

Implementasi *Scrum* pada Manajemen Proyek Pengembangan Perangkat Lunak Pemesanan Undangan (Studi Kasus: Paperlust)

Sari Kurnia Ningrum¹
Fakultas Teknologi Industri
Universitas Islam Indonesia
Yogyakarta, Indonesia
16523060@students.uui.ac.id

Andhik Budi Cahyono²
Fakultas Teknologi Industri
Universitas Islam Indonesia
Yogyakarta, Indonesia
andhik.budi@uui.ac.id

Abstrak— *Scrum* merupakan kerangka kerja yang mengimplementasikan proses *agile development*. *Scrum* memungkinkan adanya penyelesaian tipe masalah yang berubah-ubah dan kompleks secara bersamaan. Tumpuan pada kerangka kerja ini terdapat pada kekuatan kolaborasi tim, *incremental product* dan proses iterasi untuk mewujudkan hasil akhir yang memiliki nilai tinggi. Selama proses pelaksanaan kerangka kerja *scrum* pada pengembangan perangkat lunak Paperlust selama magang berlangsung, aktivitas *sprint* yang mendukung di antaranya: *Sprint Planning*, *Daily Meeting and Report*, *Sprint Review*, dan *Weekly Meeting*. Apabila dibandingkan dengan metode *waterfall* yang diterapkan pada awal pengembangan sebelumnya, kemampuan *scrum* sangat cocok dengan kebutuhan pengembangan perangkat lunak Paperlust yang mempunyai variasi penambahan dan perubahan kebutuhan yang harus segera diimplementasikan. Selain itu, adanya proses inspeksi dan peninjauan yang dilakukan pada setiap modul pengerjaan menghasilkan umpan balik yang lebih cepat. Maka dari itu, tujuan akhir dari implementasi kerangka kerja *scrum* pada pengembangan perangkat lunak Paperlust adalah tercapainya proses eksekusi pemenuhan kebutuhan dan penyelesaian masalah terlaksana dengan efektif dan efisien sampai pada saat ini.

Keywords— *Scrum*, *Pengembangan Perangkat Lunak*, *Paperlust*

I. PENDAHULUAN

Paperlust merupakan wadah pemesanan undangan secara daring berbasis *website*. Paperlust menjadi jembatan yang memudahkan para desainer serta konsumen dalam memenuhi tujuannya. Bekerja sama dengan para desainer lokal maupun internasional berbakat, Paperlust menawarkan berbagai jenis desain undangan untuk setiap kesempatan di antaranya termasuk pernikahan, pertunangan, ulang tahun, *baby shower* dan lain sebagainya. Paperlust juga menyediakan jasa untuk mencetak desain yang telah dipilih serta dimodifikasi sesuai dengan tipe kertas dan jenis cetak.

Paperlust berawal dari kesulitan para desainer kartu undangan untuk memasarkan desainnya, sementara konsumen, yang mana adalah para penyuka desain, kesulitan menemukan desain yang cocok untuk

membagikan momen kepada orang lain di antara berbagai situs yang ada. Permasalahan selanjutnya adalah sulit menemukan jasa percetakan yang mempunyai jenis dan kualitas kertas yang tepat sehingga dapat mendukung kualitas desain yang akan dicetak.

Pada awal pengembangan, Paperlust menggunakan metode pengembangan *waterfall*. *Waterfall* adalah sebuah metode klasik yang bersifat sistematis atau berurutan dalam membangun perangkat lunak [1]. Seiring berjalannya waktu, penggunaan metode ini sudah tidak lagi efektif. Hal ini dipicu dengan bertambah dan bervariasinya kebutuhan konsumen. Proses implementasi kebutuhan tersebut ke *website* Paperlust juga harus dilakukan dengan cepat. Salah satu jalan adalah mengeksekusi bentuk penyelesaian dengan bersamaan, sedangkan *waterfall* tidak memiliki kemampuan seperti ini. Dalam upaya pemenuhan kebutuhan tersebut maka Paperlust beralih menggunakan metode pengembangan *agile* dengan kerangka kerja *scrum*.

