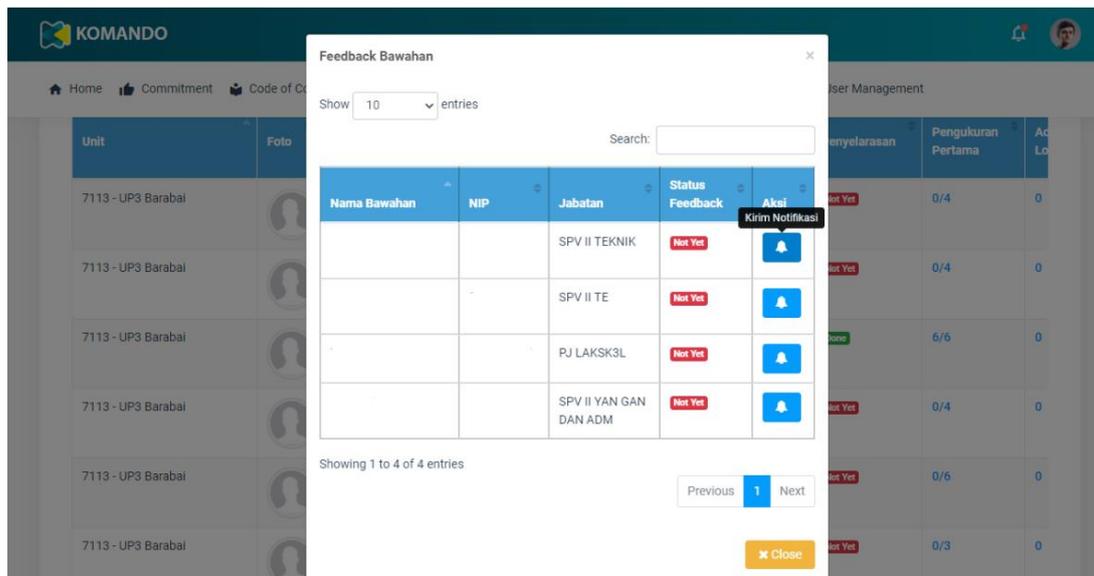
e. Membuat fitur notifikasi untuk peserta bawahan

Pada modul LIQUID khususnya admin liquid dapat melakukan *monitoring* aktivitas peserta liquid melalui tabel peserta liquid pada *dashboard* admin. Tabel peserta liquid menampilkan daftar atasan berdasarkan unit daerah admin liquid tersebut. Pada daftar tersebut, admin liquid dapat melihat bawahan yang sudah melakukan kegiatan liquid seperti pada Gambar 3.7. Angka pada tabel menampilkan nama bawahan yang telah melakukan kegiatan liquid dalam bentuk *pop-up*. Namun, seringkali terjadi kendala yaitu bawahan lupa dengan jadwal liquid, misalnya seperti bawahan belum mengisi pengukuran pertama padahal jadwalnya akan segera usai.

atasan	NIP	Jabatan	Jumlah Bawahan	Feedback Bawahan	Penyeselaran	Pengukuran Pertama	Activity Log	Pengukuran Kedua	Aktifkan Pengukuran Kedua
		VP DAL LAYANAN TI	5	2/5	Not Yet	0/5	0	0/5	Buka Pengukuran
		VP OPRS SIS DAN TI SUM	6	0/6	Not Yet	0/6	0	0/6	Tutup Pengukuran
		VP REN DAN STRATEGI TI	4	4/4	Done	2/4	2	1/4	Buka Pengukuran
		VP OPRS SIS DAN TI KAL, SUL, MALUKU, PAP	7	1/7	Not Yet	0/7	0	0/7	Buka Pengukuran
		VP LOLA INFRASTRUKTUR TI	4	1/4	Not Yet	0/4	0	0/4	Buka Pengukuran

Gambar 3.7 Tampilan tabel daftar peserta Liquid

Untuk mengatasi permasalahan tersebut, dibuat fitur notifikasi pada setiap *pop-up* yang menampilkan nama bawahan seperti yang dipaparkan pada Gambar 3.8. Notifikasi yang dikirimkan berupa jadwal pelaksanaan Liquid melalui pemberitahuan pada aplikasi. Notifikasi akan diterima oleh bawahan yang telah dikirimkan notifikasi, sehingga meminimalisir bawahan lupa terhadap jadwal liquidnya.



Gambar 3.8 Tampilan fitur notifikasi untuk peserta bawahan

3.1.5 Pemantauan dan Pengendalian Proyek

Tahap pemantauan dan pengendalian proyek dilakukan untuk memantau proyek yang sedang berjalan, sehingga apabila ada kendala dapat langsung ditindaklanjuti. Pemantauan dan pengendalian proyek menggunakan beberapa tools, di antaranya:

a. Telegram

Telegram merupakan aplikasi yang digunakan untuk bertukar pesan atau *chatting*. PT Javan Cipta Solusi telah memanfaatkan telegram untuk melakukan komunikasi proyek, tim analis, tim *programmer*, individu, dan bahkan general.

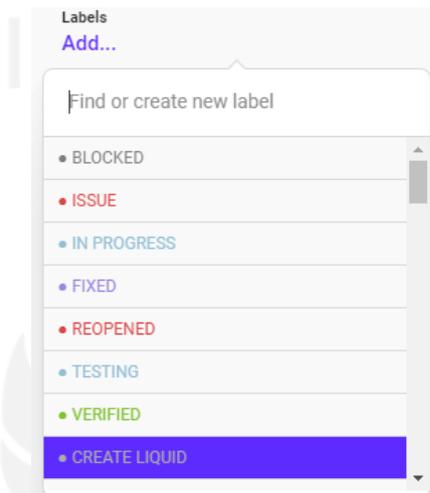
b. WhatsApp

WhatsApp merupakan aplikasi untuk bertukar pesan atau *chatting* tanpa pulsa, melainkan data internet. WhatsApp merupakan aplikasi yang populer dan hampir semua kalangan menggunakan aplikasi ini. Penggunaan WhatsApp juga digunakan pada PT Javan Cipta Solusi yaitu untuk melakukan komunikasi dengan *klien*.

c. Active Collab

Active Collab merupakan *software* untuk manajemen proyek yang diakses secara *online*. Active Collab sangat membantu untuk mengelola pekerjaan dalam pengembangan

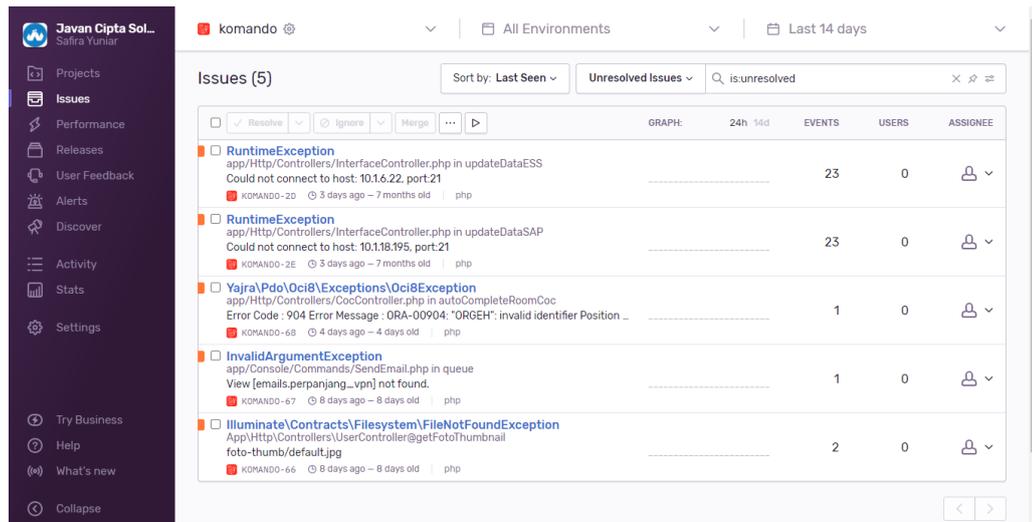
suatu aplikasi. Dalam aplikasi ini menampilkan daftar *task* atau pekerjaan yang harus dikerjakan oleh *programmer*. Selain itu, *ActiveCollab* menyediakan fitur label *task* yang digunakan untuk *tracking* status pengerjaan pada *task* yang dimiliki oleh setiap *programmer* yang bisa dilihat pada Gambar 3.9. Label *task* yang digunakan di antaranya *backlog*, *to-do*, *in progress*, *ready to test*, dan *tested*. Setiap *task* yang sedang dieksekusi akan diberi label *in progress*. Ketika *task* diberi label *ready to test*, maka *task* tersebut sudah bisa diuji oleh QA dan *task* akan diberi label *tested*.



