



## PERAN STRATEGIS SCRUM MASTER DALAM PENGEMBANGAN PERANGKAT LUNAK *INVENTORY* PT. WAHANA *IT SERVICES* BERBASIS ANDROID

Sofyan Mufti Prasetyo<sup>1</sup>, Ahmad Jacky<sup>2</sup>, Diky Hernadi<sup>3</sup>, Ramdan Dwi Cahyo<sup>4</sup>, Very  
Novan Azhari<sup>5</sup>

<sup>1-5</sup> Teknik Informatika, Universitas Pamulang, Tangerang Selatan, Indonesia  
E-mail: <sup>1</sup>[dosen01809@unpam.ac.id](mailto:dosen01809@unpam.ac.id), <sup>2</sup>[aazackyabiyyan@gmail.com](mailto:aazackyabiyyan@gmail.com), <sup>3</sup>[Dikyhernadi05@gmail.com](mailto:Dikyhernadi05@gmail.com),  
<sup>4</sup>[Ramdandwicahyo@gmail.com](mailto:Ramdandwicahyo@gmail.com), <sup>5</sup>[veravesana17@gmail.com](mailto:veravesana17@gmail.com)

**Abstrak-** Aplikasi berbasis seluler semakin banyak digunakan, terutama dalam pengaturan literasi dan pekerjaan, karena kebutuhan akan fleksibilitas dan akses cepat. Studi ini berencana untuk menganalisis penggunaan teknik Scrum dalam kemajuan pemrograman portabel dengan melihat pekerjaan penting dari Scrum Master. Scrum adalah metode untuk mengembangkan aplikasi yang dinamis dan mudah beradaptasi. Tahapan dalam Scrum terdiri dari periode pemeriksaan prasyarat kerangka kerja, perencanaan kerangka kerja, pengujian kerangka kerja, dan pengakuan item. Wawancara digunakan sebagai metode pengumpulan data dalam pendekatan penelitian kualitatif. Setelah itu, hasil wawancara direduksi, literatur dianalisis, dan kajian dipresentasikan dan diakhiri. Penelitian dilakukan di PT. Wahana IT Service sebagai lokasi penerapan aplikasi Inventory berbasis Android. Menurut temuan penelitian ini, peran strategis dari Scrum Master termasuk mengelola ekspektasi peran, mengubah resistensi, mematuhi jadwal, menangani permintaan perubahan yang mendesak, dan mendistribusikan anggota tim. Pemilihan Scrum Master didasarkan pada peran strategis ini, yang harus dipertimbangkan dan berkualitas berdasarkan poin-poin tersebut.

**Kata Kunci:** Dinamis, Pengembangan Aplikasi, Scrum Master

**Abstract-** Mobile-based apps are increasingly being used, especially in literacy and work settings, due to the need for flexibility and fast access. This study plans to analyze the use of Scrum techniques in the progress of portable programming by looking at the important work of Scrum Masters. Scrum is a method for developing dynamic and adaptable applications. The stages in Scrum consist of periods of framework prerequisite checking, framework planning, framework testing, and item recognition. Interviews are used as a data collection method in a qualitative research approach. After that, the results of the interviews were reduced, the literature was analyzed, and the study was presented and concluded. The research was conducted at PT. Wahana IT Service as the location for implementing the Android-based Inventory application. According to the findings of this study, the strategic roles of the Scrum Master include managing role expectations, changing resistance, adhering to schedules, handling urgent change requests, and distributing team members. The selection of the Scrum Master is based on this strategic role, which must be considered and qualified based on these points.

**Keywords:** Dynamic, Application Development, Scrum Master

### 1. PENDAHULUAN

Perkembangan perangkat lunak untuk teknologi informasi telah menjadikan era baru dalam kehidupan manusia. Hal ini menyebabkan perangkat lunak menjadi salah satu bagian dari peradaban manusia modern [1]. Pressman mengemukakan bahwa perangkat lunak menjadi alat bantu dalam pengambilan keputusan dalam semua bidang pada masa kini seperti bisnis, militer, dan pekerjaan. Selain itu perangkat lunak juga berfungsi sebagai basis layanan yang telah dibuat dalam bentuk sistem

informasi untuk mendukung proses bisnis seperti sistem transportasi, medis, telekomunikasi, industri, dan hiburan [2].