Tidak seperti metode *waterfall* yang mengharuskan pekerjaan mengalir secara bertahap dan linear, *scrum* memungkinkan adanya penyelesaian tipe masalah yang berubah-ubah dan kompleks secara bersamaan [2]. *Scrum* bertumpu pada kekuatan kolaborasi tim, *incremental product* dan proses iterasi untuk mewujudkan hasil akhir yang memiliki nilai tinggi [2]. Kelebihan lain dari kerangka kerja ini adalah adanya proses inspeksi dan peninjauan pada setiap pengerjaan serta fleksibilitas pengerjaan yang dapat disesuaikan dengan perkembangan teknologi [3]. Selain itu, proses pengujian dapat dilakukan pada setiap modul selama pengembangan berlangsung. Sebuah aplikasi manajemen proyek perangkat lunak dikatakan mempunyai kualitas yang baik, apabila kebutuhan tim proyek dapat disesuaikan secara fleksibel [4].

Berdasarkan uraian yang telah dikemukakan sebelumnya, Paperlust memilih metode pengembangan *scrum* dengan harapan proses eksekusi pemenuhan kebutuhan dan penyelesaian masalah dapat terlaksana dengan efektif dan efisien sampai

pada saat ini. [1] [2]



II. DASAR TEORI

A. Agile

Agile development adalah salah satu metode pengembangan perangkat lunak yang mengedepankan interaksi anggota tim dan kolaborasi dengan klien daripada proses dan jenis perangkat yang digunakan [7]. Metode *agile* cocok diterapkan pada perangkat lunak yang diharuskan responsif serta toleran terhadap perubahan kebutuhan yang cepat.

Prioritas utama penerapan *agile* pada sebuah perangkat lunak adalah kepuasan pelanggan melalui produk perangkat lunak secara berkelanjutan. Salah satu prinsip dari *agile* untuk mendukung hal ini salah satunya adalah proses pemaparan hasil yang dilakukan dalam jangka waktu dua sampai empat minggu, dengan preferensi pada skala waktu yang lebih pendek [5].

B. Perangkat Lunak

Perangkat lunak adalah sebuah program komputer yang menyediakan layanan interaksi antara pengguna dan perangkat keras [8]. Perangkat lunak dibangun melalui proses rekayasa yang terstruktur dengan berbagai metode tergantung kebutuhan fungsionalitas dan hasil akhir yang diharapkan.

C. Scrum

Scrum dikembangkan Jeff Sutherland pada 1993 yang dengan tujuan sebagai metode pengembangan dan pengelolaan yang mengikuti prinsip *agile* [6]. Pengembangan *scrum* selanjutnya dilakukan oleh Schwaber dan Beedle. *Scrum* sendiri mempunyai proses yang kompleks karena adanya banyak faktor yang memengaruhi hasil akhir.

Scrum yang terdiri dari *scrum team* dan peran-peran yang diperlukan, acara (*event*), artefak (*artifact*), dan aturan main [1]. Jantung dari pelaksanaan *scrum* adalah *sprint*. *Sprint* merupakan batasan waktu yang diberikan untuk menyelesaikan masalah. Pada praktiknya, Paperlust menetapkan waktu selama dua minggu untuk setiap satu *sprint* yang berjalan. Tahapan *sprint* yang berjalan di antaranya: *Sprint Planning*, *Daily Meeting and Report*, *Sprint Review*, dan *Weekly Meeting*.