Gambar 3.9 Fitur label *task* pada *ActiveCollab* untuk menunjukkan status pengerjaan

d. Sentry

Sentry merupakan aplikasi yang membantu tim pengembang untuk *monitoring* dan *tracking* error pada suatu aplikasi berbasis web. *Sentry* akan otomatis mendeteksi *error* yang muncul dalam aplikasi. Apabila terdeteksi *error*, maka *sentry* akan mengirimkan pemberitahuan melalui email tim pengembang yang terdaftar dan menampilkan *error* tersebut pada daftar *issue sentry* yang dapat dilihat pada Gambar 3.10. *System analyst* juga akan membuat *task* jika terdapat *error* pada *sentry*, supaya *error* tersebut bisa langsung ditindaklanjuti oleh *programmer*.



Gambar 3.10 Tampilan daftar *issue sentry* pada pengembangan modul LIQUID

e. Slack

Slack merupakan aplikasi yang mendukung komunikasi dan koordinasi antar tim. Dalam pengembangan modul LIQUID, *slack* digunakan untuk melakukan komunikasi antara tim dengan klien. Selain komunikasi *slack* juga dapat digunakan untuk *monitoring* aktivitas *git production*.

3.1.6 Penutupan Proyek

Pada tahap penutupan proyek, ada beberapa tahapan yang harus dilakukan. Tahapan tersebut di antaranya mengadakan UAT, penyerahan dokumen *Software Design Document* (SDD), dokumen *Timesheet*, dan dokumen Berita Acara Serah Terima (BAST). Proyek dapat dikatakan selesai apabila dokumen Berita Acara telah ditandatangani kedua belah pihak antara penyedia jasa dan pengguna jasa. Untuk penandatanganan berita acara akan dilakukan apabila aplikasi sudah berjalan dengan baik di lingkungan *production*. Selama proses penandatanganan berita acara, penyedia jasa akan melakukan perbaikan aplikasi apabila ditemukan *bug* atau *error*.

3.2 Implementasi *Scrum* pada Pengembangan Modul LIQUID

Secara keseluruhan, pengembangan modul LIQUID dikembangkan dengan metode *agile development* menggunakan kerangka kerja *scrum*. Proyek yang dikembangkan dengan kerangka kerja *scrum* umumnya bersifat kompleks adaptif (Ken Schwaber dan Jeff Sutherland, 2017). Secara garis besar, *scrum* memiliki beberapa tahapan dalam proses pengembangannya, seperti pada Gambar 2.1. Dalam penerapannya, *scrum* tim terdiri dari tiga bagian di antaranya:

- *Product Owner*, terdiri dari *Project Manager* (PM) dan *Business Analyst*.

- *Development Team*, terdiri dari *Programmer*, *QA*, *Designer*, dan *Technical Document*.
- *Scrum Master*, terdiri dari *System analyst*.

3.2.1 *Sprint planning*

Tahap *sprint planning* dapat digambarkan sebagai tahap analisis pada masalah yang akan diselesaikan. Pada pengembangan modul LIQUID analisis yang dilakukan meliputi: menguraikan ruang lingkup, menentukan *product backlog*, menjabarkan pekerjaan untuk setiap anggota, dan menentukan *sprint goal*. Dalam menentukan *product backlog* yang akan dikerjakan dalam satu *sprint* dibantu dengan *grafik gantt chart*. Gantt chart adalah sebuah grafik batang yang memberikan visualisasi dalam perencanaan dan penjadwalan proyek. Di dalamnya memaparkan ruang lingkup pekerjaan serta jadwal dan waktu pelaksanaannya, seperti waktu dimulainya pekerjaan dan batas waktu yang digunakan untuk menyelesaikan suatu pekerjaan. Selain itu, terdapat persentase yang menunjukkan *progress* dari setiap pekerjaan yang telah dikerjakan dan *system analyst* berhak untuk memperbarui status persentase seperti pada Gambar 3.1.

Meskipun metode yang dipakai adalah *scrum*, namun penggunaan *gantt chart* tetap masih digunakan. Dengan adanya *gantt chart*, *deadline* setiap ruang lingkup pekerjaan dapat terlihat dengan jelas sehingga proses *monitoring* kemajuan proyek dapat dilakukan dengan mudah. Selain itu, adanya estimasi waktu dalam *gantt chart* memudahkan untuk mendeteksi masalah secepatnya, sehingga apabila ruang lingkup yang dikerjakan terlambat dari jadwal *deadline*, maka *system analyst* dapat memperkirakan strategi apa yang harus diubah.

System analyst akan mengubah persentase *gantt chart* pada setiap ruang lingkup yang telah diselesaikan. Selain penerapan *gantt chart*, pembagian *task* yang dipilih dari *product backlog* untuk *sprint* ditentukan oleh *system analyst*. Terdapat beberapa fitur yang telah dikerjakan selama magang berlangsung seperti yang terlihat pada Tabel 3.3.

Tabel 3.3 *Product backlog* pengembangan modul LIQUID

No	Fitur	Deskripsi
1	Fitur memajukan jadwal pengukuran kedua	Menambahkan fitur untuk mengaktifkan pengukuran kedua pada setiap atasan.
2	Fitur <i>report</i> LIQUID	Membuat fitur <i>report</i> yaitu rekapitulasi <i>top 5</i> kelebihan, rekapitulasi <i>top 5</i> kekurangan, rekapitulasi <i>feedback</i> lainnya, dan rekapitulasi partisipan untuk keperluan manajemen perusahaan.

3	Optimasi <i>query</i> fitur <i>report</i>	Melakukan optimasi dengan membuat <i>view</i> agar pengolahan data pada halaman <i>report</i> tidak memakan banyak waktu.
4	Filter <i>company code</i>	Menambah filter <i>company code</i> pada fitur <i>report</i> yang akan menampilkan unit tertentu berdasarkan <i>business area</i> yang telah dipilih.
5	Fitur notifikasi untuk peserta bawahan	Menambah fitur notifikasi untuk peserta bawahan dengan tujuan untuk memberikan notifikasi kepada peserta yang belum mengisi <i>feedback</i> pada waktu yang sudah ditentukan.
6	Fitur grup peserta liquid berdasarkan jenjang jabatan	Menambah grup berdasarkan jenjang jabatan pada tabel peserta liquid dalam halaman <i>dashboard</i> admin.