Salah satu penggunaan teknologi informasi pada PT. Wahana IT Service adalah Inventory. Inventory adalah Barang yang dikelola oleh perusahaan dengan tujuan untuk di jual [3]. Inventory dapat berupa bahan mentah yang dibeli dan diubah menjadi sesuatu yang sama sekali baru, selain itu dapat berupa produk masal yang diuraikan menjadi bagian bagian penyusunnya dan jual secara terpisah[4]. Peran Inventory dapat mengontrol stok barang yang mana hal ini merupakan Langkah awal untuk membuat perencanaan Panjang soal kebutuhan barang setiap bulannya [5]. Pengembangan aplikasi atau perangkat lunak seperti Aplikasi inventori, membutuhkan metode pengembangan aplikasi. Metode ini sangat penting dalam proses pembuatan aplikasi untuk memberikan pegangan bagi pengembang aplikasi untuk meluncurkan aplikasi hingga aplikasi siap. Salah satu cara untuk mengembangkan perangkat lunak adalah dengan menggunakan framework Scrum [6], [7]

Scrum adalah metode pengembangan perangkat lunak yang iteratif dan inkremental [8]. Metode ini pertama kali dipresentasikan pada tahun 1997 oleh Ken Schwaber dalam sebuah artikel yang menjelaskan tentang metodologi Scrum pada tahun 1995 [9]. Alur kerja Scrum adalah serangkaian iterasi, yang disebut sprint, yang berlangsung antara satu hingga empat minggu [10]. Kelompok setuju dengan dasar pekerjaan sebagai bagian dari basis produk, yang merupakan daftar persyaratan dan prioritas [11]. Dalam setiap sprint ada pertemuan harian dengan masing-masing anggota tim mempresentasikan apa yang telah dilakukan pada hari sebelumnya, apa yang perlu dilakukan pada hari berikutnya dan jika ada kendala untuk kelanjutan kegiatan pengembangan [12]. Pada akhir setiap Sprint terdapat Product Demo atau Sprint Review, dan setelah setiap Sprint Review terdapat Learning Session atau Sprint Retrospective [13].

Hermanto et al., (2018) melakukan tinjauan dan analisis metodologi Scrum dalam konteks pengembangan seluler dan menemukan bahwa pendekatan Scrum efektif dalam pengembangan aplikasi seluler. Penggunaan metode Scrum pada aplikasi seluler dapat dikembangkan dalam waktu singkat. Hal ini menjadikan metode Scrum menjadi alasan utama keberhasilan proyek ketika waktu terbatas dan di sepanjang proyek konten pengembangan aplikasi seluler dapat dipelajari tepat waktu [15]. Keberhasilan penggunaan metode Scrum sangat ditentukan oleh Scrum Master [16]. Scrum Master bertanggung jawab untuk membuat Scrum sesuai dengan Pedoman Scrum yang disepakati oleh tim dan pemilik proyek. Scrum Master juga berperan dalam membantu setiap orang memahami teori dan praktik Scrum dalam tim dan organisasi Scrum [17]. Scrum Master bertanggung jawab atas efektivitas Tim Scrum. Scrum Master melakukan ini untuk memungkinkan Tim Scrum meningkatkan praktik dan kinerjanya selama proses aplikasi. Dengan kata lain, Scrum Master adalah pemimpin sejati yang melayani Tim Scrum dan tim yang lebih besar [18]. Tanggung jawab Scrum Master untuk pengembangan aplikasi menggunakan praktik Scrum dalam konteks pengembangan perangkat lunak Android memastikan bahwa proses Scrum efektif dalam skenario pengembangan proyek di mana konsep sistem sering berubah dengan cepat [16], [19].

Berdasarkan penjelasan di atas, Scrum Master merupakan kunci keberhasilan pengembangan sistem. Namun ketika menerapkan metode Scrum, Scrum Master memiliki kendala tersendiri dalam menerapkan metode tersebut. Tujuan dari penelitian ini adalah untuk mengetahui apa saja yang ditemui oleh pengembang sistem saat menggunakan metode Scrum saat mengembangkan aplikasi seluler Android. Dengan adanya penelitian ini diharapkan dapat menjadi dasar bagi pengembang sistem dalam menggunakan Scrum untuk membuat perangkat lunak yang baik dan memenuhi keinginan pengguna.

## 2. METODE

### 2.1. Jenis Penelitian

Penelitian ini menggunakan metode kualitatif. Metode penelitian kualitatif menekankan metode penelitian observasional dan interaktif di lapangan (wawancara mendalam), dimana data dianalisis dengan cara non statistik [25]. Penelitian ini mengumpulkan data dengan menggunakan teknik wawancara dengan informan survei yaitu Scrum Masters. Analisis data penelitian terdiri dari tiga tahapan yaitu ringkasan, penyajian dan kesimpulan.