Proses implementasi kerangka kerja ini memiliki *key practice* yaitu: (1) *Scrum* memungkinkan pengerjaan dan pengumpulan kebutuhan, perancangan arsitektur dan antarmuka secara bersamaan, (2) Fokus pada *sprint*, pengkajian hasil, dan jadwal pengerjaan, (3) Fokus pada jadwal yang telah disepakati, (4) Bekerja sesuai dengan *sprint* secara konsisten dan terstruktur, (5) Semua pekerjaan ditandai sebagai *product backlog*, (6) *Product*

backlog dasar melakukan *sprint* dan tim harus dapat memutuskan skala prioritas terhadap daftar *product backlog* yang telah disusun sebelumnya, (7) Melakukan pertemuan setiap hari, (8) *Scrum master* bertanggung jawab menerima dan mengevaluasi hasil *sprint* [7].



Gambar 1 Konseptual kerangka kerja *scrum*

III. METODOLOGI

Selama aktivitas magang, tahapan pengumpulan informasi mengenai hambatan dan tantangan dalam proses implementasi kerangka kerja *scrum* dilakukan dengan cara konsultasi dengan beberapa orang yang telah terlibat langsung. Cara ini memberikan pengetahuan tentang kerangka kerja *scrum* dan implementasinya pada manajemen proyek pengembangan perangkat lunak Paperlust secara mendalam. Hasil akhir yang didapatkan kemudian dikomparasikan dengan informasi yang didapatkan dari studi literatur.

IV. PEMBAHASAN DAN HASIL

A. Pembahasan

Implementasi kerangka kerja *scrum* pada pengembangan perangkat lunak Paperlust terbagi atas peran-peran sebagai berikut:

- *Project Manager*, bertanggung jawab atas monitor, kontrol dan pengelolaan produk secara keseluruhan.
- *Growth Team*, terdiri atas beberapa peran yang bertanggung jawab diantaranya: pemasaran produk, analisa data, dan perancangan bentuk eksekusi kebutuhan konsumen.
- *Quality Assurance (QA)*, bertanggung jawab untuk memastikan implementasi solusi yang telah dikerjakan oleh tim pengembang berjalan sesuai dengan kesepakatan bahkan dengan *silly flow* sekalipun.
- *Software Developer Team*, bertanggung jawab atas eksekusi dan implementasi rancangan kebutuhan konsumen dalam bentuk kode program.

Pengembangan perangkat lunak Paperlust mengadopsi konseptual proses *scrum* seperti yang diperlihatkan pada Gambar 1 dengan langkah-langkah sebagai berikut:

1) *Sprint Planning*

Sprint Planning dapat digambarkan sebagai tahap analisa yang berfokus pada masalah yang akan diselesaikan. Durasi *sprint planning* dapat berbeda-beda berdasarkan seberapa besar dan kompleks masalah yang akan diselesaikan. Pada tahap ini, *scrum master* harus memastikan semua anggota dapat memahami masalah serta metode penyelesaiannya [2].

Paperlust menerapkan *sprint planning* secara kolaboratif dengan komponen analisa meliputi: penjelasan alur pengguna serta kebutuhan fungsionalitas, menguraikan *product backlog*, menentukan *sprint goal* atau MVP (*Minimum Viable Product*), penjabaran tugas masing-masing anggota, dan menentukan estimasi *sprint* selesai hingga pada tahap *staging live check*.

Selama aktivitas magang berlangsung, terdapat lima aktivitas yang dikerjakan dengan ketentuan yang berbeda beda seperti yang terlihat pada Tabel 1.

Tabel 1 *Product backlog* Paperlust

No	Jenis Aktivitas	Deskripsi
1	Tombol hapus pada <i>community review</i> di halaman administrator	Menambahkan tombol hapus pada tabel <i>community review</i> di halaman administrator.
2	<i>Popup login/register</i> [Studi kasus: <i>Mobile</i>]	Memperbaharui perilaku <i>login/register</i> pengguna menggunakan <i>popup</i> [studi kasus: <i>mobile</i>] yang semula masih diarahkan ke halaman baru.
3	Membuat “get sample” <i>trrello card</i> secara otomatis	Membuat fungsi <i>helper</i> baru untuk menangani proses <i>generate trrello card</i> secara otomatis serta memisahkannya berdasarkan negara tujuan pengiriman. Selain itu juga membuat fungsi serupa yang dapat dijalankan di CRON.
4	Fungsi <i>email reminder</i> untuk daftar nama tamu.	Memperbaharui alur pengiriman <i>email</i> kepada konsumen yang semula menggunakan <i>mailto()</i> menjadi menggunakan MailChimp, Mandrill. Selain itu menambahkan keterangan waktu terakhir kali <i>email reminder</i> dikirimkan.