3.2.2 *Sprint backlog*

Sprint backlog berisi daftar pekerjaan atau *task* hasil kesepakatan dari *sprint planning* yang akan dikerjakan selama satu *sprint*. Dalam pengembangan modul LIQUID, pembuatan *sprint backlog* dibantu dengan aplikasi proyek manajemen yaitu *ActiveCollab*. *Task* pada *product backlog* yang telah disepakati pada *sprint planning* akan dipindahkan dalam daftar *sprint backlog*. Daftar *sprint backlog* berupa ruang lingkup pekerjaan yang ada pada *gant chart* seperti daftar pada Gambar 3.11. Selain itu, *system analyst* juga akan memberikan ulasan rinci pada setiap *task* yang ada dalam daftar *sprint backlog* berupa deskripsi, ekspektasi, skenario, *url* halaman, dan akun login yang dipakai. Gambar 3.12 merupakan representasi 1 *task* yang ada dalam daftar *sprint backlog*.

Manajemen Banner and Medi... (19)
Dashboard (33)
Pengukuran Kedua (5)
Modul Penyelarasan (24)
Non-Teknis (45)
Feedback Bawahan (24)
Create Jadwal Liquid (50)
Master Data Kelebihan dan... (17)
Termin (2)
Modul Pengukuran Pertama (10)

Gambar 3.11 Daftar *sprint backlog*

 **[Check] Fitur Memajukan jadwal Pengukuran ke 2 untuk Atasan yang akan mutasi**

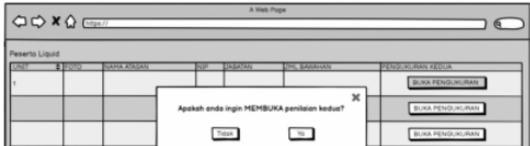
PLN - Komando [Modul Liquid] | Task #822 | Created by Safira Yuniar on Sep 11. 2020

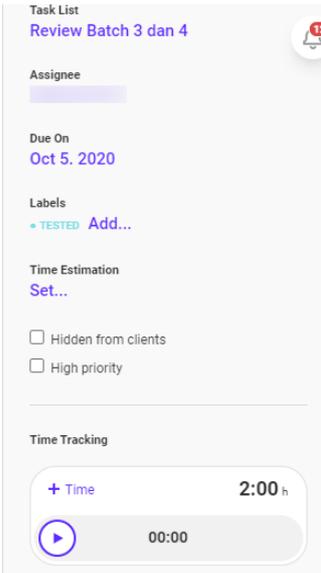
URL :

AKUN :

DESKRIPSI :
 Buat fitur untuk mengaktifkan jadwal pengukuran kedua. Fitur ini digunakan untuk mengaktifkan pengukuran kedua pada atasan yang akan mutasi.

1. Pertama buatlah 1 kolom pada tabel liquid_peserta yaitu "force_pengukuran_kedua"
2. Buat permission untuk fitur ini-> liquid_mutasi
3. Tampilannya berupa button "Buka Pengukuran" seperti mockup dibawah ini dan ketika di click akan muncul pop up. Ketika user memilih "ya", maka input pengukuran kedua akan aktif.





Gambar 3.12 Representasi 1 *task ActiveCollab*

3.2.3 Daily scrum

Selama *sprint* berlangsung, akan diadakan *daily scrum* yang umumnya berdurasi 15 menit untuk setiap pertemuan. 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 daftar *sprint backlog* seperti pada Gambar 3.12, *project manager* akan melakukan proses monitor dan kontrol melalui *ActiveCollab*. Pada *ActiveCollab* setiap *task* yang sedang dieksekusi oleh *programmer* akan diberi label *In Progress*. Kemudian apabila *task* diberi label *Ready to Test*, maka *task* tersebut sudah bisa diuji oleh *QA* dan *task* akan diberi label *Tested* seperti pada Gambar 3.13.

UAT 2 0 of 23 open

+ Add a Task to This List

- 1 - READY TO TEST
- ISSUE - READY TO TEST
- READY TO TEST

Show 20 more completed tasks

Gambar 3.13 Contoh task dengan label *Ready to Test*

3.2.4 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, *sprint review* dilaksanakan bersamaan dengan kegiatan *User Acceptance Test* (UAT), sehingga *system analyst* akan mengacu dokumen UAT ketika mendemonstrasikan hasil pekerjaan. Dokumen UAT berisi *test case scenario* dari semua *product backlog* yang telah dikelompokkan berdasarkan ruang lingkup pekerjaan seperti pada Gambar 3.6. *Stakeholder* mengisi status dan catatan 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 yang disampaikan oleh *stakeholder* akan ditinjau pada tahap *sprint retrospective*.

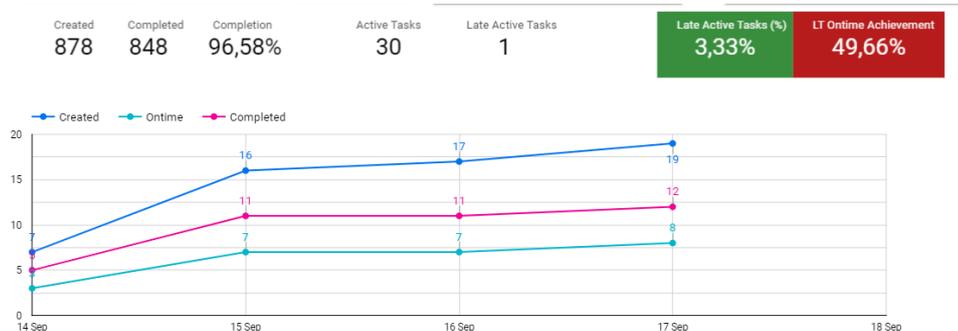
3.2.5 *Sprint retrospective*

Sprint retrospective diadakan setelah dilaksanakannya *sprint review*. Pada tahap ini PM, *Business Analyst*, dan *System analyst* meninjau perubahan yang disampaikan oleh *stakeholder* dalam *sprint review*. Perubahan ini ditinjau apakah masih tercakup dalam ruang lingkup pekerjaan atau tidak. Apabila perubahan tidak tercakup dalam ruang lingkup pekerjaan, maka *stakeholder* perlu membuat dokumen *change request* sebagai bentuk penawaran baru. Dalam praktiknya, terdapat banyak perubahan yang disampaikan oleh *stakeholder* setelah terselenggara UAT dan sosialisasi modul LIQUID di mana perubahan masih mencakup ruang lingkup pekerjaan. Perubahan yang banyak disampaikan berupa penambahan fitur dan perbaikan *bug*, sedangkan waktu yang tersisa dalam kontrak tidak banyak. Hal ini menyebabkan proses pengerjaannya melewati waktu batas kontrak dan bisa menjadi catatan bagi tim *scrum* untuk membatasi permintaan *user* pada pengembangan selanjutnya.

