

# Pengembangan Aplikasi *People Management* dalam *Software Development* dengan Pendekatan *Project Management Antipatterns* (*People Management Application in Software Development Based on Project Management Antipatterns*)

Reza Aprillia Arshanty<sup>1</sup>, Umi Sa'adah<sup>2</sup>, Desy Intan Permata Sari<sup>3</sup>, Maulidan Bagus A. Rasyid<sup>4</sup>

**Abstract**—Human factor in software development process is one of the determinants of success of a project, regardless of the technology and tools used. In practice, many project managers do not have any dataset about hard skills and soft skills of their members. This led the software development project team to be trapped in an antipatterns irrational management situation. Based on several main factors that cause software development project failures, a mitigation plan is needed to assist the project manager in managing human resources in their team. The mitigation plan that can be done is to develop people management applications based on principles of Project Management Antipatterns. This research proposes new modeling in the form of applications for people management in software development with the Project Management Antipatterns approach. The results given are in the form of mapping the developer team to the relevant task based on hard skills and soft skills possessed. Besides, this application also provides results in the form of treatment recommendations that can be done to the team to increase productivity and reduce the potential risks arising from the diagnosis of poor practice. The approach used is proven to be able to increase overall team productivity. The refactored solution also proved effective in reducing the value of antipatterns from one sprint to the next.

**Intisari**—Faktor manusia dalam proses pengembangan perangkat lunak menjadi salah satu penentu keberhasilan suatu proyek, terlepas dari teknologi dan *tools* yang digunakan. Dalam praktiknya, banyak *project manager* tidak memiliki *dataset* mengenai *hard skill* dan *soft skill* anggotanya. Hal ini mengantarkan tim proyek pengembangan perangkat lunak terjebak dalam situasi *antipatterns irrational management*. Berdasarkan beberapa faktor utama penyebab kegagalan proyek pengembangan perangkat lunak, diperlukan adanya suatu *mitigation plan* yang dapat membantu *project manager* dalam melakukan manajemen sumber daya manusia dalam sebuah tim. *Mitigation plan* yang dapat dilakukan adalah dengan mengembangkan aplikasi *people management* yang didasarkan pada prinsip-prinsip *Project Management Antipatterns*. Makalah ini mengajukan suatu pemodelan baru dalam bentuk aplikasi untuk melakukan *people management* di dalam pengembangan perangkat lunak dengan pendekatan *Project Management Antipatterns*. Penelitian ini mengajukan suatu pemodelan baru dalam bentuk aplikasi untuk melakukan *people management* di

dalam pengembangan perangkat lunak dengan pendekatan *Project Management Antipatterns*. Hasil yang diberikan berupa pemetaan tim *developer* kepada *task* yang relevan berdasarkan *hard skill* dan *soft skill* yang dimiliki. Selain itu, aplikasi ini juga memberikan hasil berupa rekomendasi *treatment* yang bisa dilakukan terhadap tim untuk meningkatkan produktivitas serta mengurangi potensi risiko yang ditimbulkan dari hasil diagnosis praktik yang buruk. Pendekatan yang digunakan terbukti mampu meningkatkan produktivitas tim secara keseluruhan. *Refactored solution* yang direkomendasikan juga terbukti efektif menurunkan nilai *antipatterns* dari satu *sprint* ke *sprint* berikutnya.

**Kata Kunci**—*People Management, Project Management Antipatterns, Treatment, Refactored Solution.*

## I. PENDAHULUAN

Industri yang ada saat ini membutuhkan keahlian-keahlian yang diperlukan untuk menyongsong revolusi industri 4.0. Industri di bidang teknologi informasi (TI) pun saat ini membutuhkan beragam jenis keahlian tersebut. Salah satu keahlian yang diperlukan adalah keahlian manajerial, khususnya di bidang *people management*. *People management* sendiri terdaftar dalam sepuluh jenis keahlian teratas pada tahun 2020 versi *World Economic Forum* [1].

Sebuah proyek memerlukan seorang pimpinan yang disebut sebagai *project manager*. Dalam dunia manajemen proyek di industri TI, *project manager* adalah orang yang bertanggung jawab terhadap kesuksesan perencanaan dan eksekusi suatu proyek pengembangan perangkat lunak. Proyek ini relatif lebih kompleks dan fleksibel untuk dikembangkan serta diintegrasikan dibandingkan dengan proyek tradisional. *People management* masuk ke dalam salah satu bagian dari *project management* dan menempati urutan keempat dari sepuluh penyebab utama kegagalan *software development* (pengembangan perangkat lunak) [2]. Buruknya keahlian dalam bidang manajerial yang dimiliki oleh seorang *project manager* dalam industri TI dapat menjadi penyebab utama kegagalan suatu proyek. Selain itu, permasalahan mengenai staf atau anggota tim juga menjadi penyebab kegagalan proyek pengembangan perangkat lunak. Tim pengembang perangkat lunak adalah sekumpulan individu yang melakukan aktivitas pengembangan yang melibatkan banyak aktivitas manusia yang penuh dengan ketidakpastian (*human-centric*). Ketidakpastian yang dimaksudkan meliputi aspek-aspek kepribadian dan keahlian yang dimiliki oleh setiap anggota tim, yang memiliki kepribadian dan keahlian berbeda-beda. Faktor manusia dalam proses pengembangan perangkat lunak menjadi

<sup>1,2,3</sup> Politeknik Elektronika Negeri Surabaya, Kampus ITS Jl. Raya ITS, Keputih, Sukolilo 60111 (telp: 031 594 7280; fax: 031 594 6114; e-mail: rezaprillia@it.student.pens.ac.id; umi@pens.ac.id; desy@pens.ac.id)

<sup>4</sup> Research and Development Department PT Maulidan Teknologi Kreatif, Jl. Klampis Anom VIII F-150, Surabaya, 60117, Indonesia (e-mail: contact@maulidangames.com)

salah satu penentu keberhasilan suatu proyek, terlepas dari teknologi dan *tools* yang digunakan. Dari segi *project manager* selaku pimpinan dalam proyek, diperlukan keahlian manajerial, terutama keahlian untuk melakukan manajemen anggotanya. Dari segi anggota tim *developer*, diperlukan keahlian berupa *hard skill* dan *soft skill* yang sesuai dengan *task* yang akan dikerjakan selama masa pengembangan.

Dalam praktiknya, banyak *project manager* tidak memiliki *dataset* mengenai *hard skill* dan *soft skill* anggotanya. Hal ini mengantarkan tim proyek pengembangan perangkat lunak terjebak dalam situasi *antipattern irrational management*. *Irrational management* adalah sebuah kondisi manajemen yang buruk, yang secara signifikan mengarah pada ketidakmampuan manajer dalam mengarahkan timnya. *Irrational management* juga disebabkan oleh ketidakmampuan seorang manajer dalam mengolah fakta-fakta yang ada menjadi data. Masalah lain yang sering dihadapi adalah pemberian beban *task* kepada tim *developer* yang kurang relevan dengan keahlian yang dimiliki (*extinct by instinct*) [3].

