

PERAN STRATEGIS *SCRUM MASTER* PADA PENGEMBANGAN PERANGKAT LUNAK PERPUSTAKAAN SEKOLAH BERBASIS ANDROID

Agung Wahyudi¹⁾, Sunardi²⁾, Imam Riadi³⁾

¹⁾Program Studi Magister Informatika Universitas Ahmad Dahlan

²⁾Program Studi Teknik Elektro Universitas Ahmad Dahlan

³⁾Program Studi Sistem Informasi Universitas Ahmad Dahlan

Jl. Ahmad Yani (Ringroad Selatan) Tamanan, Banguntapan, Bantul, Yogyakarta

e-mail: agungchandis1979@gmail.com¹⁾, sunardi@mti.uad.ac.id²⁾, imam.riadi@is.uad.ac.id³⁾

ABSTRAK

Kebutuhan akan fleksibilitas dan akses yang cepat menjadikan aplikasi berbasis mobile semakin banyak digunakan, termasuk dalam bidang pendidikan dan literasi. Penelitian ini bertujuan untuk mengkaji penerapan metode Scrum dalam pengembangan perangkat lunak mobile dengan mengkaji peran strategis Scrum Master. Scrum dikenal sebagai metode pengembangan aplikasi yang dinamis dan adaptif. Adapun tahapan dalam Scrum terdiri dari fase analisis kebutuhan sistem, design sistem, pengujian sistem, dan penerimaan produk. Metode penelitian yang digunakan adalah kualitatif dengan teknik wawancara sebagai cara pengumpulan data. Hasil wawancara kemudian direduksi, dianalisis dengan analisis dari literature, kemudian disajikan, dan disimpulkan. Penelitian ini dilakukan di SMA Negeri 88 Jakarta sebagai lokasi penerapan aplikasi perpustakaan online berbasis Android. Hasil dari penelitian ini menunjukkan bahwa peran strategis Scrum Master diantaranya adalah mengelola ekspektasi roles, perubahan resistensi, menepati jadwal, permintaan perubahan yang mendesak, dan pendistribusian tim. Peran strategis ini menjadi dasar untuk penunjukan atau pemilihan Scrum Master hendaknya mempertimbangkan dan dikualifikasi berdasarkan poin-poin tersebut.

Kata Kunci: *Dinamis, pengembangan aplikasi, scrum master.*

ABSTRACT

The need for flexibility and fast access makes mobile-based applications increasingly used, including in the fields of education and literacy. This study aims to examine the application of the Scrum method in mobile software development by examining the strategic role of the Scrum Master. Scrum is known as a dynamic and adaptive application development method. The stages in Scrum consist of system requirements analysis, system design, system testing, and product acceptance phases. The research method used is qualitative with interview techniques as a method of collecting data. The results of the interviews were then reduced, analyzed by analysis of the literature, then presented, and concluded. This research was conducted at SMA Negeri 88 Jakarta as the location for the application of an Android-based online library application. The results of this study indicate that the Scrum Master's strategic roles include managing role expectations, changing resistance, keeping schedules, urgent change requests, and team distribution. This strategic role forms the basis for the appointment or selection of a Scrum Master should be considered and qualified on the basis of these points.

Keywords: *Dynamic, Scrum master, software development.*

I. 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 pendidikan. 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 sekolah adalah perpustakaan. Perpustakaan adalah salah satu bagian kelengkapan yang harus ada disetiap lembaga pendidikan formal maupun non-formal di berbagai tingkatan karena perpustakaan sebagai guru kedua setelah guru yang ada di sekolah tersebut [3]. Perpustakaan adalah tempat dimana didalamnya terdapat banyak ilmu pengetahuan yang sangat bermanfaat bagi siswa [4].

Peran perpustakaan sangatlah signifikan dalam mencerdaskan masyarakat penggunaannya, khususnya dalam mencetak siswa berprestasi. Fasilitas perpustakaan yang baik membuat siswa dapat dan terbiasa belajar dengan baik. Sinergi antara siswa dan pustakawan akan berbuah prestasi bagi siswa serta kinerja yang baik bagi pustakawan

[5]. Pengembangan aplikasi atau *software* seperti aplikasi perpustakaan sekolah memerlukan metode pengembangan aplikasi. Metode ini sangat perlu dalam proses membuat aplikasi sehingga pengembang aplikasi memiliki pijakan untuk memulai aplikasi sampai aplikasi tersebut selesai. Salah satu cara dalam mengembangkan perangkat lunak adalah menggunakan *framework* Scrum [6], [7].