3.2.6 *Weekly Report*

Weekly report merupakan laporan kegiatan proyek yang dikirimkan kepada manajemen perusahaan dan klien setiap minggu. Tujuannya agar manajemen perusahaan dan klien dapat mengetahui sejauh mana *progress* dari pengembangan proyek yang dikerjakan. Setiap hari Kamis, *system analyst* akan mengirimkan *weekly report* kepada manajemen perusahaan dan klien via email. Pada *weekly report* mencakup kendala apa saja yang ditemukan dalam satu minggu, solusi apa yang dilakukan untuk menyelesaikan kendala tersebut, dan pekerjaan apa saja yang telah dikerjakan dalam satu minggu. Selain itu, dalam *weekly report* juga mencantumkan *burn-up chart* dan *issue sentry*. *Sentry* merupakan *tools* yang dapat mendeteksi *error* secara otomatis pada aplikasi, namun pada *weekly report* hanya akan

dicantumkan *error* dengan jumlah *event* terbanyak atau sering terjadi. Pada *burn-up chart* berisi jumlah pekerjaan yang telah direncanakan dan jumlah pekerjaan yang telah diselesaikan seperti pada Gambar 3.14. Sehingga, adanya *burn-up chart* memudahkan pihak perusahaan dan klien dalam memantau *progress* proyek setiap minggunya serta mengetahui capaian dari proyek



Gambar 3.14 Contoh *burn-up chart* pada *weekly report* modul LIQUID

3.3 Praktik Non-Scrum pada Pengembangan modul LIQUID

Dalam *scrum*, peran dan acara merupakan komponen utama yang harus ada dan wajib diterapkan. Kedua komponen tersebut yang menjadi perhatian dalam pembahasan karena ternyata masih ada beberapa hal yang belum sesuai dengan praktik *scrum*. Terdapat beberapa praktik non-*scrum* atau praktik yang tidak sesuai dengan *scrum guide* yang diterapkan dalam pengembangan modul LIQUID. Praktik ketidakaturan *scrum* biasa disebut dengan “*ScrumBut*” di mana dalam kasus ini terjadi pada peran dan acara *scrum*. Ketidaksesuaian dalam peran *scrum* terjadi pada *scrum master* yang merupakan posisi penulis yang juga sebagai *system analyst* dalam pengembangan modul LIQUID. Sedangkan, ketidakaturan pada acara *scrum* yang paling menonjol yaitu pada kegiatan *sprint retrospective*.

Dalam pengembangan modul LIQUID, peran *scrum master* memiliki tugas yang sama seperti *product owner* di mana *scrum master* bertanggung jawab untuk memberikan pekerjaan kepada *development team*. Berbeda dengan *scrum* secara teori di mana *development team* akan memilih pekerjaan dari *product backlog* untuk *sprint*. Perbedaan ini disebabkan karena budaya perusahaan yang tidak sejalan dengan aturan *scrum* dan perusahaan tidak ingin berubah. Adanya kasus tersebut menyebabkan tidak adanya sifat *self-organized* yang diterapkan oleh tim *scrum*, sehingga *development team* akan bekerja sesuai dengan pekerjaan yang telah diberikan oleh *scrum master*. Di sisi lain, *scrum master* juga bertanggung jawab untuk memastikan tim *scrum* dapat menerapkan *scrum* dengan benar. Namun, pada beberapa acara *scrum* terdapat ketidakaturan dalam praktiknya, misalnya pada *sprint planning* dalam

menentukan *product backlog item* untuk *development team* ditentukan oleh *system analyst* dan *daily scrum* durasinya lebih dari 15 menit. Hal ini membuktikan bahwa *scrum master* juga belum menunjang penerapan *scrum* dengan baik dan sesuai.

Praktik ketidakaturan pada acara *scrum* yang paling menonjol terdapat dalam kegiatan *sprint retrospective*. Dalam praktiknya, kegiatan *sprint retrospective* sangat berbeda dengan penerapan *scrum* secara teori. Kegiatan *sprint retrospective* dalam pengembangan modul LIQUID tidak dihadiri oleh seluruh tim *scrum* dan pembahasannya hanya terkait pekerjaan apa yang akan dikerjakan untuk *sprint* berikutnya. Di sisi lain, tidak ada proses evaluasi terhadap *sprint* yang telah dikerjakan sebelumnya. Hal ini menyebabkan kendala atau kekurangan dalam *sprint* sebelumnya dapat terulang kembali pada *sprint* berikutnya.



BAB IV

REFLEKSI PELAKSANAAN MAGANG

4.1 Teknis

4.1.1 Bentuk Implementasi *Scrum*

Proses pengembangan modul LIQUID dilakukan dengan metode pengembangan *agile* dengan kerangka kerja *scrum*. Salah satu manfaat yang diperoleh dari *scrum* adalah fleksibilitas terhadap perubahan, sehingga bisa menyesuaikan apabila terdapat kebutuhan produk yang perlu ditambahkan. Untuk dapat menerapkan kerangka kerja *scrum*, perlu pengetahuan terkait teori, praktik, dan aturan *scrum*. Penggunaan *scrum* dapat dipelajari melalui buku panduan *scrum* yang dapat diakses secara gratis melalui *website scrum*.

Dalam pengembangan modul LIQUID, penerapan *scrum* telah diterapkan sejak proyek dimulai. Penerapan *scrum* mengacu pada aturan dalam buku panduan *scrum* atau *Scrum Guide*. Tabel 4.1 memaparkan perbandingan implementasi *scrum* antara *Scrum Guide* dan *scrum* berdasarkan aktivitas pada pengembangan modul LIQUID.

Tabel 4.1 Perbandingan implementasi *scrum* antara *Scrum Guide* dan modul LIQUID

No	Tahapan <i>Scrum</i>	<i>Scrum Guide</i>	Modul LIQUID
1	<i>Product backlog</i>	Menentukan daftar prioritas kebutuhan yang ditentukan oleh <i>Product Owner</i> .	Menentukan daftar prioritas kebutuhan yang ditentukan oleh PM dan <i>Business Analyst</i> .
2	<i>Sprint planning</i>	Menganalisis pekerjaan yang akan dikerjakan dalam satu <i>sprint</i> . Komponen analisis meliputi: - Menentukan <i>sprint goal</i> - Menentukan <i>product backlog</i> yang dipilih oleh <i>development team</i> : <i>Sprint backlog</i>	Menganalisis pekerjaan yang akan dikerjakan dalam satu <i>sprint</i> . Komponen analisis meliputi: - Menguraikan ruang lingkup - Menentukan pekerjaan untuk setiap anggota - Menentukan <i>sprint goal</i> - Menentukan <i>product backlog</i> yang ditentukan oleh <i>system analyst</i> .
3	<i>Sprint backlog</i>	<i>Sprint backlog</i> menggambarkan keadaan terkini mengenai sisa pekerjaan yang telah direncanakan <i>Development Team</i> untuk diselesaikan dalam <i>sprint</i> dan merupakan hak milik <i>Development Team</i> sepenuhnya. Apabila ada pekerjaan baru yang diperlukan, maka <i>development team</i>	<i>Sprint backlog</i> menggambarkan keadaan terkini mengenai sisa pekerjaan yang telah ditentukan oleh <i>system analyst</i> dalam <i>sprint</i> . Daftar <i>sprint backlog</i> berupa ruang lingkup pekerjaan yang ada pada <i>gant chart</i> . Apabila ada pekerjaan baru yang diperlukan, maka <i>system analyst</i>