### 2.2. Tinjauan Pustaka

#### a. Metode *Agile*

Pengembangan perangkat lunak *Agile* adalah salah satu metode pengembangan perangkat lunak. Kata *Agile* berarti cepat, ringan, bebas bergerak, dan gesit. *Agile* adalah istilah yang digunakan untuk menggambarkan konsep model proses yang berbeda dari konsep model proses yang ada [20]. Konsep pengembangan perangkat lunak *Agile* yang dicetuskan oleh Kent Beck dan 16 rekannya menyatakan bahwa *Agile Software Development* adalah cara untuk membangun perangkat lunak dengan cara melakukannya dan membantu orang lain untuk membangun semuanya sekaligus [9], [21].

Dalam Pengembangan Perangkat Lunak *Agile* interaksi dan personel lebih penting daripada proses dan alat, perangkat lunak yang berfungsi lebih penting daripada dokumentasi lengkap, kolaborasi dengan klien lebih penting daripada negosiasi kontrak, dan responsif terhadap perubahan lebih penting daripada mengikuti rencananya [13]. Namun, seperti model proses lainnya, *Agile Software Development* memiliki kelebihan sendiri dan tidak cocok untuk semua jenis proyek, produk, orang, dan situasi. *Agile Software Development* memungkinkan model proses yang toleran terhadap perubahan kebutuhan sehingga respon terhadap perubahan dapat dilakukan lebih cepat [19].

#### b. *Scrum*

*Scrum* dipresentasikan pada tahun 1995 oleh Ken Schwaber pada konferensi OOPSLA di Austin, tetapi dasar-dasar *Scrum* muncul pada tahun 80-an [9]. Menurut [20], kunci sukses *Scrum* diturunkan dari pengalaman dan menciptakan bagian penting dari pengetahuan di perusahaan. *Scrum* didasarkan pada empirisme yang dicapai dengan beberapa kualitas. Literatur yang disurvei, kualitas yang membangun empirisme di *Scrum* adalah kejelasan dalam setiap proses, inspeksi untuk mendeteksi masalah dalam proyek dan adaptasi terhadap perubahan [19]. *Scrum* terdiri dari tiga elemen utama yang disajikan di bawah ini [20]:

##### 1. *Roles*

Dalam peran (*roles*) ada empat peran utama. Pertama adalah Tim *Scrum* yang terdiri dari *Scrum Master*, Tim Pengembang dan Pemilik Produk. Kedua adalah *Scrum Master*, yaitu orang yang bertanggung jawab untuk memahami dan menggunakan nilai dan aturan *Scrum* oleh Tim Pengembang dan Pemilik Produk. Tugas utama *Scrum Master* adalah melayani Tim *Scrum* untuk mencapai tujuan proyek. *Scrum Master* bukanlah manajer tradisional namun lebih seperti konselor yang membangun hubungan dengan anggota tim berdasarkan kepercayaan dan kerjasama [22]. Ketiga adalah Pemilik Produk, yaitu orang yang mengetahui kasus bisnis yang terkait dengan proyek. Tugas utama Product Owner adalah mengontrol Product Backlog. Pemilik Produk harus memiliki kekuatan pengambilan keputusan untuk mengambil keputusan atas nama perusahaan [23]. Keempat adalah Tim

Pengembang yang mengembangkan produk sesuai dengan kebutuhan. Development Team adalah 'self-organizing', artinya memiliki tingkat otonomi yang tinggi.

2. *Artifacts*

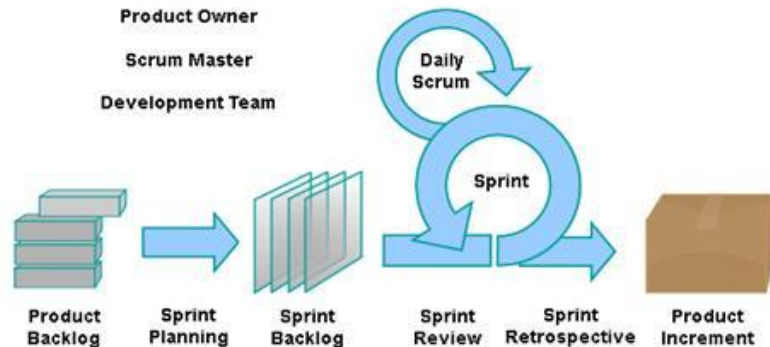
Artefak (*artifacts*) adalah hasil material atau non-material dari suatu pengembangan perangkat lunak. Dalam beberapa literatur, pada artefak ada beberapa hal yang perlu dipertimbangkan. Pertama adalah Product Backlog yang menentukan ruang lingkup dan daftar fitur yang harus dilaksanakan selama proyek berlangsung. Elemen dalam Product Backlog disusun secara hierarkis dan setiap tugas memiliki deskripsi, nomor seri, dan nilai perkiraan (kompleksitas tugas) serta prioritas. Product Backlog dapat dimodifikasi selama proyek berlangsung. Kedua adalah Sprint Backlog, yaitu bagian dari Product Backlog yang dipilih selama perencanaan Sprint menunjuk semua tindakan yang harus dilakukan untuk mencapai Sprint. Ketiga adalah Tujuan Sprint sebagai elemen penting untuk membuka jalan bagi Tim Pengembang dan Scrum Master. Keempat adalah Increment yang menentukan total komponen dari Product Registry yang diselesaikan selama Sprint dan Sprints di masa lalu. Kelima adalah Definisi Selesai yang fokus pada kejelasan pemahaman ketika elemen dari Product Backlog dapat diterima sebagai selesai. Definisi tersebut harus diakui dan dipahami oleh semua anggota tim.

