

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

by Sari Kurnia Ningrum

Submission date: 20-Nov-2019 02:59PM (UTC+0700)

Submission ID: 1216315904

File name: ngan_Perangkat_Lunak_Pemesan_Undangan_Studi_Kasus_Paperlust.pdf (441.73K)

Word count: 2316

Character count: 15014

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

Abstrak— *Scrum* adalah 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. *Scrum* didukung oleh aktivitas *sprint* yang diantaranya: *Sprint Planning*, *Daily Meeting*, *Sprint Review*, dan *Sprint Retrospective*. Kemampuan *scrum* sangat cocok dengan kebutuhan pengembangan perangkat lunak Paperlust yang mempunyai variasi perubahan dan penambahan kebutuhan yang harus segera diimplementasikan. Ada beberapa hal yang dimodifikasi dari teori atau panduan *scrum* yang telah ada sebelumnya. Hal ini dimaksudkan untuk menyesuaikan keadaan yang sedang terjadi selama proses pengembangan Paperlust.

Keywords— *Scrum*, Pengembangan Perangkat Lunak, Paperlust

I. PENDAHULUAN

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

Paperlust berawal dari kesulitan para perancang kartu undangan untuk memasarkan desainnya, sementara para desain antusias kesulitan menemukan desain yang cocok untuk membagikan momen kepada orang lain di antara berbagai situs yang ada. Sulit menemukan jasa percetakan yang mempunyai jenis dan kualitas kertas yang tepat juga masuk kedalam daftar permasalahan.

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 para desain antusias. Proses eksekusi kebutuhan tersebut agar dapat diimplementasikan 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.

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

II. DASAR TEORI

A. Agile Development

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 [4]. Metode *agile* cocok diterapkan pada perangkat lunak yang diharuskan responsif terhadap perubahan kebutuhan yang cepat.

B. Perangkat Lunak

Perangkat lunak adalah sebuah program komputer yang menyediakan layanan interaksi antara pengguna dan perangkat keras [5]. 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 mempengaruhi hasil akhir.



Gambar 1 Konseptual kerangka kerja *scrum*

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 2 minggu untuk setiap satu *sprint* yang berjalan. Tahapan selama *sprint* yang berjalan diantaranya: *Sprint Planning*, *Daily Meeting and Report*, *Sprint Review*, dan *Weekly Meeting* merupakan modifikasi kerangka kerja *scrum*.

III. METODOLOGI

Model pengembangan yang digunakan selama aktivitas magang adalah *scrum*. *Scrum* adalah kerangka kerja yang mengimplementasikan proses *agile development* serta memiliki *key practice* yaitu: (1) *Scrum* memungkinkan pengerjaan dan pengumpulan kebutuhan, perancangan arsitektur dan antarmuka secara bersamaan, (2) Fokus pada *sprint*, review hasil, dan jadwal pengerjaan, (3) Fokus pada jadwal, (4) Bekerja sesuai dengan *sprint* secara konsisten dan terstruktur, (5) Semua pekerjaan harus 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].

Pada proses pengembangannya, Paperlust mengadopsi konseptual proses *scrum* yang diperlihatkan pada Gambar 1 dengan langkah-langkah sebagai berikut :

A. 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, menentukan *sprint goal* atau MVP (*Minimum Viable Product*), penjabaran tugas masing-masing anggota, dan menentukan estimasi *sprint* selesai hingga pada tahap *live check*.

B. Daily Meeting and Report

Daily meeting umumnya berdurasi selama kurang lebih 15 menit untuk setiap pertemuannya, namun pada prakteknya, durasi dapat disesuaikan tergantung jumlah anggota dan jenis pembahasan. Struktur dari pertemuan ini fleksibel dan bisa 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 diantaranya: 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. Tahap ini dilakukan setiap hari kecuali pada hari jum'at selama *sprint* berlangsung.

C. Sprint Review

Sprint review dilaksanakan untuk untuk menginspeksi *increment* dan meninjau apa saja pekerjaan yang telah dan belum diselesaikan. Inspeksi dan peninjauan tidak hanya dilakukan pada hasil, tapi juga bagaimana tim bekerja sehingga dapat ditentukan segala sesuatu 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.

D. 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 kedepannya. Hasil diskusi dari pertemuan ini akan masuk dalam daftar *backlog* tim yang bersangkutan.