		menambahkannya ke <i>sprint backlog</i> .	menambahkan ke dalam <i>sprint backlog</i> .
4	<i>Daily scrum</i>	<i>Daily scrum</i> dilaksanakan setiap hari dan memiliki batas waktu yaitu 15 menit. <i>Development team</i> wajib hadir pada acara ini. Pada acara ini, <i>development team</i> membuat rencana kerja untuk 24 jam ke depan.	<i>Daily scrum</i> dilaksanakan pada hari Senin hingga Kamis dan umumnya dilakukan selama 15 menit atau lebih tergantung pada pembahasannya. <i>Development team</i> wajib hadir pada acara ini dan akan membuat rencana kerja untuk 24 jam ke depan.
5	<i>Sprint review</i>	<i>Sprint review</i> dilaksanakan pada akhir <i>sprint</i> untuk menginspeksi <i>increment</i> dan meninjau <i>product backlog</i> yang telah dan belum diselesaikan. Tim <i>Scrum</i> dan <i>stakeholder</i> akan mendiskusikan terkait <i>product backlog</i> yang akan direvisi dan mendefinisikan <i>item</i> yang mungkin ditambahkan pada <i>sprint</i> selanjutnya.	<i>Sprint review</i> dilaksanakan setiap dua minggu sekali di hari Kamis. Tujuannya untuk menginspeksi <i>increment</i> dan meninjau <i>product backlog</i> yang telah dan belum diselesaikan. Tim <i>scrum</i> dan <i>stakeholder</i> menentukan <i>product backlog</i> yang akan direvisi dan ditambahkan pada <i>sprint</i> selanjutnya. Dalam <i>sprint</i> terakhir, pelaksanaan <i>sprint review</i> dilaksanakan bersamaan dengan kegiatan UAT.
6	<i>Sprint retrospective</i>	<i>Sprint retrospective</i> diselenggarakan oleh tim <i>scrum</i> setelah <i>sprint review</i> untuk menginspeksi dirinya sendiri dan membuat perencanaan mengenai peningkatan yang akan dilakukan di <i>sprint</i> berikutnya. Tim <i>scrum</i> juga akan meninjau pekerjaan yang dilakukan dan yang tidak dilakukan untuk <i>sprint</i> selanjutnya.	<i>Sprint retrospective</i> diselenggarakan oleh <i>Project Manager</i> , <i>Business Analyst</i> , dan <i>System analyst</i> setelah <i>sprint review</i> . Acara ini meninjau pekerjaan yang akan dikerjakan atau tidak dikerjakan untuk <i>sprint</i> selanjutnya. Pekerjaan akan ditinjau berdasarkan ruang lingkup pekerjaan yang sudah ditetapkan sebelumnya.

Penerapan *scrum* dalam pengembangan modul LIQUID menghasilkan dampak positif maupun negatif. *Scrum* memiliki keunggulan dalam menangani proyek yang kompleks dan banyak perubahan. Hal itu berdampak pada penerapan *scrum* dalam pengembangan modul LIQUID di antaranya sebagai berikut:

- a. Setiap masalah yang timbul dapat segera diidentifikasi dengan baik pada *daily scrum*, sehingga masalah dapat diselesaikan dengan cepat.
- b. *Monitoring* pekerjaan menjadi lebih mudah dan cepat karena adanya *sprint*.

- c. Peninjauan sisa pekerjaan dalam satu *sprint* dapat dilihat dengan mudah karena adanya *sprint backlog*.
- d. *Feedback* yang disampaikan *stakeholder* dapat diterima tanpa harus menunggu hasil sepenuhnya diimplementasikan.
- e. Adanya kolaborasi tim yang kuat sehingga produk memiliki nilai yang tinggi.

Selama berada di proyek modul LIQUID, penulis pernah mengalami kesulitan yaitu ketika mendampingi sosialisasi modul LIQUID. Kegiatan tersebut dibagi menjadi 4 *batch*, di mana setiap *batch* akan dihadiri 8 unit pelaksana dan 15 divisi dari kantor pusat. Dalam setiap *batch*, penulis mendapatkan *feedback* dari pengguna aplikasi berupa *bug* dan penambahan fitur. Kesulitan yang dialami adalah ketika banyak pengguna yang melaporkan *bug* dan meminta untuk segera diselesaikan. Sehingga, penulis terkadang kewalahan untuk memprioritaskan mana yang harus diselesaikan lebih dahulu padahal sudah memasuki tenggat waktu proyek. Oleh karena itu, hal yang dilakukan adalah membatasi permintaan *user* yang berupa penambahan fitur dan fokus terhadap penanganan *bugs* hingga pada akhirnya proyek selesai dalam waktu 5 bulan.

4.1.2 Bentuk Implementasi Non-Scrum

Dalam buku panduan *scrum*, terdapat beberapa elemen utama yang harus diterapkan dalam penggunaan *scrum* terutama pada peran dan acara. Namun, kedua komponen tersebut menjadi perhatian dalam pembahasan ini karena terdapat beberapa hal yang tidak sesuai dengan praktik dalam buku panduan *scrum*. Dalam *scrum*, apabila hanya menerapkan sebagian dari *scrum* saja, maka *scrum* tidak dapat disebut sebagai *scrum* (Schwaber & Sutherland, 2017). Adapun proses ketidakaturan *scrum* atau tidak mengikuti semua aturan *scrum* disebut dengan *ScrumBut*.

Perbedaan penerapan *scrum* terjadi ketika mengembangkan modul LIQUID yaitu pada peran dan pelaksanaan acara *scrum*. Dalam peran pada pengembangan modul LIQUID memiliki bagian yang sama dengan *scrum* teori, namun terdapat perbedaan pada tanggung jawab pada masing-masing peran seperti pada Tabel 4.2. Pada acara *scrum* terdiri dari *sprint* yang merupakan jantung dari *scrum*. Adapun perbedaan penerapan acara *scrum* antara *Scrum Guide* dan *scrum* pada pengembangan modul LIQUID telah dipaparkan pada Tabel 4.3.

Tabel 4.2 Perbedaan implementasi *scrum* berdasarkan peran

No	Peran Scrum	Scrum Guide	Modul LIQUID	Gap
1	<i>Product Owner</i>	<i>Product owner</i> bertanggung jawab	<i>Product owner</i> terdiri dari <i>Project Mananger</i>	<i>Project manager</i> dan <i>Business Analyst</i>

		<p>untuk memaksimalkan nilai bisnis dari produk, mengelola <i>product backlog</i>, dan menyampaikan isi <i>product backlog item</i> yang akan dikerjakan oleh <i>development team</i>.</p>	<p>dan <i>Business Analyst</i>. Keduanya bertanggung jawab dalam mengelola proyek agar mencapai tujuan, mengoptimalkan nilai bisnis dari produk, dan memastikan <i>product backlog</i> jelas bagi tim <i>scrum</i>.</p>	<p>memiliki tugas yang sesuai, namun keduanya juga menjadi <i>product owner</i> di proyek lain atau bekerja secara multiproyek.</p>
2	<i>Development Team</i>	<p><i>Development team</i> bertanggung jawab dalam membuat dan mengirimkan produk yang dapat dirilis pada setiap akhir "<i>sprint</i>". Selain itu, <i>development team</i> juga bertanggung jawab atas <i>product backlog</i> yang sudah disampaikan oleh <i>product owner</i>, mereka akan menyusun dan mengelola pekerjaan mereka sendiri</p>	<p><i>Development team</i> terdiri dari <i>Programmer</i>, <i>QA</i>, <i>Designer</i>, dan <i>Technical Document</i>. Peran-peran tersebut memiliki tugas yang berbeda-beda, namun tetap memiliki tujuan yang sama yaitu mengirimkan produk yang dapat dirilis pada akhir <i>sprint</i>.</p>	<p><i>Development team</i> tidak bersifat <i>self-organized</i> karena pekerjaan akan ditentukan oleh <i>system analyst</i> untuk masing-masing anggota.</p>
3	<i>Scrum Master</i>	<p><i>Scrum Master</i> bertanggung jawab untuk memfasilitasi penggunaan <i>scrum</i> sesuai dengan praktik dan tata nilai <i>scrum</i>, serta menghilangkan hambatan yang memperlambat perkembangan pekerjaan <i>development team</i>. Selain itu, <i>scrum</i></p>	<p><i>Scrum Master</i> terdiri dari <i>system analyst</i> yang bertanggung jawab dalam pengelolaan <i>sprint</i>, menyampaikan isi dari <i>product backlog item</i> dalam bentuk <i>task</i>, memprioritaskan <i>feedback</i>, dan memberikan tugas kepada <i>development team</i>. Selain itu, <i>scrum</i></p>	<p><i>Scrum Master</i> dan <i>product owner</i> memiliki tanggung jawab yang hampir sama yaitu dalam pengelolaan proyek seperti memberikan pekerjaan kepada <i>development team</i>. Selain itu, <i>scrum master</i> masih belum menunjang</p>

	master juga memastikan bahwa tim scrum dapat mengimplementasikan scrum secara baik dan sesuai.	master juga memfasilitasi penggunaan <i>scrum</i> dan menghilangkan hambatan yang muncul.	penggunaan <i>scrum</i> dengan baik dan sesuai.
--	--	---	---