Scrum adalah pendekatan pengembangan perangkat lunak yang bersifat iteratif dan inkremental [8]. Metode ini pertama kali diperkenalkan oleh Ken Schwaber pada tahun 1997 pada makalah yang menjelaskan Metodologi Scrum pada tahun 1995 [9]. Alur kerja Scrum adalah urutan iterasi, bernama Sprint, yang memiliki durasi antara satu hingga empat minggu [10]. Tim dipandu oleh landasan kerja sebagai bagian dari *product backlog* yang merupakan daftar persyaratan dan prioritas [11]. Setiap Sprint memiliki pertemuan harian dengan setiap anggota tim mempresentasikan apa yang telah dilakukan pada hari sebelumnya, apa yang akan dilakukan hingga keesokan harinya, dan apakah ada hambatan untuk melanjutkan kegiatan pengembangan [12]. Di akhir setiap Sprint terdapat demo produk atau Review Sprint, dan setelah setiap Sprint Review ada sesi pembelajaran atau Retrospektif Sprint [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 seperti yang didefinisikan dalam Panduan Scrum yang telah disepakati oleh tim dan *owner project*. *Scrum Master* juga berperan membantu semua orang memahami teori dan praktik Scrum di dalam Tim Scrum maupun organisasi [17]. *Scrum Master* bertanggung jawab atas efektivitas Tim Scrum. *Scrum Master* melakukan ini untuk memungkinkan Tim Scrum untuk meningkatkan praktiknya dan kinerja selama proses pengembangan aplikasi. Dengan kata lain, *Scrum Master* adalah pemimpin sejati yang melayani Tim Scrum dan yang lebih besar [18]. Tanggung jawab *Scrum Master* pada pengembangan aplikasi dengan praktik Scrum dalam konteks pengembangan perangkat lunak Android, memastikan bahwa proses Scrum efektif dalam skenario pengembangan proyek dimana konsep sistem sering berubah dalam waktu yang cepat [16], [19].

Dari penjelasan diatas maka *Scrum Master* menjadi kunci keberhasilan pengembangan sistem. Akan tetapi, penggunaan metode Scrum, terdapat hambatan-hambatan tersendiri pada *Scrum Master* saat penggunaan metode tersebut. Hal ini menjadi tujuan dari penelitian ini untuk mengkaji apa saja yang dihadapi oleh pengembang sistem ketika menggunakan metode Scrum pada pengembangan aplikasi mobile Android. Dengan adanya kajian ini diharapkan bisa menjadi landasan bagi pengembang sistem ketika menggunakan Scrum untuk menciptakan *software* yang baik dan sesuai dengan keinginan pengguna.

II. METODOLOGI

A. Jenis Penelitian

Penelitian ini menggunakan metode kualitatif. Metode penelitian kualitatif menekankan pada metode penelitian observasi dan dialog (wawancara mendalam) di lapangan dan datanya dianalisis dengan cara non-statistik [25]. Pengumpulan data pada penelitian ini menggunakan teknik wawancara dengan informan penelitian yaitu seorang *Scrum Master*. Analisis data penelitian terdiri dari tiga tahap, yaitu reduksi, penyajian, dan kesimpulan.

B. Tinjauan Pustaka

1) Metode Agile

Pengembangan *Agile Software* merupakan salah satu metodologi dalam pengembangan suatu perangkat lunak. Kata Agile berarti cepat, ringan, bergerak bebas, waspada. Agile adalah kata yang digunakan untuk mendeskripsikan konsep model proses yang berbeda dengan konsep model proses yang sudah 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].