Terdapat hubungan antara komposisi kepribadian tim dan kinerja tim dalam tim skala kecil serta pengaruh heterogenitas kepribadian terhadap kinerja tim. Dalam penelitian yang pernah dilakukan, dideskripsikan tiga pembagian peran yang dipetakan dalam tim pengembangan perangkat lunak, yaitu *leader*, *system analyst*, dan *programmer*. Dari hasil penelitian tersebut didapatkan bahwa manajer harus memilih anggota tim berdasarkan tipe kepribadian yang sesuai. *Leader* sebaiknya memiliki kepribadian *N* dan *J*, *system analyst* sebaiknya memiliki kepribadian *T* dan *S*, sedangkan *programmer* sebaiknya memiliki kepribadian *E*, *S*, dan *J* [4]. Alokasi sumber daya manusia dalam sebuah tim pengembangan perangkat lunak dapat ditentukan berdasarkan karakteristik tingkat individu dan tim. Pada tingkat individu, tingkat pemahaman *developer* terhadap *task* yang diberikan diperhitungkan karena memengaruhi kinerja *developer* tersebut. Setiap *developer* memiliki tingkat produktivitas berbeda-beda, tergantung kemampuan dan pengalaman masing-masing. Pedoman yang paling umum digunakan untuk memperkirakan produktivitas *developer* adalah dengan menggunakan metode COCOMO II. Alokasi yang dihasilkan lebih realistis dan efisien berdasarkan pendekatan yang digunakan. Hasil dari validasi yang dilakukan menunjukkan bahwa pendekatan yang digunakan dapat secara efektif mengurangi waktu yang dibutuhkan dalam optimasi dan mempercepat waktu untuk memasarkan proyek [5]. Namun, metode ini masih belum bisa menjawab faktor-faktor di luar proyek yang berkaitan dengan sifat *human-centricity* pada pelaku proyek.

Di lain sisi, *project management* dalam tim yang *agile* memerlukan identifikasi terhadap faktor manusia dan metode manajemen yang tepat yang akan dipertimbangkan agar tim *agile* menjadi tim yang efektif. Berdasarkan penelitian yang pernah dilakukan dengan menganalisis metodologi manajemen proyek *Agile*, diuji dalam praktik dengan digunakan perhitungan *defect rate* dan *rework level* dari kualitas *software* yang dihasilkan. Dari sisi manajemen, digunakan team *feedback* dalam bentuk kuesioner dengan skala Likert. Hasilnya menunjukkan bahwa ukuran tim memiliki dampak langsung

pada kinerja tim dan komunikasi. Manajemen *Agile* sangat fokus pada orang, sedangkan Manajemen Proyek Perangkat Lunak lebih berfokus pada disiplin teknis. Hal ini menunjukkan bahwa faktor manusia sering tidak dipertimbangkan ketika mengelola tim *agile*, sehingga tim kurang dapat bekerja pada level yang optimal [6]. Penelitian lain mengidentifikasi dan mengategorikan *soft skill* yang mampu meningkatkan kesuksesan pengembangan proyek perangkat lunak (*Software Development Process/SDP*). Survei penelitian ini dilakukan dalam bentuk kuesioner yang sekaligus menjadi instrumen dalam penelitian ini. Kemudian, dilakukan analisis kuantitatif pada hasil survei dan ditemukan bahwa berbagai *soft skill* berkontribusi penting dalam meningkatkan kesuksesan proyek pengembangan perangkat lunak. Hasil penelitian menunjukkan bahwa semua orang yang terlibat dalam penelitian memiliki *soft skill* yang berbeda. Selain itu, hasil menunjukkan bahwa peran yang berbeda memiliki *soft skill* yang berbeda pula dan hal tersebut penting bagi pekerjaan yang dilakukan. Masing-masing peran tersebut dianalisis berdasarkan data yang dikumpulkan dengan teknik *probability sampling* agar didapatkan jenis-jenis *soft skill* yang harus dimiliki sesuai perannya [7].

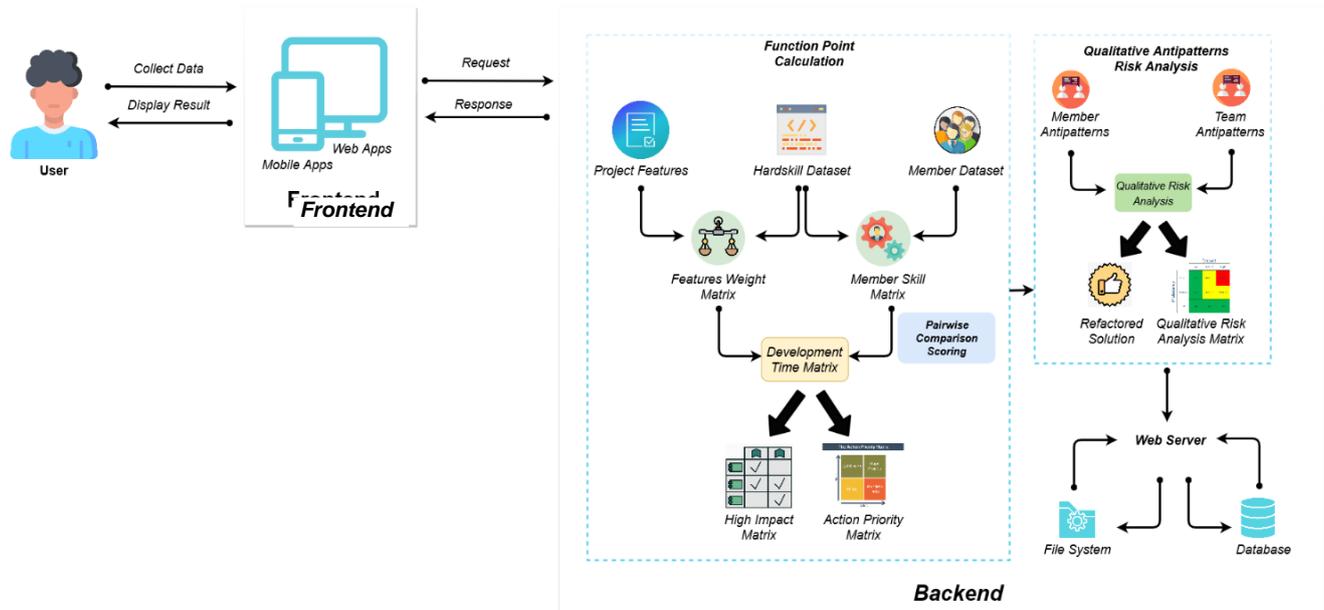
Dalam rangka mendapatkan pendekatan terbaik untuk dapat menyelesaikan permasalahan alokasi tim yang menggunakan metodologi *Agile* dalam proses pengembangan perangkat lunak, beberapa peneliti menggunakan pendekatan *hybrid* (gabungan) berdasarkan *Mamdani Fuzzy Inference System*. Sebuah penelitian menggunakan *competence matrix* untuk mendapatkan nilai kinerja *developer*. Hasil dari penelitian ini menunjukkan bahwa pendekatan yang digunakan menghasilkan derajat yang sama dengan pemetaan yang dilakukan secara manual oleh *project manager* berdasarkan hasil penghitungan derajat pada *fuzzy inference system* dan terbukti mampu meningkatkan produktivitas tim [8]. Metode alokasi *member* kepada *task* berdasarkan arsitektur perangkat lunak dan jaringan sosial dalam proyek menggunakan *Work Breakdown Structure* (WBS) untuk memetakan *member* berdasarkan arsitektur dan jaringan sosial juga sering diterapkan. Rumus digunakan untuk menghitung tingkat kecocokan karyawan dan *task* diajukan yang didasarkan pada atribut *task*, arsitektur perangkat lunak, keterampilan karyawan, dan preferensi karyawan serta hubungan jejaring sosial. Berdasarkan pengujian yang dilakukan terhadap sebuah proyek *e-business system*, didapatkan hasil pemetaan yang optimal berdasarkan tiga solusi pemetaan *member* kepada *task* yang ditawarkan oleh sistem [9].

Berdasarkan penelitian terkait yang pernah dilakukan oleh peneliti-peneliti sebelumnya, masih belum ada penelitian yang berfokus pada *mitigation plan* yang terkait dengan *people* di dalam tim pengembangan perangkat lunak. Makalah ini mengajukan suatu pemodelan baru untuk melakukan *mitigation plan* di dalam pengembangan perangkat lunak dengan pendekatan *Project Management Antipatterns*.