Berdasarkan Tabel 4.2 peran *scrum* dalam pengembangan modul LIQUID tidak sepenuhnya sesuai dengan *scrum* secara teori terutama peran *scrum master*. Pada praktiknya, *scrum master* memiliki tanggung jawab yang hampir sama dengan *product owner* yaitu dalam menyampaikan isi *product backlog item* dan memberikan pekerjaan kepada *development team*. Sehingga, terkadang *scrum master* harus bekerja secara *multitasking* antara menjalankan tanggung jawab sebagai *scrum master* dan *product owner*. *Scrum master* juga harus menjaga komunikasi dengan *product owner* terkait prioritas kebutuhan, namun terkadang komunikasi yang dibutuhkan tidak dapat mendesak karena peran *product owner* yang multiproyek. Di sisi lain, terdapat ketidakaturan dalam beberapa acara *scrum* karena *scrum master* masih belum menunjang penggunaan *scrum* dengan baik dan sesuai. Perbedaan tersebut menjadi sebuah tantangan bagi *scrum master* untuk dapat bekerja secara *multitasking* di antaranya dalam membagi waktu dan fokus untuk menjalankan tanggung jawab sebagai *product owner* maupun *scrum master*. Selain itu, *scrum master* juga harus memahami teori *scrum* lebih dalam lagi agar *scrum* dalam pengembangan modul LIQUID dapat diterapkan dengan baik dan sesuai.

Tabel 4.3 Perbedaan implementasi *scrum* berdasarkan acara

No	Acara Scrum	Scrum Guide	Modul LIQUID
1	<i>Sprint planning</i>	<i>Product backlog</i> yang dipilih untuk <i>sprint</i> merupakan keputusan <i>development team</i> sepenuhnya.	<i>Product backlog</i> yang dipilih untuk <i>sprint</i> merupakan keputusan <i>system analyst</i> .
2	<i>Sprint backlog</i>	Ketika ada pekerjaan baru yang diperlukan, maka <i>development team</i> menambahkan ke dalam <i>sprint backlog</i> .	Ketika ada pekerjaan baru yang diperlukan, maka <i>system analyst</i> menambahkan ke dalam <i>sprint backlog</i> .
3	<i>Daily scrum</i>	<i>Daily scrum</i> dilaksanakan setiap hari dan memiliki batas waktu yaitu 15 menit.	<i>Daily scrum</i> dilaksanakan pada hari Senin hingga Kamis. Batasan waktu bisa lebih dari 15 menit tergantung pada jenis pembahasannya.

4	<i>Sprint review</i>	<i>Sprint review</i> dilaksanakan pada akhir <i>sprint</i> untuk menginspeksi <i>increment</i> dan meninjau <i>product backlog</i> yang telah dan belum diselesaikan.	<i>Sprint review</i> dilaksanakan setiap dua minggu di hari Kamis. Dalam <i>sprint</i> terakhir, pelaksanaan <i>sprint review</i> dilaksanakan bersamaan dengan UAT.
5	<i>Sprint retrospective</i>	<i>Sprint retrospective</i> diselenggarakan oleh tim <i>scrum</i> setelah <i>sprint review</i> untuk menginspeksi dirinya sendiri dan membuat perencanaan mengenai peningkatan yang akan dilakukan di <i>sprint</i> berikutnya. Tim <i>scrum</i> juga akan meninjau pekerjaan yang dilakukan dan yang tidak dilakukan untuk <i>sprint</i> selanjutnya.	<i>Sprint retrospective</i> hanya diselenggarakan oleh <i>Project Manager</i> , <i>Business Analyst</i> , dan <i>System analyst</i> setelah <i>sprint review</i> . Acara ini meninjau pekerjaan yang akan dikerjakan atau tidak berdasarkan ruang lingkup pekerjaan yang sudah ditetapkan sebelumnya. Tidak ada proses inspeksi terhadap penerapan <i>scrum</i> .

Tabel 4.3 memaparkan ketidaksesuaian pada acara scrum yang telah diterapkan pada pengembangan modul LIQUID terutama acara *sprint retrospective*. Pada praktiknya, *sprint retrospective* yang diterapkan sangat berbeda dengan penjelasan dalam *scrum* teori karena tidak ada evaluasi terkait penerapan *scrum* pada *sprint* sebelumnya. Adanya perbedaan pada peran dan acara *scrum* dalam pengembangan modul LIQUID berdampak pada beberapa hal seperti berikut:

- a. Tidak ada *self-organized* yang diterapkan oleh *development team* karena pekerjaan ditentukan oleh *system analyst*. Sehingga, pekerjaan *development team* terkadang terlalu banyak dan mengalami *overdue*.
- b. Batasan waktu dalam *daily scrum* tidak sesuai dengan aturan *scrum*, sehingga memakan banyak waktu tim *scrum* yang tidak berkepentingan. Hal ini disebabkan karena *scrum master* belum menunjang penerapan *scrum* dengan baik dan sesuai.
- c. Kurangnya informasi yang diterima oleh *development team* karena pada *sprint retrospective* hanya dilaksanakan oleh *Project Manager*, *Business Analyst*, dan *System analyst*.
- d. Tidak ada peningkatan terkait penerapan *scrum* dalam *sprint retrospective* sehingga kendala atau kekurangan pada *sprint* sebelumnya dapat terulang kembali.

Meskipun demikian, tim *scrum* tetap menerapkan kerangka kerja *scrum* dengan bantuan *tools* manajemen proyek pada acara *scrum* agar praktik *scrum* yang diterapkan dapat mempermudah dalam mengelola proyek. Beberapa *tools* yang digunakan di antaranya adalah penggunaan *gant chart* pada *sprint planning* dan pembuatan *weekly report*. Penggunaan kedua

tools tersebut dapat membantu dalam menangani kendala yang muncul. Berikut penjelasan *tools* manajemen proyek yang telah diterapkan.

Penggunaan grafik *gant chart* memudahkan dalam proses *monitoring* kemajuan proyek. Dengan adanya *gant chart*, penulis dapat melihat perkembangan proyek dan persentase ruang lingkup pekerjaan yang telah selesai. Selanjutnya, pembuatan *weekly report* membantu untuk menggambarkan capaian *progress* yang telah dilakukan setiap minggunya. Dalam *weekly report* terdapat *burn-up chart* yang berguna untuk memantau *progress* proyek dengan memaparkan jumlah pekerjaan yang telah selesai dikerjakan. Dengan menggunakan *tools* tersebut, *monitoring* dan *controlling* pekerjaan dalam pengembangan modul LIQUID menjadi lebih mudah dan cepat.