2) *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]:

a) *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.

b) *Artifacts*

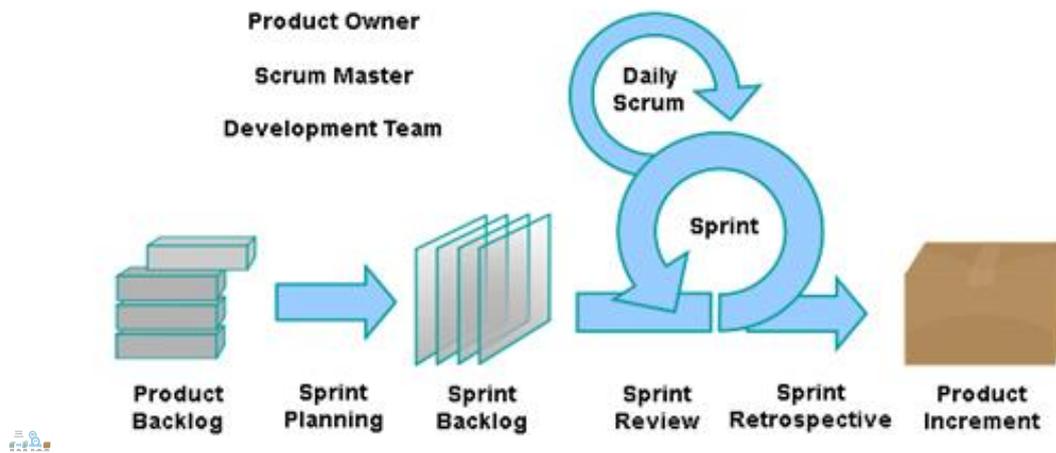
Artefak (artifacts) adalah hasil material atau non-material dari suatu pengembangan perangkat lunak. Dalam beberapa literatur, pada artefak ada beberapa hal yang perlu dipertimbangkan. Pertama adalah Product Backlog yang menentukan ruang lingkup dan daftar fitur yang harus dilaksanakan selama proyek 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.

c) *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.

3) *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

4) Software

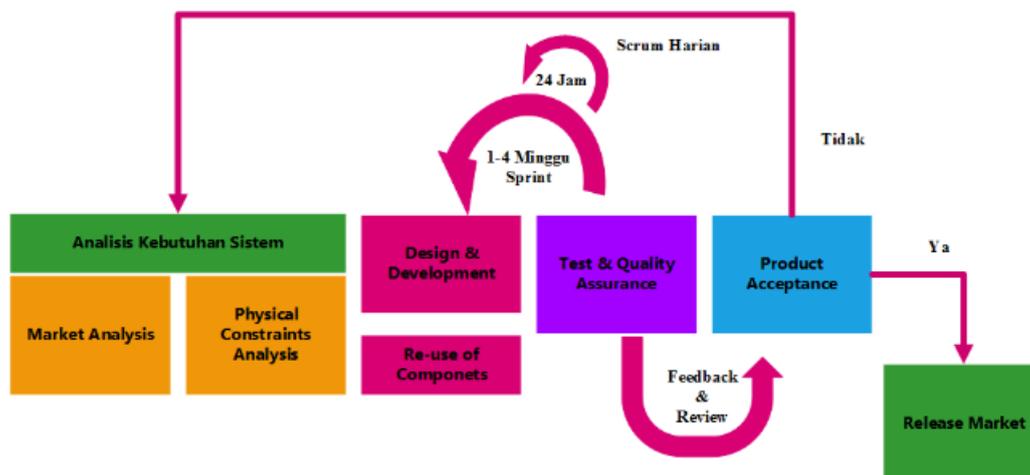
Perangkat lunak (*software*) merupakan program komputer yang berfungsi sebagai sarana interaksi antara pengguna dan hardware. Perangkat lunak juga dapat dikatakan sebagai "penerjemah" dari perintah-perintah yang dijalankan oleh pengguna komputer untuk diteruskan atau diproses oleh perangkat keras [24].

III. HASIL 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 perpustakaan online SMA Negeri 88 Jakarta.

A. Penerapan Metode Scrum dalam Pengembangan Perpustakaan Online

Metode pengembangan sistem dengan Scrum menyarankan agar proyek berkembang melalui serangkaian sprint [26]. Metode pengembangan perangkat lunak dengan Scrum pada aplikasi perpustakaan online SMA Negeri 88 Jakarta 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.

1) 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.