5	Integrasi One Tree Planted	Membuat halaman baru sebagai pusat informasi mengenai projek One Tree Planted yang dilaksanakan oleh Paperlust, menambahkan beberapa komponen kecil pada halaman <i>payment</i> dan <i>cart page</i> , serta membuat <i>banner</i> informasi yang ada di halaman <i>browse design</i> .
---	----------------------------	---

2) *Daily Meeting and Report*

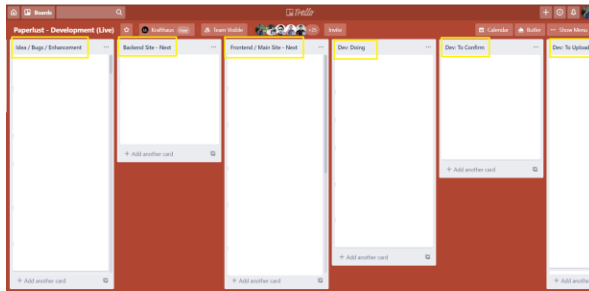
Daily meeting umumnya berdurasi selama kurang lebih 15 menit untuk setiap pertemuannya, namun pada praktiknya, durasi dapat disesuaikan tergantung jumlah anggota dan jenis pembahasan. Struktur dari pertemuan ini fleksibel dan dapat diadakan lewat berbagai macam cara selama pertemuan ini fokus terhadap kemajuan menuju *sprint goal* [2]. *Daily Meeting and Report* berfungsi untuk mengoptimalkan kolaborasi dan performa dari tim dengan melakukan inspeksi pada pekerjaan yang dilakukan semenjak *daily meeting* sebelumnya [2].

Komponen pembahasan pada *daily meeting developer team* pada tahap ini di antaranya: hal yang telah dilakukan sejak pertemuan terakhir tim, kendala yang ditemui selama pengerjaan, serta rencana yang telah disusun untuk mencapai sesuatu sebelum rapat tim berikutnya. Tujuan dari tahap ini adalah dapat mengetahui sisa pekerjaan yang perlu dikejar selama sisa waktu *sprint*. Tahap ini dilakukan setiap hari kecuali pada hari Jum'at selama *sprint* berlangsung.

Setelah *product backlog* diidentifikasi seperti pada Tabel 1, *project manager* dan *team lead* melakukan proses monitor dan kontrol melalui *trrello board* seperti pada Gambar 2. Proses kerja terdiri dari beberapa komponen diantaranya: *Idea/Bugs/Enhancement, Backend Site and Frontend Site-Next, Dev: Doing-To Confirm, Check on Sandbox, Live Check*.

Idea/Bugs/Enhancement dan *Backend Site and Frontend Site-Next* digunakan untuk menguraikan tugas atau kasus yang direncanakan untuk dikerjakan. Pekerjaan yang masih dalam proses pengerjaan dimasukkan ke dalam *Dev: Doing*, sedangkan yang telah selesai dikerjakan masuk ke proses *staging* pertama yaitu *Dev: To Confirm*. Adanya ketidaksesuaian komponen hasil, pekerjaan akan menerima umpan balik dari tim QA. Apabila tim QA telah selesai melakukan pengecekan, *card* akan dikembalikan ke tim *software developer* agar umpan balik dapat segera implementasi.

Staging selanjutnya adalah *Check on Sandbox*. *Sandbox* adalah lingkungan pengujian yang disediakan sebelum kode program di implementasikan ke *live code*. Hal ini dimaksudkan untuk mencegah terjadinya kerusakan kode program atau adanya *bugs* yang tidak terdeteksi sebelumnya. Apabila pada tahap ini kode program berjalan dengan baik dan lolos pengecekan yang dilakukan oleh tim QA, kode program siap diimplementasikan ke *live code*.