## II. METODOLOGI

*Project Management Antipatterns* adalah identifikasi skenario yang buruk dalam hal manajemen yang dapat merusak

## System Design



Gbr. 1 Desain sistem.

proses pengembangan perangkat lunak [10]. Mitigasi yang dihasilkan berupa pemetaan tim *developer* kepada *task* yang relevan berdasarkan *hard skill* dan *soft skill* yang dimiliki. Selain itu, mitigasi ini juga memberikan hasil berupa rekomendasi *treatment* yang dapat dilakukan terhadap tim untuk meningkatkan produktivitas serta mengurangi potensi risiko yang ditimbulkan dari hasil diagnosis praktik yang buruk. Aplikasi yang dikembangkan ini diberi nama aplikasi “GoPMGo!”. Aplikasi ini mengolah berbagai masukan dari pengguna yang terkait dengan proses pengembangan perangkat lunak, yang terdiri atas *project statement*, *dataset hard skill* dan *soft skill* dari setiap *member*, diagnosis *antipatterns*, serta daftar fitur (*product backlog*) yang akan dieksekusi disertai nilai prioritas setiap fitur yang akan dipecah menjadi *task-task* yang lebih kecil. GoPMGo! memberikan hasil berupa rekomendasi pemetaan *member* kepada *task*, disesuaikan dengan *hard skill* dan *soft skill* masing-masing *member* dan estimasi waktu penyelesaian setiap *task* oleh *member* tersebut. Selain itu, aplikasi ini juga memberikan hasil berupa rekomendasi *treatment* yang dapat dilakukan oleh *project manager* untuk menurunkan risiko kegagalan berdasarkan hasil diagnosis *antipatterns* yang diberikan. Aplikasi yang dikembangkan ini terdiri atas *frontend* dan *backend* yang masing-masing terdiri atas tahapan *function point calculation* dan *Qualitative Risk Analysis* (QRA). Desain sistem ditunjukkan pada Gbr. 1.

### A. Function Point Calculation

*Function point* mengukur dari perspektif fungsional perangkat lunak yang akan dibangun, terlepas dari bahasa pemrograman, metode pengembangan, atau *platform* perangkat keras yang digunakan. Hasil dari pengukuran menggunakan *function points* dapat digunakan untuk mengestimasi *effort* yang diperlukan dalam pengembangan perangkat lunak [7].

Setiap fitur diberikan nilai berdasarkan skala perhitungan *Complexity Adjustment Factor* (CAF).

- 0 – No influence
- 1 – Incidental
- 2 – Moderate
- 3 – Average
- 4 – Significant
- 5 – Essential

Nilai CAF disimpan sebagai nilai estimasi *effort* yang diperlukan dalam pengembangan perangkat lunak dalam satuan *function points* (*fp*) [11]. Nilai ini menggambarkan besarnya *effort* yang diperlukan untuk menyelesaikan *task* tersebut. Nilai *fp* yang dihasilkan dinyatakan sebagai *spatium* (*s*). Setiap *hard skill* fitur didefinisikan sebagai *task*. *Task* inilah yang akan dipetakan kepada *member* sebagai bentuk terkecil dari komponen dalam proyek. Bobot diberikan untuk setiap *task* dalam skala CAF.

*Function point matrix* dan *member skill matrix* akan menyusun *development time matrix*. *Development time matrix* merupakan matriks waktu yang dibutuhkan oleh setiap *member* untuk menyelesaikan *task* (*t*). Waktu dihitung dengan rumus seperti pada (1).

$$t = \frac{s}{v} \tag{1}$$

dengan

- t* = waktu (*t*)
- s* = bobot fitur (*fp*)
- v* = *skill member* (*fp/h*).

*High impact matrix* merupakan matriks yang memetakan *task* kepada *member* sesuai dengan *skill* yang dimiliki. *Member* yang mendapatkan suatu *task* merupakan *member* dengan

*development time* paling kecil di dalam *development time matrix* untuk *task* tersebut jika dibandingkan dengan *member* lainnya di dalam tim.

**B. Qualitative Antipatterns Risk Analysis Matrix**

Dalam makalah ini digunakan *root cause analysis* untuk mengategorikan *antipatterns* berdasarkan *root cause*-nya, yaitu *antipatterns member* dan *antipatterns tim*. Setiap *antipatterns* memiliki nilai yang dinyatakan dalam skala Likert (1-5). Nilai ini kemudian digunakan untuk menghasilkan QRA. Matriks ini menampilkan persebaran pemetaan *antipatterns* dalam tim berdasarkan nilai *likelihood* dan *severity*-nya. Nilai *likelihood* menunjukkan seringnya *antipatterns* tersebut muncul dari waktu ke waktu. Nilai *likelihood* dari *antipatterns* diperoleh menggunakan (2).

$$likelihood = \frac{\sum_{j=1}^N \frac{\sum_{i=1}^n a_i}{n}}{N} \quad (2)$$

dengan

- $N$  = jumlah periode (*sprint*)
- $j$  = indeks periode (*sprint*)
- $n$  = jumlah anggota tim
- $i$  = indeks anggota tim
- $a$  = nilai *antipatterns*, dalam skala Likert.

Nilai *severity* menunjukkan besar dampak yang ditimbulkan oleh *antipatterns* terhadap nilai produktivitas tim. Produktivitas tim dihitung dari persentase *task* yang selesai di setiap periode berdasarkan *burn down chart*. *Task* yang telah diselesaikan oleh tim *developer* dibandingkan dengan jumlah *task* yang seharusnya diselesaikan dalam satu *sprint*.

$$\%p = \frac{completed\ task}{total\ task} \times 100\% \quad (3)$$

dengan  $\%p$  adalah persentase produktivitas tim.

Nilai *severity* dari *antipatterns* diperoleh menggunakan (4),

$$\begin{bmatrix} s_1 \\ s_2 \\ \dots \\ s_n \end{bmatrix} = \begin{bmatrix} a_{11} & \dots & a_{1n} \\ a_{21} & \dots & a_{2n} \\ \dots & \dots & \dots \\ a_{m1} & \dots & a_{mn} \end{bmatrix}^{-1} \begin{bmatrix} p_1 \\ p_2 \\ \dots \\ p_m \end{bmatrix} \quad (4)$$

sehingga diperoleh (5).

$$[s_1 \ s_2 \ \dots \ s_n] = [p_1 \ p_2 \ \dots \ p_m] \begin{bmatrix} a_{11} & \dots & a_{1n} \\ a_{21} & \dots & a_{2n} \\ \dots & \dots & \dots \\ a_{m1} & \dots & a_{mn} \end{bmatrix}^{-1} \quad (5)$$

dengan

- $s$  = nilai *severity antipatterns* ke- $n$
- $n$  = indeks *antipatterns*
- $a$  = nilai *antipatterns* ke- $n$  pada *sprint* ke- $m$  dari jawaban *user*
- $m$  = indeks periode (*sprint*)
- $p$  = persentase produktivitas tim pada *sprint* ke- $m$ .

Dalam makalah ini, dihasilkan dua matriks. Matriks pertama merupakan persebaran *member antipatterns* dan matriks kedua merupakan persebaran tim *antipatterns*. Matriks ini memprioritaskan risiko yang diidentifikasi menggunakan

		Impact →				
		Negligible	Minor	Moderate	Significant	Severe
Likelihood ↑	Very Likely	Low Med	Medium	Med Hi	High	High
	Likely	Low	Low Med	Medium	Med Hi	High
	Possible	Low	Low Med	Medium	Med Hi	Med Hi
	Unlikely	Low	Low Med	Low Med	Medium	Med Hi
	Very Unlikely	Low	Low	Low Med	Medium	Medium

Gbr. 2 Matriks QRA.

skala penilaian. Nilai risiko ditentukan berdasarkan nilai *likelihood* dan juga *severity* dari setiap jenis kategori. Gbr. 2 menampilkan matriks QRA dengan ukuran 5x5. Matriks ini kemudian dibagi menjadi empat kuadran sebagai berikut.