4.1.3 Proyek Lain

Selama magang berlangsung terdapat beberapa proyek terlibat yang terkait dalam implementasi *scrum*. Penulis berperan sebagai *system analyst* dalam proyek Dashboard Bappeda dan E-Jawatan & E-HR yang dikembangkan dengan kerangka kerja *scrum*. Praktik *scrum* dalam proyek tersebut juga mengalami ketidaksesuaian penerapan *scrum* terutama dalam peran dan acara *scrum*. Misalnya pada proyek Dashboard Bappeda tidak ada kegiatan *daily scrum* dan kegiatan *sprint review* hanya dilakukan oleh tim *scrum*. Selanjutnya, pada proyek E-Jawatan & E-HR, peran *scrum master* lebih condong kepada *development team* dan tidak ada kegiatan *daily scrum*. Sebagai *scrum master* dalam proyek tersebut, penulis kurang menggali lebih dalam terkait *scrum* yang diterapkan, sehingga masih terdapat ketidaksesuaian dalam penerapannya. Maka, hal tersebut menjadi pengalaman dan pelajaran bagi penulis untuk memahami dan mempraktikkan *scrum* secara benar agar kedepannya praktik *scrum* bisa jauh lebih baik.

Selain proyek yang terkait dengan implementasi *scrum*, penulis juga terlibat dalam pembuatan surat menggunakan *JasperReports* pada proyek pengembangan aplikasi Sistem Informasi Manajemen Penyelesaian Laporan (SIMPeL). *JasperReports* merupakan *reporting tools* untuk memudahkan dalam pembuatan surat atau laporan dalam jumlah banyak. Dengan *JasperReports* pembuatan surat menjadi mudah dan cepat karena adanya fitur *drag and drop* elemen-elemen yang tersedia. Pembuatan surat atau laporan dengan *JasperReports* tidak hanya diterapkan pada proyek SIMPeL. Beberapa proyek yang telah terlibat dan menggunakan *JasperReports* yaitu proyek E-Jawatan & E-HR dan PPSDM. Penggunaan *JasperReports* merupakan hal baru bagi penulis. Namun, seiring berjalannya waktu dan terbiasa membuat surat dengan *JasperReports*, penulis semakin paham.

4.2 Non Teknis

4.2.1 Komunikasi

Selama pelaksanaan magang di PT Javan Cipta Solusi, terdapat hal non-teknis yang penulis dapatkan. Salah satunya adalah komunikasi. Dalam bekerja, komunikasi menjadi salah satu kunci keberhasilan tim. Komunikasi merupakan hal yang penting untuk menjaga keselarasan dan menghindari adanya *misunderstanding* dalam sebuah tim. Komunikasi dilakukan secara tertulis dengan *chatting* ataupun secara lisan dengan *conference call*. Misalnya ketika penulis kurang paham atau butuh konfirmasi terkait pekerjaan yang diberikan, maka penulis harus bertanya kepada tim atau *supervisor*. Selain dalam sebuah tim, komunikasi juga dibutuhkan untuk *stakeholder*. Hal ini dilakukan apabila terjadi kendala atau terdapat suatu hal yang harus di konfirmasi. Dengan adanya tuntutan untuk aktif berkomunikasi, penulis menjadi lebih berani dan percaya diri.

4.2.2 Knowledge

Pengetahuan dan pengalaman juga didapatkan oleh penulis ketika menjadi seorang *system analyst*. Sebagai contoh adalah kegiatan hari “Jumat Berkualitas” yang diadakan oleh PT Javan Cipta Solusi setiap hari Jumat. Kegiatan tersebut terdiri dari sesi *review project* dan sesi training. Sesi *review project* dilakukan oleh tim proyek tertentu yang akan menyampaikan terkait *progress* dan kendala proyek kepada penonton. Pada umumnya *review project* dilakukan oleh *system analyst* kemudian penonton akan memberikan saran atau kritik terhadap apa yang telah dipresentasikan. Sedangkan untuk sesi training merupakan kegiatan sharing materi terkait teknologi maupun non-teknologi. Pembicara dalam sesi training akan bergilir setiap pekannya. Dengan adanya kegiatan “Jumat Berkualitas” penulis mendapat pengetahuan dari berbagai materi yang telah disampaikan, seperti kemampuan analisis dan manajemen diri. Misalnya, pada materi analisis penulis mendapat ilmu tentang *automation testing* dengan selenium dan visualisasi dengan metabase, sedangkan pada manajemen diri penulis mendapat ilmu dan kesempatan menjadi pembicara tentang materi *self-control*.

Selain adanya kegiatan Jumat Berkualitas, PT Javan Cipta Solusi juga mengadakan kegiatan *upgrading skill* pada setiap karyawan. Kegiatan ini bertujuan untuk meninjau *skill* setiap karyawan, sehingga dapat mengetahui apakah karyawan sudah memenuhi *skill* yang ditentukan atau belum. Apabila karyawan belum memenuhi *skill* yang ditentukan, maka akan diadakan sesi *upgrading skill* pada waktu yang sudah ditentukan HCD. Kegiatan ini dibagi

menjadi dua divisi, yaitu divisi *engineer (programmer)* dan divisi *solution development (system analyst dan tester)*. Materi dari *upgrading skill* ditentukan berdasarkan kedua divisi tersebut.

4.2.3 Self-Management

Berada di dunia kerja sangat berbeda dibandingkan dengan dunia perkuliahan. Hal ini karena adanya tuntutan dan tanggung jawab yang telah diberikan oleh perusahaan. Maka dari itu, penulis perlu menerapkan manajemen diri untuk dapat mengatur dan mengelola diri agar tercapai *goal* yang diinginkan. Penulis telah menerapkan manajemen diri pada beberapa aspeknya salah satunya yaitu pengelolaan waktu.

Dalam menjalani magang dan kuliah, tentunya penulis harus dapat mengatur waktu dengan baik dalam menyelesaikan pekerjaan kantor maupun tugas kuliah dengan tepat waktu dan sebaik mungkin. Selain itu, dalam bekerja seringkali penulis merasa jenuh terutama ketika menjalani *Work From Home (WFH)*. Maka dari itu, penulis juga memerlukan kemampuan untuk memotivasi diri agar pekerjaan dan tugas kuliah tidak menumpuk, misalnya membiasakan untuk tidak menunda waktu dan membuat *to-do list* supaya aktivitas sehari-hari lebih tertata.

4.2.4 Menemukan Passion

Passion merupakan keinginan seseorang untuk melakukan sesuatu yang disukai atau dianggap penting untuk dilakukan. *Passion* dapat ditemukan ketika seseorang mencoba hal baru. Selama magang di PT Javan Cipta Solusi, penulis sempat berada di beberapa posisi yaitu *programmer*, *tester*, dan *system analyst*. Dari ketiga posisi yang digeluti, menjadi *system analyst* merupakan posisi yang cocok untuk penulis. Pada saat menjadi *system analyst*, penulis mulai belajar dari berbagai proyek tentang bagaimana menyelesaikan permasalahan dengan tepat. Meskipun begitu, penulis juga harus belajar sedikit demi sedikit dan belajar dari kesalahan yang pernah dilakukan.

BAB V

KESIMPULAN DAN SARAN

5.1 Kesimpulan

Pada umumnya, kerangka kerja *scrum* cocok untuk diterapkan pada proyek yang kompleks adaptif seperti pada proyek pengembangan modul LIQUID. Karena dalam pengembangan, modul LIQUID membutuhkan banyak interaksi dengan *stakeholder* yang memunculkan penambahan dan perubahan kebutuhan yang harus segera diimplementasikan dalam aplikasi. Selain itu, *monitoring* dan *controlling* pekerjaan menjadi lebih mudah karena adanya *sprint*.