Gambar 2 Trello board Paperlust

3) Sprint Review

Sprint review dilaksanakan untuk menginspeksi *increment* dan meninjau apa saja pekerjaan yang telah dan belum diselesaikan. Inspeksi dan peninjauan tidak hanya dilakukan pada hasil, proses bekerja tim juga dinilai sehingga dapat ditentukan apa saja yang perlu ditingkatkan untuk proses pengembangan pada *sprint* berikutnya [2]. *Sprint* dianggap selesai apabila telah menghasilkan produk (*deliverable product*) yang sesuai dengan *acceptance criteria* yang telah ditetapkan dan disepakati sebelumnya.

Tim *developer* Paperlust melakukan tahap ini setiap minggu di hari Jum'at sebelum pelaksanaan *weekly meeting*. Berbeda dengan ketentuan *sprint review* yang berlangsung pada akhir masa *sprint*, pemilihan waktu ini didasarkan pada proses inspeksi dan peninjauan perkembangan pekerjaan yang lebih cepat. Pada tahap ini, *developer* juga dapat melakukan negosiasi penambahan waktu *sprint* sesuai dengan kendala yang ditemui selama pengerjaan bila diperlukan.

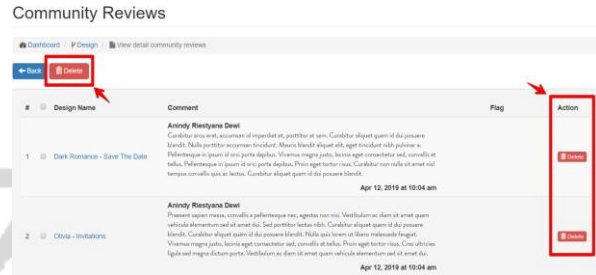
4) Weekly Meeting

Pada tahap ini semua anggota tim Paperlust melaporkan pencapaian yang didapat selama satu minggu, kendala yang dialami, serta penyampaian ide atau usulan untuk pengembangan Paperlust ke depannya. Hasil diskusi dari pertemuan ini akan masuk dalam daftar *backlog* tim yang bersangkutan.

B. Hasil

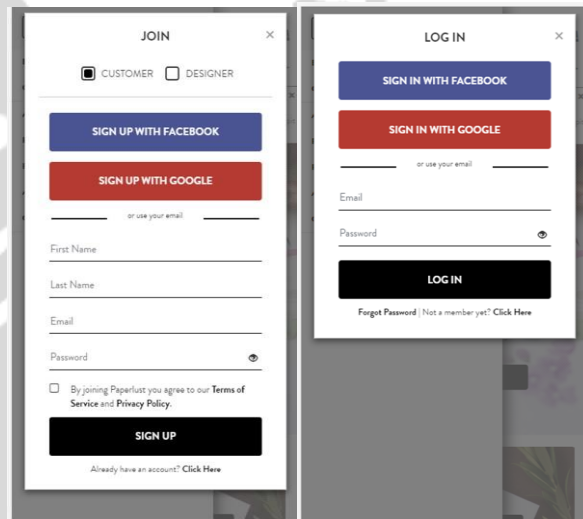
Berdasarkan pembahasan proses implementasi kerangka *scrum* dalam pengembangan perangkat lunak Paperlust menghasilkan hal berikut:

- 1) Tombol hapus pada *community review* di halaman administrator



Gambar 3 Halaman *community review* back-end

- 2) *Popup login/register* [Studi kasus: *Mobile*]