3. *Events*

Event adalah hal yang penting untuk memberikan keteraturan di Scrum. Event Scrum terdiri dari beberapa bagian. Pertama adalah Sprint, yaitu kotak waktu satu bulan atau kurang di mana Increment produk 'Selesai', dapat digunakan, dan berpotensi dirilis dibuat [23]. Sprint terdiri dari Sprint Planning, Sprint Review, dan Sprint Retrospective. Kedua adalah Perencanaan Sprint. Tujuan utama dari event ini adalah menyiapkan ruang lingkup pekerjaan yang harus dilakukan selama iterasi. Selama Sprint Planning, item dari Product Backlog dipilih untuk diimplementasikan dan memastikan terciptanya peningkatan produk. Ketiga adalah Ulasan Sprint, yaitu event di akhir setiap Sprint yang digunakan untuk memeriksa fungsionalitas yang diberikan. Pertemuan tersebut berorientasi pada pemeriksaan dan penyempurnaan Product Backlog. Keempat adalah Retrospektif Sprint yang berkonsentrasi pada refleksi dari Sprint yang telah berakhir. Tim Pengembang dan Scrum Master fokus pada pemeriksaan tugas yang sudah selesai dan rencana perbaikan untuk sprint berikutnya.

c. *Scrum Flow*

Sebuah proyek dengan metode Scrum diawali dengan penggambaran sistem yang akan dikerjakan, kemudian Pemilik Proyek menggambarkan proses bisnis atau rencana kedalam Product Backlog [9]. Product Backlog adalah daftar rencana yang harus dikerjakan oleh tim. Ada istilah di Scrum yang disebut Sprint. Sprint adalah tujuan yang ingin dicapai dalam Scrum sprint berikutnya (30 hari kedepan). Setiap Sprint diawali dengan Sprint Meeting Planning yang merupakan kegiatan untuk menentukan jenis sprint yang akan dilakukan selanjutnya. Setiap hari, setiap tim berkumpul dan berdiskusi tentang “Apa yang telah dilakukan sejak Rapat Scrum Harian terakhir?”, “Masalah apa yang dihadapi selama pengerjaan?”, dan “Apa yang akan dilakukan untuk Scrum selanjutnya?” [16]. Rapat dipimpin oleh Scrum Master dan di akhir sprint akan ada rapat selama 4 jam untuk melakukan demo terhadap semua yang telah dilakukan. **Gambar 1** merupakan alur Scrum.



**Gambar 1.** Alur Scrum

d. *Software*

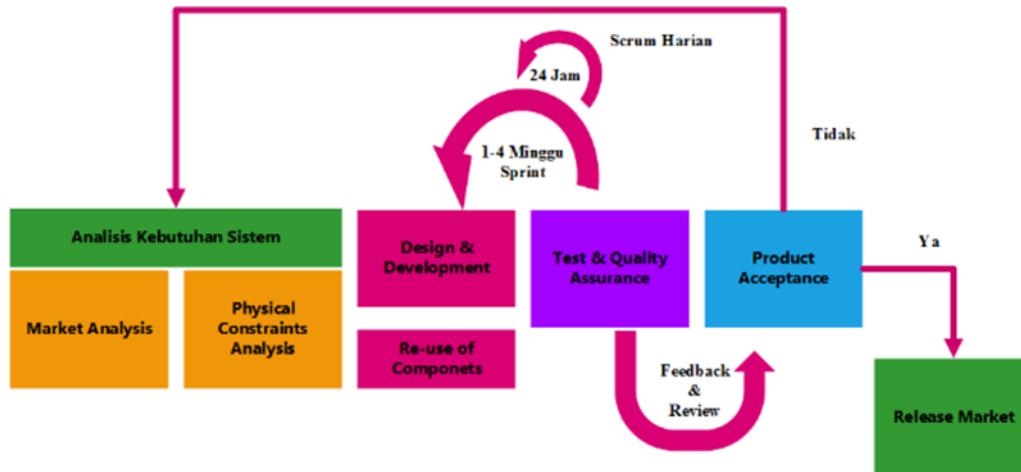
Perangkat lunak (*software*) merupakan program komputer yang berfungsi sebagai sarana interaksi antara pengguna dan hardware. Perangkat lunak juga dapat dikatakan sebagai "penerjemah" dari perintah-perintah yang dijalankan oleh pengguna komputer untuk diteruskan atau diproses oleh perangkat keras [24].