IV. HASIL DAN PEMBAHASAN

Sesuai dengan pengadaptasian kerangka kerja *scrum*, proses pengembangan menghasilkan hal-hal seperti berikut:

A. Menambahkan fungsi *delete* pada *community review* di halaman *back-end*

Untuk memperkaya informasi pada setiap produk desain, Paperlust menyediakan fitur *review* yang bisa ditambahkan oleh pengguna. Adanya *review* yang diberikan akan meningkatkan reputasi desain tersebut pada mesin pencari. Semakin banyak *review* yang diberikan, ketika kata kunci yang dimasukkan cocok dengan desain tersebut, maka desain akan berada pada baris hasil teratas. Meningkatnya reputasi setiap produk desain yang ada di Paperlust juga akan mempengaruhi reputasi Paperlust itu sendiri.

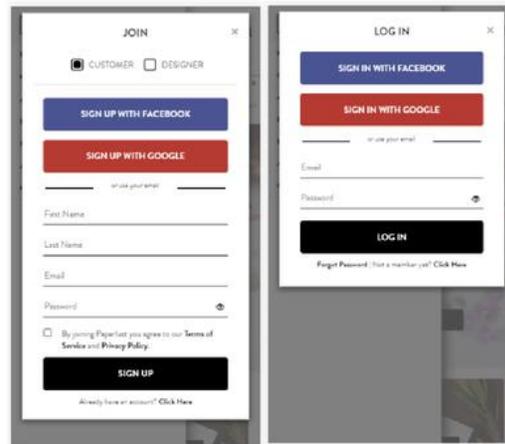
Disisi lain dampak dari penambahan *review* berlebih, apalagi mempunyai konten yang sama, justru akan menurunkan reputasi desain tersebut. Mesin pencari akan menganggap desain ini sebagai *spam* dan membuat reputasi Paperlust semakin buruk. Tentu hal ini akan mempengaruhi *user visit* dan jumlah transaksi di Paperlust.

Solusi alternatif untuk mencegah kemungkinan penurunan reputasi di mesin pencari adalah menyediakan fitur dasar untuk menghapus *review* yang dimungkinkan sebagai *spam* seperti yang terlihat pada **Gambar 2.1**. Nantinya fitur ini akan digunakan oleh tim *Account Executive* dalam proses seleksi tersebut.



Gambar 2 Halaman *community review back-end*

B. Mengubah perilaku login/ register pengguna menggunakan popup [studi kasus: *mobile*]



Gambar 3 Blanko *join* dan *login* versi *mobile* yang baru

Paperlust mempunyai beberapa jenis atau cara *login* baik sebagai *customer* atau *designer*. Pada perangkat *mobile*, ketika pengguna belum masuk ke sistem, pengguna akan diarahkan ke halaman baru. Sedangkan pada *desktop* dan *iPad* atau *tablet*, pengguna akan disuguhkan sebuah *popup*.

Permasalahan yang dihadapi adalah ketika kondisi percabangan ingin diubah, maka tim pengembang yang bertugas harus mengubah di beberapa tempat dengan model berbeda – beda. Untuk mengurangi beban kerja tim pengembang, tim *Quality Assurance* (QA) menyarankan model atau cara pengguna untuk masuk ke sistem menggunakan satu jenis model saja yaitu *popup*, baik pada perangkat *mobile*, *tablet* atau *desktop* seperti yang terlihat pada Gambar 3.

C. Membuat dan memisahkan kartu trello untuk jenis produk “get sample” berdasarkan negara tujuan pengiriman.

Pada awalnya, kartu trello dibuat manual oleh tim *Account Executive* setelah mereka menerima *email* dari *customer*. Selain itu mereka juga harus memisahkan

order tersebut ke beberapa board trello berbeda berdasarkan *continent* negara tujuan pengiriman paket undangan. Untuk memudahkan tim AE dalam melakukan pekerjaan ini, solusi alternatif yang ditemukan adalah menambahkan button untuk membuat card trello dari halaman back-end seperti yang terlihat pada Gambar 2.3. Button tersebut hanya akan muncul pada daftar order masuk yang sudah dibayar baik menggunakan kartu kredit atau Paypal dan yang hanya dengan target pengiriman di beberapa negara tertentu. Setelah card trello berhasil dibuat, button akan berubah fungsi untuk melihat card yang sudah terbuat sebelumnya.