1) *Minor Zone (Hijau)*: *Antipatterns* yang berada di zona ini memiliki tingkat kemunculan yang rendah. Selain itu, dampak yang diberikan terhadap tim tidak terlalu signifikan.

2) *Moderate Zone (Kuning)*: *Antipatterns* yang berada di zona ini memiliki kemungkinan kemunculan yang tinggi. Dampak yang ditimbulkan merupakan dampak minor, tetapi tetap harus segera ditangani agar tidak meningkatkan potensi kemunculan *antipatterns* yang lain.

3) *Major Zone (Oranye)*: *Antipatterns* yang berada di zona ini cukup sering muncul dan sangat berpotensi terjadi berulang di waktu mendatang. Dampak yang ditimbulkan merupakan dampak yang cukup signifikan bagi tim.

4) *Severe Zone (Merah)*: Zona ini merupakan zona yang berbahaya karena *antipatterns* yang berada di zona ini sangat sering muncul dan juga berpotensi untuk terjadi berulang jika sebelumnya sudah pernah terjadi. Dampak yang diberikan pun sangat signifikan dan memengaruhi *culture* tim.

**C. Refactored Solution**

Rekomendasi *treatment* yang dihasilkan oleh sistem diwujudkan dalam bentuk *refactored solution*. Setiap *antipatterns* memiliki beberapa *refactored solution* yang terbagi ke dalam tiga kategori, yaitu sebagai berikut.

1) *Band Aid*: *Band aid* merupakan strategi jangka pendek yang dapat diambil oleh orang-orang yang berada di sekitar *antipatterns* atau menjadi korban dari *antipatterns* itu sendiri dan tidak memiliki waktu ataupun pengaruh untuk memperbaikinya.

2) *Self Repair*: *Self repair* merupakan langkah pertama yang dapat diambil bagi seseorang atau manajer yang menjadi *antipatterns* itu sendiri. Rekomendasi ini lebih menekankan pada cara meningkatkan kualitas diri.

3) *Refactoring*: *Refactoring* merupakan rekomendasi perubahan yang sebaiknya dilakukan untuk memperbaiki

TABEL I  
TIMELINE DEVELOPMENT TIM WPPL PENS

No.	Sprint ke-	Jenis Kegiatan	Development Time		
			Sept 2019	Okt 2019	Nov 2019
1.	1	UX Design Process			
2.	2	Development Web dan Mobile Apps			
3.	3	Development Web dan Mobile Apps			

TABEL II  
TIMELINE DEVELOPMENT TIM INTERNSHIP MAULIDAN GAMES

No.	Batch	Jenis Kegiatan	Development Time		
			Jan 2020	Feb 2020	Mar 2020
1.	1	Development Web dan Mobile Apps			
		Development Web dan Mobile Apps			
		Development Web dan Mobile Apps			

situasi dari praktik buruk yang terjadi, bisa melibatkan perubahan di sisi organisasi maupun *culture*.

*Refactored solution* direkomendasikan oleh sistem berdasarkan jawaban dari kuesioner *member's antipatterns* dan *team's antipatterns* yang diisi oleh *project manager* dan *tim developer*. Kemudian, diambil nilai rata-rata dalam satu tim dari setiap *antipatterns*. *Antipatterns* dikatakan terjadi dalam suatu *sprint* jika nilai rata-rata suatu *antipatterns* lebih besar atau sama dengan 3.

### III. HASIL DAN PEMBAHASAN

#### A. Hasil Pengujian Sistem dan Pembahasan

Dalam makalah ini, dilakukan uji coba dengan menggunakan data dari Tim Mahasiswa semester 5 Politeknik Elektronika Negeri Surabaya yang menjalani mata kuliah Workshop Pengembangan Perangkat Lunak (WPPL) dan Tim *Internship* dari PT Maulidan Teknologi Kreatif (Maulidan Games). Data yang diperoleh berupa data terkait proyek yang dikembangkan, meliputi data *member*, fitur, *hard skill*, dan *antipatterns*. Data tersebut kemudian digunakan untuk membuat dan menguji coba aplikasi GoPMGo!. Waktu penyelenggaraan uji coba dari proyek akhir ini terbagi menjadi dua periode. Pada studi kasus Tim WPPL PENS, uji coba dilakukan selama tiga *sprint*, dengan *timeline* ditunjukkan pada Tabel I. Pada studi kasus Tim *Internship* Maulidan Games, uji coba dilakukan selama tiga bulan. Tabel II menjelaskan *timeline* dari pelaksanaan uji coba Tim *Internship* Maulidan Games.

Berdasarkan data dari keempat tim selama tiga *sprint*, didapatkan nilai *likelihood* dari setiap *antipatterns* untuk studi

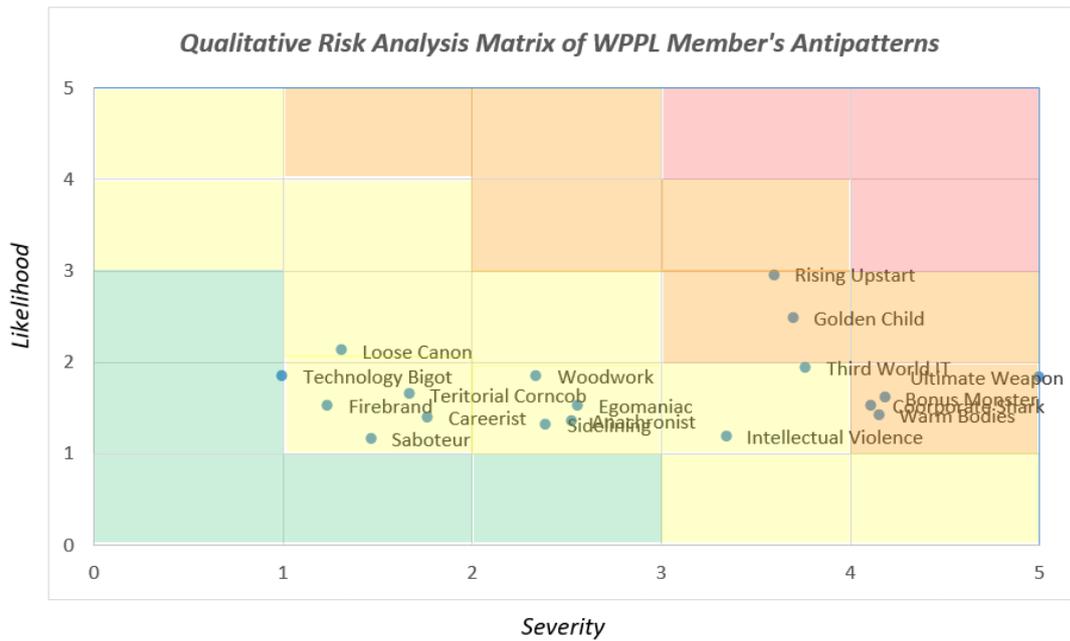
TABEL III  
PERBEDAAN KUADRAN PADA TIM ANTIPATTERNS

No.	Kode	Tim Internship Maulidan Games	Tim WPPL PENS
1.	BLJ	Moderate	Major
2.	VWE	Moderate	Major
3.	AYH	Moderate	Major
4.	CMN	Moderate	Major
5.	FRH	Moderate	Major
6.	PLH	Moderate	Major
7.	RNW	Major	Moderate
8.	BBD	Moderate	Major
9.	DVG	Moderate	Major
10.	DKS	Moderate	Major
11.	FRI	Major	Severe
12.	FDS	Moderate	Major
13.	OYK	Moderate	Minor