Gambar 4 Blanko *join* dan *login* versi *mobile* yang baru

- 3) Membuat “get sample” *trello card* secara otomatis



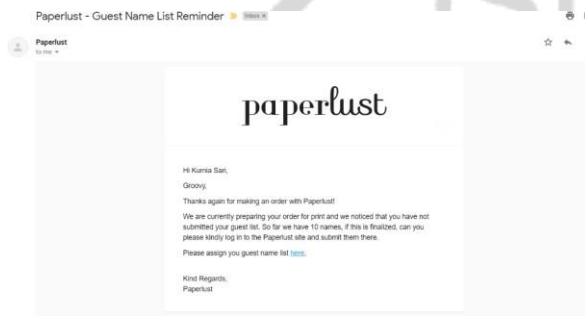
Gambar 5 Bagian tombol *generate and view trello card*

- 4) Fungsi *email reminder* untuk daftar nama tamu.

Number of Valid Guest Name	: 3 of 100
Guest Names / Table Numbers State	: Not submitted yet Force Submit
Remind To Customer	: Contact Customer Last sent : 01 Jul 2019 12:50:52
Special Request	

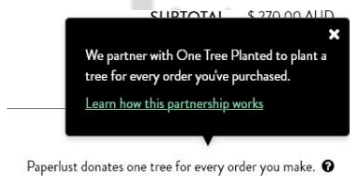
[Download](#)

Gambar 6 Bagian tombol pengiriman email dan



Gambar 7 Kerangka email yang diterima oleh customer

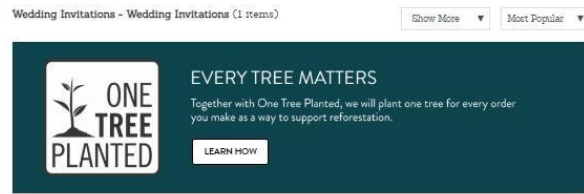
5) Integrasi One Tree Planted



Gambar 7 Tooltip informasi di halaman



Gambar 8 Halaman informasi One Tree Planted



Gambar 9 Banner One Tree Planted pada halaman browse design Paperlust

V. KESIMPULAN

Berdasarkan uraian yang telah dikemukakan sebelumnya, maka beberapa kesimpulan dapat ditarik sebagai berikut:

- Kualitas dan risiko proyek terlihat lebih cepat. Hal ini dikarenakan pengecekan dan peninjauan dilakukan secara berkala dan per modul.
- Umpan balik diterima tanpa menunggu hasil sepenuhnya diimplementasikan
- Adanya kolaborasi sebagai tim yang kuat dan proses iterasi pengembangan produk sebagai tumpuan, membuat hasil akhir mempunyai nilai yang tinggi.
- Pendekatan bertahap dan berkelanjutan yang diterapkan di *scrum* mampu mengoptimalkan kemampuan prediksi dan mengendalikan risiko

REFERENSI

- [1] R. S. Pressman, *Rekayasa Perangkat Lunak: Pendekatan Praktisi Ed. 7*, Yogyakarta: ANDI, 2012.
- [2] K. S. dan J. S., *Panduan Scrum*, 2017.
- [3] A. I. Agrawal S, "Mobile Application Development: A Development Survey," 2010.
- [4] G. Maatita, F. Samopa dan R. Wibowo, "Pengembangan Aplikasi Manajemen Proyek

Perangkat Lunak Berbasis Spring: Modul Core System dan Management Source Code,” 2011.

- [5] R. C. Martin dan M. Martin, Agile Principles, Patterns, and Practice in C#, Upper Saddle River: Pearson Education, 2006.
- [6] A. e. a. Pham, Scrum in Action Agile Software Project Management and Development, Boston: Course Technology PTR, 2011.
- [7] R. C. Martin, Agile Software Development: Principles, Patterns, and Practice, Upper Saddle River: Pearson Education, 2003.
- [8] S. L. J. M. A. Pfleeger, Software Engineering: Theory and Practice, Upper Saddle River: Pearson Prentice Hall, 2006.
- [9] R. S. Kenett dan E. Beker, Process Improvement and CMMI for Systems and Software, Francis: CRC Press, 2010.