### 3. ANALISA DAN PEMBAHASAN

Pada bagian ini dibahas hasil penelitian yang telah dilakukan, yaitu aplikasi yang telah dibuat dan kendala selama menerapkan metode *Scrum* untuk membangun aplikasi Inventory PT. Wahana *IT Service*.

#### 3.1. Penerapan Metode *Scrum* dalam Pengembangan *Inventory*

Metode pengembangan sistem dengan *Scrum* menyarankan agar proyek berkembang melalui serangkaian sprint [26]. Metode pengembangan perangkat lunak dengan *Scrum* pada aplikasi *Inventory* PT. Wahana *IT Service* dilakukan dengan menerapkan beberapa fase seperti ditunjukkan pada **Gambar 2**.



**Gambar 2.** Tahapan penerapan metode Scrum dalam pengembangan perangkat lunak

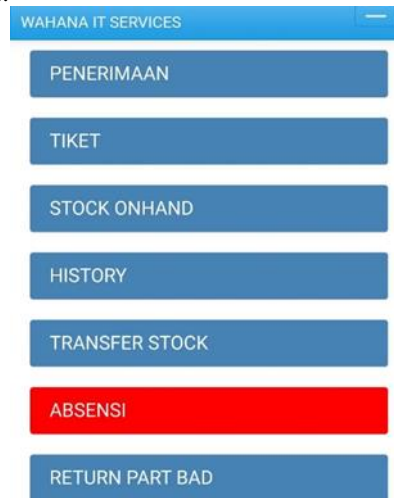
**Gambar 2** menunjukkan penerapan metode Scrum memerlukan beberapa fase yang dijelaskan sebagai berikut.

- a. **Analisis Kebutuhan Sistem**  
 Pada fase ini, Scrum Master dan owner aplikasi melakukan pertemuan untuk membahas mengenai kebutuhan dari market, kebutuhan sistem, dan kebutuhan penerapan sistem. Tahap ini mendapatkan gambaran dengan jelas tentang apa yang dikerjakan pada analisa sistem kemudian dilanjutkan dengan memikirkan bagaimana membentuk sistem tersebut. Layanan, batasan, dan tujuan dari sistem ditetapkan melalui konsultasi dengan pengguna sistem. Semua itu didefinisikan secara detail dan bertindak sebagai spesifikasi sistem.
- b. **Desain sistem dan *software (Design)***  
 Proses desain sistem membagi kebutuhan menjadi hardware atau software. Hal ini menetapkan hampir seluruh perancangan sistem. Desain software melibatkan identifikasi dan penggambaran mengenai pemisahan dasar *system software* dan hubungannya. Untuk dapat dimengerti oleh mesin, dalam hal ini adalah komputer, maka desain tadi harus diubah menjadi bentuk yang dapat dimengerti oleh mesin, yaitu kedalam bahasa pemrograman melalui proses *coding*. Tahap ini merupakan implementasi dari tahap design yang secara teknis nantinya dikerjakan oleh *programmer*.
- c. ***Test & Quality Assurance***  
 Satuan program atau kumpulan program diintegrasikan dan diuji sebagai sistem yang telah selesai untuk menjamin bahwa kebutuhan *software* telah terpenuhi. Setelah pengujian, *system software* dikirimkan kepada pelanggan.
- d. ***Product Acceptance***  
 Biasanya ini adalah bagian siklus hidup software yang paling lama. Sistem di-install dan dimasukkan dalam penggunaan. Pemeliharaan melibatkan pembenaran kesalahan yang tidak ditemui dalam tahap awal siklus, meningkatkan implementasi satuan sistem, dan meningkatkan layanan sistem sehubungan ditemukannya kebutuhan baru. Sesuai dengan metodologi agile, sprint dibatasi waktu tidak lebih dari satu bulan, paling sering dua minggu. **Gambar 3** merupakan alur penerapan Scrum dalam pengembangan sistem.