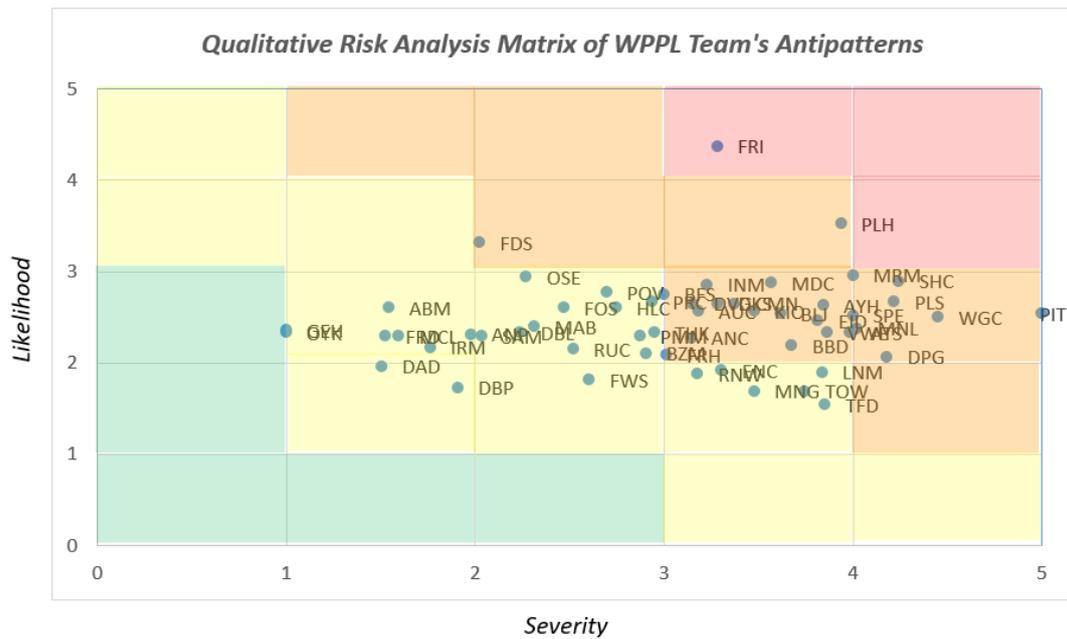
kasus ini dengan menggunakan rumus menghitung *likelihood* dan *severity*. Setelah tiga *sprint*, didapatkan nilai persebaran *antipatterns* berdasarkan nilai *likelihood* dan *severity* yang dijelaskan dalam Gbr. 3 sampai Gbr. 6. Berdasarkan QRA tersebut, dapat dilihat bahwa persebaran *antipatterns* dari kedua uji coba menghasilkan kuadran yang sama, meskipun dengan nilai *likelihood* dan *severity* yang berbeda. Kedua uji coba menghasilkan persebaran yang merata pada sumbu-x (*severity*).

Pada tim *antipatterns*, terdapat perbedaan persebaran yang diperoleh antara kedua uji coba. Perbedaan tersebut dijabarkan dalam Tabel III. Dari keseluruhan *antipatterns*, sebanyak tiga belas dari 73 *antipatterns* memberikan hasil yang berbeda pada matriks QRA. Sebelas di antaranya menjadi lebih berisiko pada studi kasus Tim WPPL PENS. Berdasarkan hasil pengamatan dan analisis yang dilakukan, hal ini terjadi karena pada studi kasus tim *internship* di Maulidan Games, pada seluruh anggota tim diberlakukan masa inisiasi, yaitu pada bulan Januari 2020. Di masa tersebut tim tidak langsung diberikan *task* yang berhubungan dengan proyek, tetapi difokuskan bagi tim *developer* untuk mengenal dan membiasakan diri dengan kultur di perusahaan. Namun, bukan berarti *task* yang diberikan bersifat "*plate spinning*" atau hanya untuk mengisi waktu, tetapi masih berkaitan dengan *project* yang akan dikerjakan di *sprint* berikutnya. Selain itu, setiap tim di dalamnya di-*assign* kepada satu orang staf yang berasal dari internal perusahaan itu sendiri. Staf ini berperan sebagai supervisor sekaligus konsultan bagi setiap tim, sehingga setiap anggota tim dipastikan dapat mengikuti dan menjalankan setiap tahapan dalam pengembangan perangkat lunak sesuai dengan yang diterapkan di perusahaan. Hal ini membuat tim menjadi lebih terarah dan termonitor dengan baik.

Pada pengujian rekomendasi *treatment*, dilakukan pengujian pada keempat Tim WPPL PENS mulai dari *sprint* pertama hingga *sprint* ketiga untuk mendeteksi *antipatterns* yang terjadi dan diberikan *treatment* yang berbeda dari satu *sprint* ke *sprint* yang lain. Pengujian pada Tim *Internship* Maulidan Games tidak dapat dilakukan secara keseluruhan karena adanya kendala pada penerapan *refactored solution* kepada tim, yaitu pada saat akan dilakukan pengujian, sedang terjadi wabah



Gbr. 3 QRA Member Antipatterns Tim WPPL PENS.

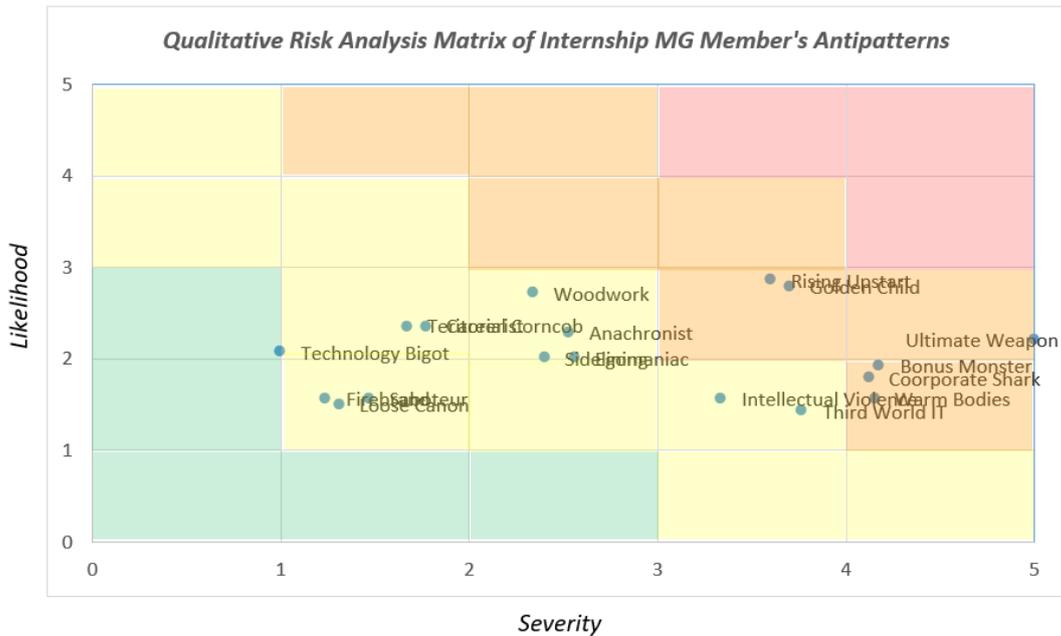


Gbr. 4 QRA Tim Antipatterns Tim WPPL PENS.