2) Desain sistem dan software (Design)

Proses desain sistem membagi kebutuhan menjadi hardware atau *software*. Hal ini menetapkan hampir seluruh

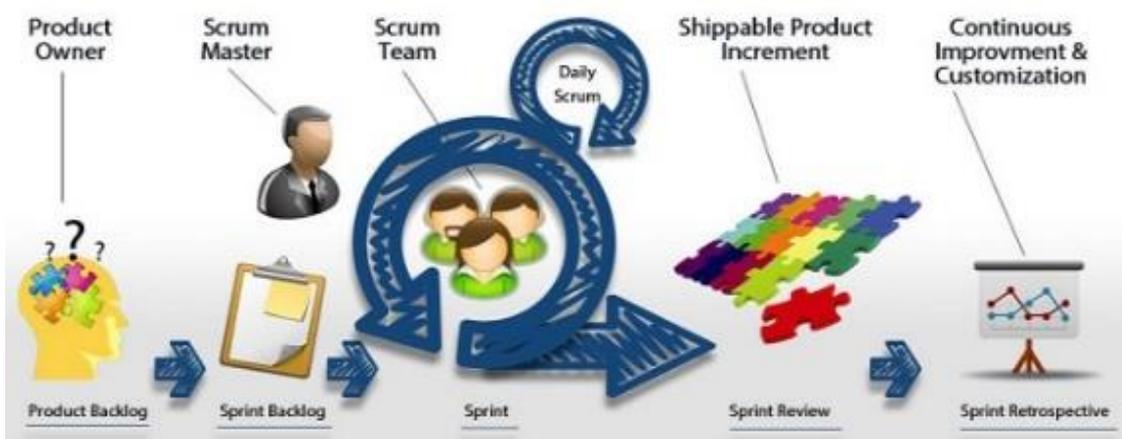
perancangan sistem. Desain *software* melibatkan identifikasi dan penggambaran mengenai pemisahan dasar system *software* dan hubungannya. Untuk dapat dimengerti oleh mesin, dalam hal ini adalah komputer, maka desain tadi harus diubah menjadi bentuk yang dapat dimengerti oleh mesin, yaitu kedalam bahasa pemrograman melalui proses coding. Tahap ini merupakan implementasi dari tahap design yang secara teknis nantinya dikerjakan oleh *programmer*.

3) *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.

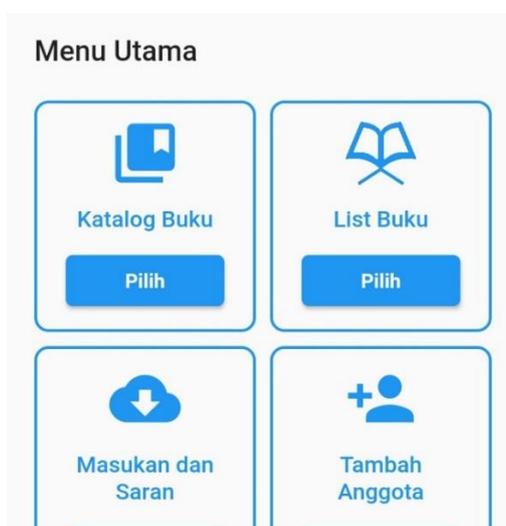
4) *Product Acceptance*

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]. Perpustakaan online yang dibuat ini terdiri dari empat menu utama yang disepakati dengan *Product Owner*, yaitu katalog Buku, List Buku, Masukan dan Saran, dan Tambah Anggota seperti ditampilkan pada Gambar 4.



Gambar 4. Menu Utama Perpustakaan Online

Selama pengembangan perpustakaan online, 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 perpustakaan 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.

B. 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.

1) 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].

2) 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.

3) 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.

4) 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.

5) 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.

IV. 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.

DAFTAR PUSTAKA

- [1] 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.
- [2] 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.
- [3] A. Wahyudi, "Pengembangan Perpustakaan Digital Berbasis Android dengan Metode SCRUM," Fakt. Exacta, vol. 11, no. 2, pp. 128–133, 2018.
- [4] 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.
- [5] J. O'Connell, "School Libraries," in Global Library and Information Science, 2017.
- [6] 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.
- [7] 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.
- [8] 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.
- [9] 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/ICCAA.2017.8229928.
- [10] 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.
- [11] 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.
- [12] 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.
- [13] 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.
- [14] 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.
- [15] 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.
- [16] 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.
- [17] 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.
- [18] 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.
- [19] 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.
- [20] 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.
- [21] C. Mathies, "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.
- [22] 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.
- [23] K. Schwaber and J. Sutherland, "The Scrum Guide," 2013.
- [24] S. Pfeleeger and J. Atlee, *Software* Engineering - Theory and Practice (3. ed.). 2006.
- [25] Sugiyono, *Metode Penelitian Kuantitatif, Kualitatif dan R&D*. Bandung: Alfabeta, 2010.
- [26] 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.
- [27] 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.
- [28] 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.