**Gambar 3.** Penerapan *Scrum* dalam pengembangan sistem

Metodologi Scrum mengharuskan tim pengembangan sistem untuk rapat membahas perencanaan di awal sprint. Anggota tim mencari tahu berapa banyak item yang dapat dijanjikan dan kemudian membuat sprint backlog, yaitu daftar tugas yang harus dilakukan selama sprint [27]. Inventory yang dibuat ini terdiri dari Tujuh menu utama yang disepakati dengan Product Owner, yaitu List Penerimaan, Tiket, Stock onhand , History, Transfer Stock , Absensi, Return Part Bad, seperti ditampilkan pada **Gambar 4**.



**Gambar 4.** Menu Utama Wahana *IT Service*

Selama pengembangan Inventory , proses sprint Scrum yang cepat memerlukan tim Scrum mengambil sekumpulan kecil fitur dari ide hingga fungsionalitas yang dikodekan dan diuji. Pada akhirnya, fitur-fitur ini selesai, artinya dikodekan, diuji, dan diintegrasikan kedalam produk atau sistem yang berkembang. Pada setiap hari sprint, semua anggota tim harus menghadiri pertemuan Scrum harian, termasuk ScrumMaster dan pemilik produk. Rapat ini memiliki batas waktu tidak lebih dari 15 menit. Selama waktu itu, anggota tim membagikan apa yang mereka kerjakan pada hari sebelumnya, akan mengerjakan apa hari itu, dan mengidentifikasi hambatan apa untuk maju.

Model Scrum melihat Scrum harian sebagai cara untuk menyelaraskan pekerjaan anggota tim saat mendiskusikan pekerjaan sprint. Di akhir sprint, tim melakukan tinjauan sprint dengan mendemonstrasikan fungsionalitas baru dari aplikasi Inventory yang dikembangkan kepada Product Owner yang ingin memberikan umpan balik yang dapat mempengaruhi sprint berikutnya.

Putaran umpan balik dalam pengembangan perangkat lunak Scrum ini dapat mengakibatkan perubahan pada fungsionalitas yang baru saja disampaikan, tetapi mungkin juga mengakibatkan merevisi atau menambahkan item ke backlog produk. Aktivitas lain dalam manajemen proyek Scrum adalah sprint retrospektif di akhir setiap sprint. Seluruh tim berpartisipasi dalam pertemuan ini, termasuk Scrum Master dan Product Owner. Pertemuan tersebut merupakan kesempatan untuk merefleksikan sprint yang telah berakhir dan mengidentifikasi peluang untuk berkembang.

### **3.2. Peran Strategis Scrum Master pada Pengembangan Aplikasi**

Dalam penerapan metode Scrum, ada beberapa masalah atau kendala yang dihadapi yang merupakan peran strategis Scrum Master yang diuraikan sebagai berikut.

- a. **Mengelola Ekspektasi *Roles***  
Selama mengembangkan aplikasi dengan metode *Scrum*, banyak pihak yang mencoba mempengaruhi peran Scrum Master. Namun, hal terpenting yang harus diketahui adalah bahwa Scrum Master adalah seorang fasilitator Scrum untuk tim maupun organisasi. Namun, terkadang ekspektasi manajemen berbeda yang membuat Scrum Master sedikit lebih sulit. Oleh karena itu, perlu memahami bahwa *Scrum* Master adalah seorang fasilitator, pemandu, pengikut proses, dan yang terpenting, Pemimpin yang Melayani [28].
- b. **Perubahan Resistensi**  
Perubahan pada aplikasi akan sering terjadi, hal ini karena pihak owner belum begitu memahami aplikasi yang diminta. Selain itu, Master Scrum akan mengalami aspek sosial perubahan yang dapat menciptakan perlawanan, semua penolakan berasal dari individu tertentu. Tim atau departemen tidak menolak untuk beralih ke Scrum, namun ada beberapa anggota tim yang pasti akan melakukan perlawanan terhadap Scrum Master. Ketika berbicara tentang implementasi Scrum, Scrum Master adalah agen perubahan yang sama. Hal ini adalah rintangan besar yang dihadapi oleh Scrum Master selama implementasi. Perubahan perlawanan bukanlah suatu kejutan namun reaksi yang paling diharapkan setiap kali perubahan baru diperkenalkan.
- c. **Menepati Jadwal**  
Ketika berbicara penerapan Sprint, setiap acara seharusnya dibatasi waktu untuk mendapatkan hasil yang produktif. Misalnya, acara Daily Scrum tidak boleh lebih dari 15 menit, tetapi kenyataannya anggota tim mulai membahas kesulitan teknis dan rapat berjalan lebih lama dari waktu yang ditentukan. Hal ini adalah salah satu tantangan paling umum bagi setiap Scrum Master. Salah satu cara untuk mengatasi rintangan ini adalah membuat anggota tim berdiri selama 15 menit dan menyembunyikan semua kursi sehingga akhirnya lelah dan menyelesaikan rapat.
- d. **Menangani Permintaan Perubahan yang Mendesak**  
Scrum Master mengikuti aturan saat mengimplementasikan Scrum, yaitu tidak pernah menerima perubahan dalam Sprint. Penanganan permintaan perubahan di akhir sprint atau di awal Sprint dapat diterima tetapi tidak diantara keduanya. Namun, dalam dunia praktis, Pemilik Produk, Pelanggan, atau Pemangku Kepentingan datang dengan permintaan perubahan atau bug yang mendesak. Namun, tidak baik juga mengikuti proses secara mendesak tanpa memahami aspek bisnis dan pasar. Selalu lebih baik untuk berkomunikasi, berkolaborasi dengan semua Pemangku Kepentingan, merencanakan ulang, dan kemudian membuat keputusan yang baik.
- e. **Pendistribusian Tim**  
Hal ini adalah salah satu hambatan paling umum yang dihadapi oleh seorang Scrum Master. Ketika tim tersebar secara geografis, terkadang ada penundaan, masalah jaringan, masalah budaya atau regional, zona waktu berbeda, jam kerja berbeda, selalu sulit untuk membuat semua orang terhubung dan berkolaborasi atau berkomunikasi dengan semua orang. Berkat semua teknologi, aplikasi diluar yang membantu mengatasi rintangan berupa komunikasi, konferensi video, dan banyak alat yang tersedia.