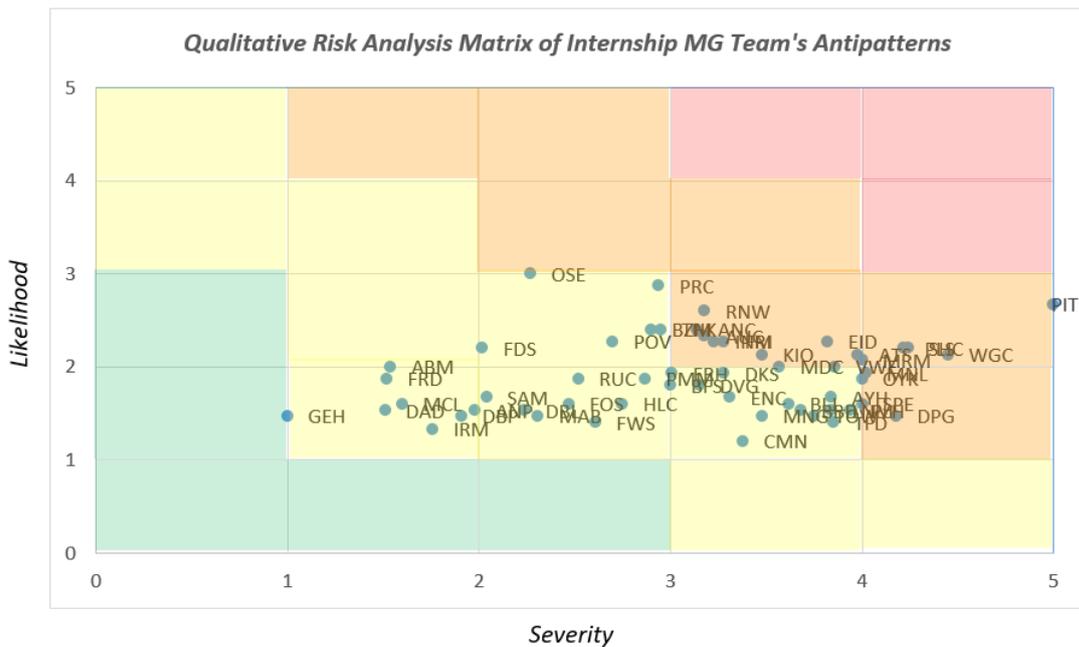
COVID-19 di tengah-tengah *timeline* yang mengharuskan seluruh tim bekerja dari rumah. Akibatnya, penerapan *refactored solution* tidak dapat dilakukan secara optimal di akhir *sprint*. Untuk menghindari ketidakakuratan hasil penelitian, pengujian ini hanya diterapkan pada Tim WPPL PENS saja.

Dalam makalah ini dijalankan skenario yang berbeda pada setiap *sprint*. Pada *sprint* pertama, ketika proyek baru saja akan dimulai, *project manager* tidak memberikan *treatment* apapun kepada tim dan membiarkan tim berjalan sendiri. Pada *sprint* kedua, *project manager* masih belum menjalankan

rekomendasi *treatment* apapun kepada timnya, meskipun sudah mengetahui *antipatterns* yang dimiliki oleh timnya berdasarkan deteksi yang dilakukan oleh sistem yang berbasis kuesioner yang wajib diisi oleh *project manager* dan juga *developer*. Memasuki *sprint* ketiga, *project manager* mulai mengimplementasikan *refactored solution* yang direkomendasikan oleh sistem kepada setiap tim. Pada setiap *sprint*, *project manager* mencatat ketercapaian *task* yang harus diselesaikan oleh *developer* menggunakan *burn down chart*. Digunakan jumlah total skor *antipatterns* yang didapatkan untuk menghitung tingkat kenaikan atau penurunan skor *antipatterns*.



Gbr. 5 QRA Member Antipatterns Tim Internship Maulidan Games.

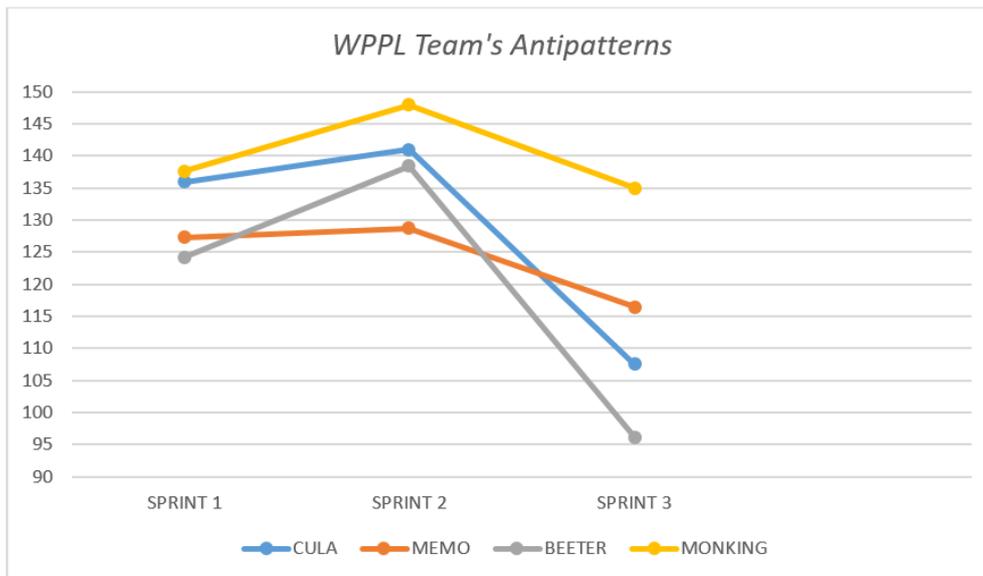
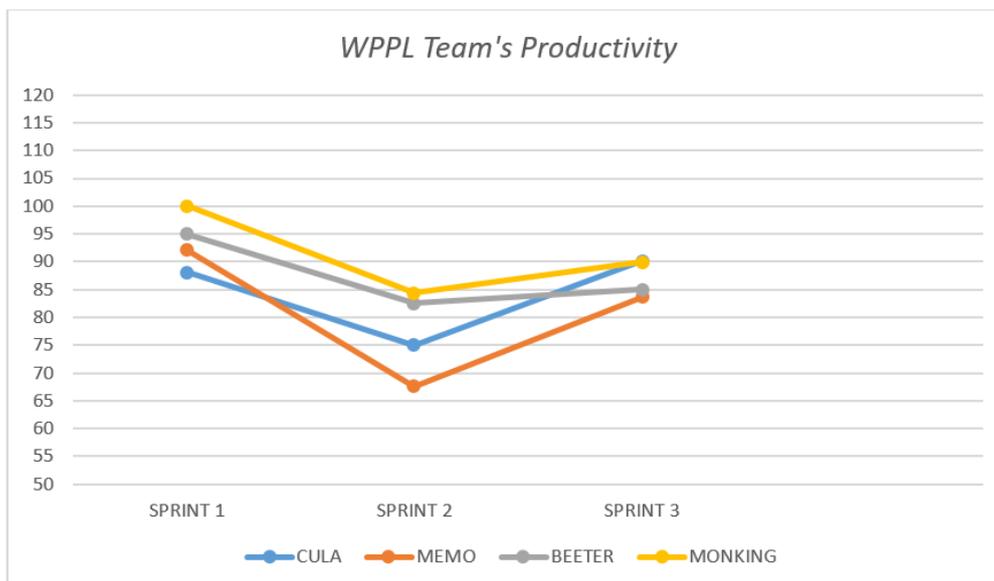


Gbr. 6 QRA Tim Antipatterns Tim Internship Maulidan Games.

Gbr. 7 menunjukkan grafik pertumbuhan *antipatterns* yang terjadi pada setiap tim di setiap *sprint*. Berdasarkan gambar tersebut, dapat dilihat bahwa tidak semua *antipatterns* yang terjadi dan di-*refactor* akan menurun atau menghilang di waktu kemudian karena hal ini sangat bergantung pada kombinasi dan keberagaman karakteristik individu di dalam tim. Selain itu, beberapa *antipatterns* juga tidak dapat diminimalkan dalam kurun waktu satu sampai tiga bulan. *Antipatterns* tersebut mungkin membutuhkan waktu bulanan, bahkan tahunan.