Scrum merupakan kerangka kerja yang mudah dipahami, namun sulit untuk diterapkan. Keberhasilan dalam proses implementasi *scrum* bergantung pada pemahaman tentang *scrum* dengan mengikuti aturan *Scrum Guide*. Dalam praktik *scrum* pada pengembangan modul LIQUID terdapat beberapa komponen dalam peran dan acara *scrum* yang tidak sesuai dengan buku panduan *scrum*. Penerapan *scrum* dalam pengembangan modul LIQUID, terdapat ketidaksesuaian pada peran *scrum* terkait tanggung jawab dari setiap perannya. Selain itu, pada acara *scrum* juga terdapat ketidaksesuaian yang muncul pada saat *sprint* berlangsung.

Adanya perbedaan tersebut disebabkan oleh peraturan dan kultur organisasi yang sudah ada dan tidak sejalan dengan aturan *scrum*. Selain itu, peran dari *scrum master* belum menunjang penggunaan *scrum* dengan baik dan benar. Perbedaan ini menyebabkan dampak pada penerapan *scrum* dalam pengembangan modul LIQUID di antaranya:

- a. Pekerjaan untuk *development team* terlalu banyak dan mengalami *overdue*.
- b. Memakan banyak waktu tim *scrum* bagi yang tidak berkepentingan.
- c. Kurangnya informasi yang diterima oleh *development team*.
- d. Tidak ada peningkatan dalam penerapan *scrum*.

Adanya perbedaan dalam acara *scrum* dapat berdampak pada nilai produk yang dihasilkan. Untuk memberikan hasil praktik yang lebih baik, penerapan *scrum* dibantu dengan *tools* manajemen proyek pada acara *scrum* di antaranya yaitu penggunaan *ganttt chart* dan *Weekly Report*. *Tools* tersebut berhasil diterapkan pada pengembangan modul LIQUID dengan baik sejak proyek dimulai hingga memasuki masa retensi. Penerapan *tools* berdampak pada pengembangan modul LIQUID di antaranya:

- a. *Monitoring* kemajuan proyek dapat dilakukan dengan mudah dan hambatan dapat terdeteksi dengan cepat.
- b. Dapat mengetahui persentase *progress* pengembangan proyek dengan jelas.
- c. Pekerjaan yang akan dikerjakan dalam *sprint* menjadi lebih tertata dan jelas.
- d. Jumlah pekerjaan yang telah diselesaikan pada setiap minggu atau hari dapat dilihat dengan mudah.

5.2 Saran

Dalam implementasi *scrum* pada pengembangan modul LIQUID yang telah dilakukan diberikan saran demi kesempurnaan ke depannya. Saran yang diberikan terutama pada peran *scrum* master terkait pemahaman penerapan *scrum*. Hal ini perlu ditingkatkan karena peran *scrum* master mempengaruhi dalam kesesuaian penerapan *scrum*. Selanjutnya, saran yang diberikan yaitu membatasi *stakeholder* dalam memberikan *feedback* terkait perubahan atau penambahan fitur. *Feedback* tersebut perlu dibatasi agar target waktu dapat tercapai. Selain itu, pada *sprint retrospective* perlu dilakukan evaluasi terkait penerapan *scrum* dan dihadiri seluruh tim *scrum*. Hal ini perlu dilakukan agar kendala atau kekurangan dalam *sprint* sebelumnya tidak terulang kembali dan tim *scrum* tidak ada yang mengalami *miscommunication*.

DAFTAR PUSTAKA

- Adi, P. (2015). *Scrum Method Implementation in a Software Development Project Management. International Journal of Advanced Computer Science and Applications*, 6(9), 198–204. <https://doi.org/10.14569/ijacsa.2015.060927>
- Akif, R., & Majeed, H. (2012). Issues and Challenges in *Scrum* Implementation. *International Journal of Scientific & Engineering Research*.
- Anzhari, I., J Sumampouw, H., & Kaparang, S. G. (2016). PENGARUH PERILAKU PEMIMPIN TERHADAP KINERJA KARYAWAN PT. BANK TABUNGAN NEGARA (PERSERO) TBK. KANTOR CABANG MANADO. *Jurnal Administrasi Bisnis*, 4(1), 1–6. <https://doi.org/10.35797/jab.4.1.2016.11278>.
- Blankenship, J., Bussa, M., Millett, S., Blankenship, J., Bussa, M., & Millett, S. (2011). The Art of Agile Development. In *Pro Agile .NET Development with Scrum*. https://doi.org/10.1007/978-1-4302-3534-7_1
- Cockburn, A., & Highsmith, J. (2001). Agile software development: The people factor. *Computer*. <https://doi.org/10.1109/2.963450>
- Ependi, U. (2018). Implementasi Model *Scrum* pada Sistem Informasi Seleksi Masuk Mahasiswa Politeknik Pariwisata Palembang. *Jurnal Informatika: Jurnal Pengembangan IT*.
- Fakhril. (2019). *AGILE dan SCRUM*. <https://medium.com/@fpvariel/agile-dan-scrum-ec76b832ce00>
- Fuadi, M. N. (2020). *ScrumBut — Deviasi Implementasi Scrum di Sektor Industri (Studi Kasus : Ralali). 1*.
- Hassani-Alaoui, S., Cameron, A.-F., & Giannelia, T. (2020). “We Use *Scrum*, but ...”: Agile Modifications and Project Success. *Proceedings of the 53rd Hawaii International Conference on System Sciences*. <https://doi.org/10.24251/hicss.2020.765>
- Krishna, V., & Basu, A. (2012). *Scrum*;: Is it *ScrumBut*; or *ScrumAnd*; <https://doi.org/10.1109/indcon.2011.6139625>
- Lei, H., Ganjezadeh, F., Jayachandran, P. K., & Ozcan, P. (2017). A statistical analysis of the effects of *Scrum* and Kanban on software development projects. *Robotics and Computer-Integrated Manufacturing*. <https://doi.org/10.1016/j.rcim.2015.12.001>
- PLN. (n.d.). *Sistem Komunikasi dan Budaya Organisasi PT. PLN*. Retrieved November 21, 2020, from <https://budaya.pln.co.id/auth/login>

- Redaksi. (2017). *PLN Aceh Gelar Liquid Forum*.
<https://harianrakyataceh.com/2017/04/26/pln-aceh-gelar-liquid-forum/>
- Schwaber, K. (n.d.). *What is ScrumBut?* Retrieved March 20, 2021, from
<https://www.scrum.org/resources/what-ScrumBut>
- Schwaber, K., & Beedle, M. (2001). *Agile Software Development with Scrum*. In *cdswebcernch*.
- Schwaber, K., & Sutherland, J. (2017). Panduan Definitif untuk *Scrum*: Aturan Main. *Imperial Journal of Interdisciplinary Research (IJIR)*.
- Sholahudin, A., Paulus, E., & Padjadjaran Jl Raya Bandung-Sumedang Km, U. (2017). *Analisis dan Desain Pengembangan Program Aplikasi DMS dengan Metodologi Scrum Framework*. September, 47–52.
<http://repository.unjani.ac.id/repository/9ec3721ead4436680675214b6fc56247.pdf>
- Sommerville, I. (2010). *Software Engineering (9th Edition)*. Pearson. In *Software Engineering*.
- Tohirin, T., & Widiyanto, S. R. (2020). Peran Trello dalam Adopsi Agile *Scrum* Pada Pengembangan Sistem Informasi Kesehatan. *Multinetics*, 6(1), 32–39.
<https://doi.org/10.32722/multinetics.v6i1.2763>
- Wan, I. I. (2019). Umpan Balik Sistem Pakar Sebagai Penilaian Kinerja Karyawan Pada PT. Multistrada Arah Sarana, Tbk. *BMAJ: Business Management Analysis Journal*, 2(1), 52–67. <https://doi.org/10.24176/bmaj.v2i1.3211>

LAMPIRAN