#### 4. KESIMPULAN

Dengan menggunakan praktik pengembangan aplikasi metode Scrum untuk mengembangkan aplikasi Android, beberapa hal yang menjadi tantangan sekaligus peran strategis bagi Scrum Master antara lain adalah pengelolaan peran dari tim yang terlibat, perubahan skenario dari pengembangan sistem, menjaga konsistensi dari penjadwalan, menangani permintaan dari Owner Product yang mendesak dan pendistribusian tim pengembang sistem. Peran strategis ini menjadi dasar untuk penunjukan atau pemilihan Scrum Master hendaknya mempertimbangkan dan dikualifikasi berdasarkan poin-poin tersebut.

#### REFERENSI

- M. El Bajta, "Analogy-Based Software Development Effort Estimation in Global Software Development," in 2015 IEEE 10th International Conference on Global Software Engineering Workshops, 2015, pp. 51–54, doi: 10.1109/ICGSEW.2015.19.
- P. S. Setiawan, M. I. Jambak, and M. I. Jambak, "The Effectiveness of Using Software Development Methods Analysis by the Project Timeline in an Indonesian Media Company," in 2017 International Conference on Data and Software Engineering (ICoDSE), 2017, pp. 1–6, doi: 10.1109/ICODSE.2017.8285890.
- A. Wahyudi, "Pengembangan Inventory Digital Berbasis Android dengan Metode SCRUM," *Fakt. Exacta*, vol. 11, no. 2, pp. 128–133, 2018.
- L. Wulandari, L. Sularto, T. Yusnitasari, and D. Ikasari, "User Requirements Analysis for Digital Library Application Using Quality Function Deployment.," *J. Phys. Conf. Ser.*, vol. 818, p. 12004, 2017, doi: 10.1088/1742-6596/818/1/012004.
- J. O'Connell, "School Libraries," in *Global Library and Information Science*, 2017.
- B. L. Romano and A. D. Da Silva, "Project Management Using the Scrum Agile Method: A Case Study within a Small Enterprise," in 2015 12th International Conference on Information Technology - New Generations, 2015, pp. 774–776, doi: 10.1109/ITNG.2015.139.
- H. Zahraoui and M. A. Janati Idrissi, "Adjusting Story Points Calculation in Scrum Effort & Time estimation," in 2015 10th International Conference on Intelligent Systems: Theories and Applications (SITA), 2015, pp. 1–8, doi: 10.1109/SITA.2015.7358400.
- Z. Wang, "Comparisons on Scrum Team Pairing Strategies: A Multi-agent Simulation," in 2020 IEEE 11th International Conference on Software Engineering and Service Science (ICSESS), 2020, pp. 224–227, doi: 10.1109/ICSESS49938.2020.9237643.
- A. Srivastava, S. Bhardwaj, and S. Saraswat, "SCRUM Model for Agile Methodology," in 2017 International Conference on Computing, Communication and Automation (ICCCA), 2017, pp. 864–869, doi: 10.1109/CCAA.2017.8229928.
- S. É. R. Ferrão and E. D. Canedo, "A Study of the Applicability of an Agile Methodology Scrum Allied to the Kanban Method," in 2015 10th Iberian Conference on Information Systems and Technologies (CISTI), 2015, pp. 1–6, doi: 10.1109/CISTI.2015.7170382.
- A. Taufiq, T. Raharjo, and A. Wahbi, "Scrum Evaluation to Increase Software Development Project Success: A Case Study of Digital Banking Company," in 2020 International Conference on Advanced Computer Science and Information Systems (ICACSIS), 2020, pp. 241–246, doi: 10.1109/ICACSIS51025.2020.9263235.
- Z. Masood, R. Hoda, and K. Blincoe, "Real World Scrum A Grounded Theory of Variations in Practice," *IEEE Trans. Softw. Eng.*, p. 1, 2020, doi: 10.1109/TSE.2020.3025317.
- L. Benedicenti et al., "Applying Scrum to the Army - A Case Study," in 2016 IEEE/ACM 38th International Conference on Software Engineering Companion (ICSE-C), 2016, pp. 725–727.
- S. Hermanto, E. R. Kaburuan, and N. Legowo, "Gamified Scrum Design in Software Development Projects," in 2018 International Conference on Orange Technologies (ICOT), 2018, pp. 1–8, doi: 10.1109/ICOT.2018.8705897.