Pada pengujian ini, dicoba *treatment* yang berbeda pada setiap *sprint*. Dari tujuh belas *antipatterns* yang muncul di

*sprint* pertama, tiga belas di antaranya tetap muncul di *sprint* kedua. Hal ini dikarenakan tidak adanya *treatment* dalam bentuk *refactored solution* yang dijalankan oleh *project manager* di akhir *sprint* 1 menuju *sprint* 2. Berbeda dengan *antipatterns* yang muncul di *sprint* kedua, dua puluh dari 29 *antipatterns* tidak lagi muncul di akhir *sprint* ketiga. Hal ini dikarenakan adanya *treatment* yang berbeda yang diberikan oleh *project manager* secara *top down* kepada tim. *Treatment* yang berbeda ini memberikan dampak yang cukup signifikan terhadap produktivitas tim. Gbr. 8 menampilkan grafik produktivitas tim WPPL.

Gbr. 7 Grafik pertumbuhan *antipatterns*.Gbr. 8 Grafik pertumbuhan *antipatterns*.

Berdasarkan grafik, diperoleh hasil bahwa semua tim memiliki pola yang sama, yaitu nilai setiap *antipatterns* meningkat dari *sprint* 1 menuju *sprint* 2, kemudian menurun dari *sprint* 2 ke *sprint* 3. Hal ini berbanding terbalik dengan produktivitas tim, yaitu produktivitas tim menurun dari *sprint* 1 ke *sprint* 2, tetapi meningkat dari *sprint* 2 menuju *sprint* 3. Gbr. 8 menunjukkan produktivitas tim dari *sprint* 1 hingga *sprint* 3 berdasarkan *burn down chart*.

Sebanyak 83% dari anggota tim, terdiri atas *product owner*, *developer*, dan *UI/UX designer*, setuju bahwa *refactoring* yang dilakukan mampu meminimalkan nilai *antipatterns* dan dapat meningkatkan keseluruhan nilai produktivitas tim.

Berdasarkan hasil yang diperoleh, metode yang digunakan terbukti mampu menjawab permasalahan yang terjadi di dalam tim pengembangan perangkat lunak, yaitu *irrational*

*management*. Metode yang digunakan mampu menghasilkan pemetaan *task* kepada *member* secara relevan, sehingga menjadikan *task* yang akan diselesaikan dapat terencana sesuai estimasi waktu yang dihasilkan secara empiris dan tidak mengandalkan insting. Metode ini cukup membantu mengurangi beban kerja seorang *project manager* sehingga tidak menghambat proses *development*. Selain itu, metode ini mampu memberikan metode yang jelas dalam membagi *task* kepada *developer* dalam situasi terbaik (*best case*). Tampilan aplikasi web dan *mobile* dari aplikasi GoPMGo! ditunjukkan pada Gbr. L1 dan Gbr. L2 dalam halaman lampiran.

#### IV. KESIMPULAN

Makalah ini berfokus pada penerapan *Project Management Antipatterns* dalam aplikasi *people management* di ranah

pengembangan perangkat lunak. Faktor manusia di dalam *software development process* adalah salah satu penentu keberhasilan sebuah proyek. *Mitigation plan* diperlukan untuk membantu *project manager* dalam mengelola sumber daya manusia di dalam tim. Makalah ini berfokus pada tim pengembangan perangkat lunak yang mengimplementasikan *top down management* dalam *software development process* dan telah diuji kepada empat tim dari kalangan mahasiswa yang masing-masing terdiri atas tujuh orang dan empat tim dari lingkungan industri. Rekomendasi *treatment* yang diberikan dalam bentuk *refactoring* tidak melibatkan faktor lainnya yang juga memengaruhi proses pengembangan perangkat lunak, seperti faktor kesehatan, usia, posisi, atau jabatan, dan sebagainya. Pemberian *treatment* dalam bentuk *refactoring* terhadap *antipatterns* terbukti memengaruhi produktivitas tim secara keseluruhan, sebesar 83%. Akan tetapi, tidak semua *antipatterns* dapat diminimalkan, atau bahkan dihilangkan, dalam beberapa bulan saja. *Member antipatterns* yang terjadi karena karakteristik dari individu tertentu atau bahkan karakter dari *project manager* itu sendiri tidak dapat secara optimal diminimalkan dalam kurun waktu tiga sprint (tiga bulan). Hal ini sangat bergantung pada kombinasi dari karakteristik dan keberagaman kepribadian dari semua pihak yang terlibat dan *software development process*. Pemetaan yang dilakukan memberikan hasil bernilai positif dengan pengujian yang dilakukan menggunakan *net promoter score*, yaitu sebesar 38,46 dan 61,54. Semua tim pada akhirnya memiliki pola yang sama, yaitu tingginya nilai *antipatterns* berbanding terbalik dengan tingkat produktivitas tim secara keseluruhan. Selain itu, *refactored solution* yang diberikan kepada tim sesuai dengan hasil diagnosis *antipatterns* terbukti sangat memengaruhi penurunan nilai dari *antipatterns* tersebut di periode/*sprint* berikutnya.

#### UCAPAN TERIMA KASIH

Terima kasih disampaikan kepada segenap civitas akademika PENS yang turut memberikan dukungan dalam

proses penelitian ini, terutama Tim WPPL Kelas 3 D4 Teknik Informatika A 2017. Ucapan terima kasih juga ditujukan kepada PT Maulidan Teknologi Kreatif yang banyak membantu dan mendukung penyelesaian penelitian ini, baik secara finansial maupun nonfinansial.