- N. Naik and P. Jenkins, "Relax, It's a Game: Utilising Gamification in Learning Agile Scrum Software Development," in 2019 IEEE Conference on Games (CoG), 2019, pp. 1–4, doi: 10.1109/CIG.2019.8848104.
- M. Müller-Amthor, G. Hagel, M. Gensheimer, and F. Huber, "Scrum Higher Education – The Scrum Master Supports as Solution-focused Coach," in 2020 IEEE Global Engineering Education Conference (EDUCON), 2020, pp. 948–952, doi: 10.1109/EDUCON45650.2020.9125304.
- L. B. Angarita and J. A. G. Hernández, "Gamified System for Learning of Scrum Development Process," in 2019 14th Iberian Conference on Information Systems and Technologies (CISTI), 2019, pp. 1–6, doi: 10.23919/CISTI.2019.8760928.
- P. L. Ayunda and E. K. Budiardjo, "Evaluation of Scrum Practice Maturity in Software Development of Mobile Communication Application," in 2020 3rd International Conference on Computer and Informatics Engineering (IC2IE), 2020, pp. 317–322, doi: 10.1109/IC2IE50715.2020.9274625.
- P. M. Ching and J. E. Mutuc, "Modeling the Dynamics of an Agile Scrum Team in the Development of a Single Software Project," in 2018 IEEE International Conference on Industrial Engineering and Engineering Management (IEEM), 2018, pp. 386–390, doi: 10.1109/IEEM.2018.8607430.
- R. Plateaux, O. Penas, S. Mule, P. Hehenberger, S. Patalano, and F. Vitolo, "SCRUM++ Framework Concepts," in 2020 IEEE International Symposium on Systems Engineering (ISSE), 2020, pp. 1–8, doi: 10.1109/ISSE49799.2020.9272233.
- C. Matthies, "Playing With Your Project Data in Scrum Retrospectives," in 2020 IEEE/ACM 42nd International Conference on Software Engineering: Companion Proceedings (ICSE-Companion), 2020, pp. 113–115.
- R. Hanslo and P. E. Mnkandla, "Scrum Adoption Challenges Detection Model: SACDM," in 2018 Federated Conference on Computer Science and Information Systems (FedCSIS), 2018, pp. 949–957.
- K. Schwaber and J. Sutherland, "The Scrum Guide," 2013.
- S. Pfleeger and J. Atlee, *Software Engineering - Theory and Practice* (3. ed.). 2006.
- Sugiyono, *Metode Penelitian Kuantitatif, Kualitatif dan R&D*. Bandung: Alfabeta, 2010.
- M. Adnan and M. Afzal, "Ontology Based Multiagent Effort Estimation System for Scrum Agile Method," *IEEE Access*, vol. 5, pp. 25993–26005, 2017, doi: 10.1109/ACCESS.2017.2771257.
- H. R. Herdika and E. K. Budiardjo, "Variability and Commonality Requirement Specification on Agile Software Development: Scrum, XP, Lean, and Kanban," in 2020 3rd International Conference on Computer and Informatics Engineering (IC2IE), 2020, pp. 323–329, doi: 10.1109/IC2IE50715.2020.9274564.
- P. K. Linos, R. Rybarczyk, and N. Partenheimer, "Involving IT Professionals in Scrum Student Teams: An Empirical Study on the Impact of Students' Learning," in 2020 IEEE Frontiers in Education Conference (FIE), 2020, pp. 1–9, doi: 10.1109/FIE44824.2020.9274190.