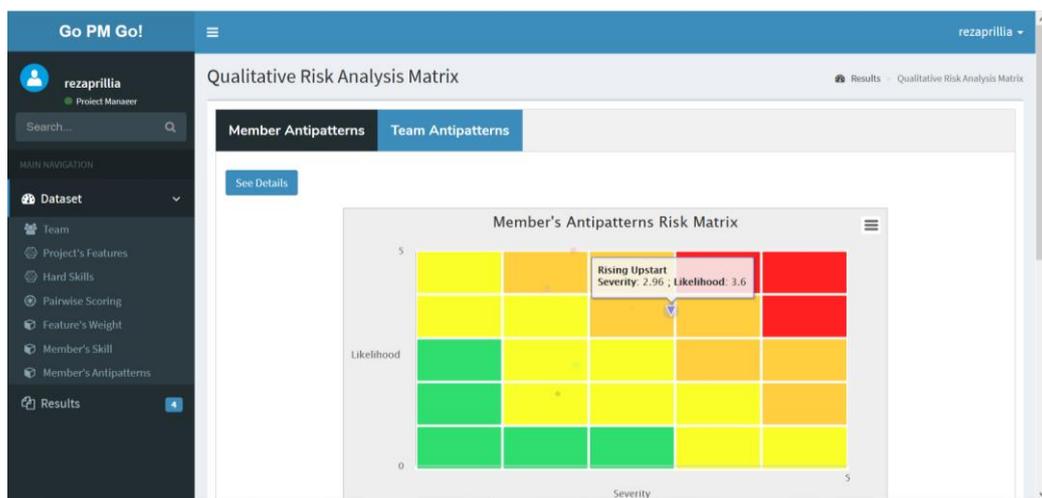
#### REFERENSI

- [1] "The Future of Jobs: Employment, Skills and Workforce Strategy for the Fourth Industrial Revolution," World Economic Forum, Geneva, 2016.
- [2] R. Schmidt, K. Lyytinen, M. Keil, dan P. Cule, "Identifying Software Project Risks: An International Delphi Study," *Journal of Management Information Systems*, Vol. 17, No. 4, hal. 5-36, Jan. 2015.
- [3] W.J. Brown, R.C. Malveau, H.W. McCormick III, dan T.J. Mowbray, *Antipatterns Refactoring Software, Architectures, and Projects in Crisis*, Hoboken, USA: Wiley, 1998.
- [4] N. Gorla dan Y.W. Lam, "Who Should Work with Whom?: Building Effective Software Project Teams," *Communications of the ACM*, Vol. 47, No. 6, hal. 79-82, Jun. 2004.
- [5] D. Kang, J. Jung, dan D.-H. Bae, "Constraint-based Human Resource Allocation in Software Projects," *Software: Practice and Experience*, Vol. 41, No. 5, hal. 551-577. 2011.
- [6] V. Lalsing, S. Kishnah, dan S. Pudaruth, "People Factors in Agile Software Development and Project Management," *International Journal of Software Engineering & Applications (IJSEA)*, Vol. 3, No. 1, hal. 117-137, 2012.
- [7] E.S. Mtsweni, T. Hörne, dan J.A. van der Poll, "Soft Skills for Software Project Team Members," *International Journal of Computer Theory and Engineering*, Vol. 8, No. 2, hal. 150-155, Apr. 2016.
- [8] R. Britto, P.S. Neto, R. Rabelo, W.A. dan T. Soares, "A Hybrid Approach to Solve the Agile Team Allocation Problem," dalam *IEEE World Congress on Computational Intelligence*, 2012, hal. 10-15.
- [9] L. Zhou, "A Project Human Resource Allocation Method Based on Software Architecture and Social Network," dalam *2008 4th International Conference on Wireless Communications, Networking and Mobile Computing*, 2008, hal. 1-6.
- [10] P.A. Laplante dan C.J. Neill, *Antipatterns: Identification, Refactoring and Management*, Boca Raton, USA: CRC Press, 2006.
- [11] R.S. Pressman, *Software Engineering: A Practitioner's Approach*, New York, USA: McGraw-Hill, 2010.

## LAMPIRAN

No.	Feature's Name	Hardskill Type	Will taken by
1	Add Account	Backend	Faiz
2	Add Account	Frontend	Citra
3	Add Account	UI/UX Design	Luluk
4	Transaction	Backend	Faiz
5	Transaction	Frontend	Citra
6	Transaction	UI/UX Design	Luluk
7	Profit and Cost Center	Backend	Faiz
8	Profit and Cost Center	Frontend	Citra
9	Profit and Cost Center	UI/UX Design	Luluk
10	Report	Backend	Faiz
11	Report	Frontend	Citra

(a)



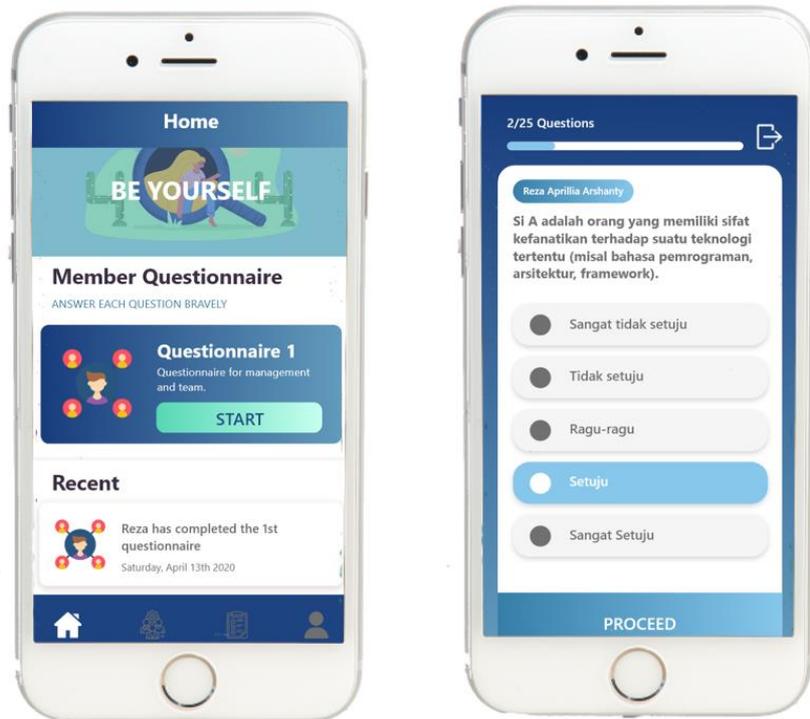
(b)

Description	Band Aid	Self Repair	Refactoring
<b>Corporate Shark</b> Likelihood: 1.52 Severity: 4.12 Corporate Shark adalah orang yang mendapatkan posisinya saat ini bukan karena keahlian teknisnya, melainkan karena keahliannya untuk mengelola hubungan dengan orang lain (pandai dalam menjalin relasi dan bermain dengan sistem).			
<b>Bonus Monster</b> Likelihood: 1.6 Severity: 4.18 Bonus Monster adalah orang yang senang mengejar keuntungan jangka pendek (seperti bonus insentif) dengan mengambil jalan pintas dalam proses pengembangan perangkat lunak, bersifat merusak dan mendorong persaingan internal.			
<b>Egomaniac</b> Likelihood: 1.52 Severity: 2.56 Egomaniac adalah orang yang suka dianggap dominan (narcistic) dan sangat menjaga citranya di lingkungannya.			
<b>Loose Canon</b> Likelihood: 2.13 Severity: 1.31 Loose Canon adalah orang yang sulit ditangani karena ke-extrovertannya yang merusak citra atau moral tim, misalnya menyebarkan informasi buruk tentang tim (blabber mouth).			

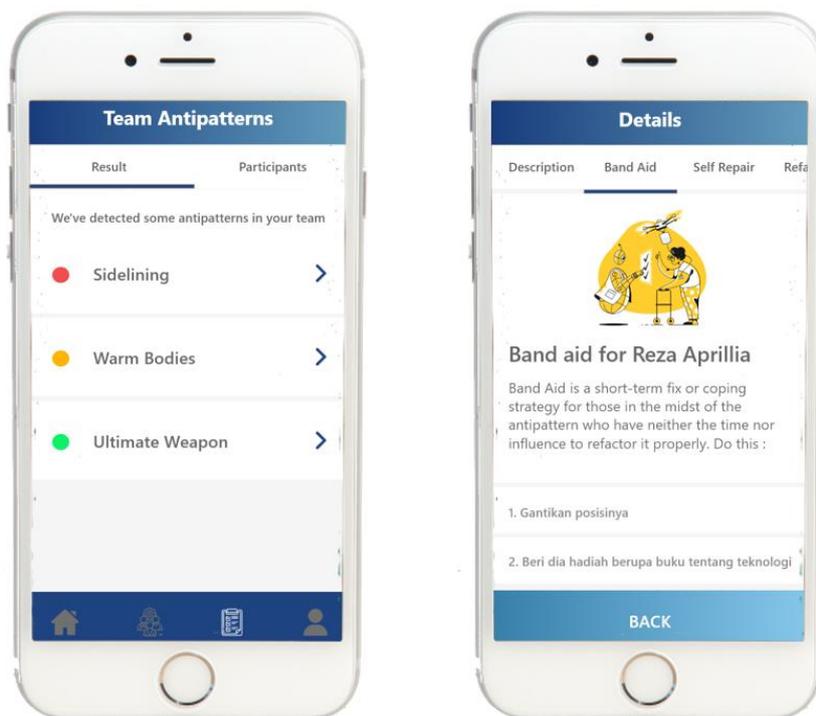
(c)

Gbr. L1 Aplikasi web GoPMGo!: (a) halaman *High Impact Matrix*; (b) halaman *Qualitative Risk Antipatterns Analysis*; (c) halaman Rekomendasi *Treatment (refactored solution)*.

LAMPIRAN



(a)



(b)

Gbr. L2 Aplikasi *mobile* GoPMGo! (Android); (a) halaman utama dan Kuesioner *Antipatterns*; (b) halaman Hasil Deteksi *Antipatterns* dan *Refactored Solution*.