Selain bisa dilakukan satu per satu berdasarkan order yang masuk dan terdaftar, dibuat juga sebuah CRON untuk membuat card trello secara otomatis berdasarkan tanggal yang dimasukkan. Namun apabila tidak ada tanggal yang dimasukkan sebagai parameter fungsinya, maka CRON akan membuat card trello berdasarkan order yang masuk pada hari dimana CRON dijalankan. CRON sendiri adalah penjadwal pekerjaan berbasis waktu di sistem operasi komputer mirip Unix.



Gambar 4 Bagian tombol *generate and view trello card* di halaman *back-end*

D. Memperbaharui fungsi *email reminder* untuk daftar nama tamu.

Salah satu layanan Paperlust adalah *customer* dapat menyertakan nama tamu yang akan mereka undang di setiap desain yang mereka pesan. Akan tetapi ada sebuah kasus dimana *customer* tidak mengisi daftar nama tamu yang mereka undang pada sejumlah desain yang mereka pesan atau ketika status dari daftar tamu tersebut belum di konfirmasi. Untuk menangani hal ini, pihak terkait perlu mengirimkan sebuah *email* untuk mengingatkan *customer* untuk segera mengkonfirmasi daftar tamu undangan pada desain yang mereka pesan.

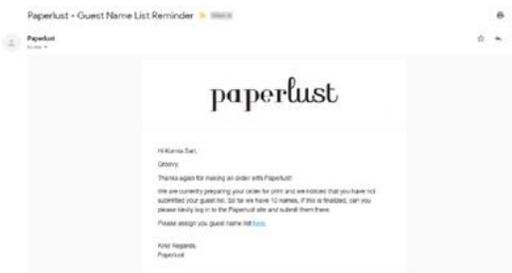
Pada awalnya, fungsi yang menangani peran ini hanya menggunakan *mailto()*. Dikarenakan terlalu banyak kasus seperti ini terjadi dan terkesan menulis ulang hal yang sama, maka dari itu fungsi ini diperbaharui dan menggunakan Mandrill.

Mandrill sendiri adalah API *email* transaksional untuk pengguna MailChimp. Mandrill memungkinkan pengguna untuk mengirim email berbasis data dengan cepat berdasarkan template yang sudah disusun

sebelumnya [8]. Selain itu setelah admin berhasil mengirimkan email pengingat tersebut, admin akan menerima nilai kembali berupa waktu email terkirim ke customer. Sehingga kecil kemungkinan akan terkirim email berkali – kali dalam waktu berdekatan yang justru akan menimbulkan email dianggap *spam*. Alasan lain kenapa menggunakan Mandrill adalah sistem pada Mandrill dapat memastikan bahwa *email* yang dikirimkan tidak dianggap sebagai spam yang memungkinkan email justru tidak akan dibaca oleh customer. Hasil dan format button dapat dilihat pada Gambar 5 sedangkan kerangka email yang akan dikirimkan ke customer adalah seperti pada Gambar 6.



Gambar 5 Bagian tombol pengiriman *email* & keterangan lain pada halaman *design orders* di halaman *back-end*



Gambar 6 Kerangka *email* yang diterima oleh *customer*

E. Integrasi One Tree Planted

Sebagai sebuah perusahaan yang bergerak di industri percetakan, Paperlust berkomitmen menemukan cara untuk memberikan produk terbaik dan mempromosikan kesadaran di bidang industri, terutama penyelenggaraan acara, secara bersamaan. Namun sayangnya tidak semua perusahaan di industri ini menyadari permasalahan proses produksi kertas yang menjadi salah satu dari empat pendorong utama deforestasi di seluruh dunia. Oleh karena itu, Paperlust tidak hanya ingin memastikan bahwa sumber kertas mereka berkualitas, tetapi juga turut berkontribusi melalui program reboisasi bersama One Tree Planted.

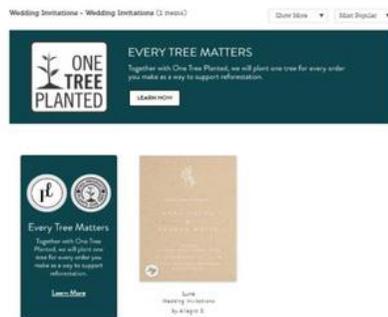
One Tree Planted adalah organisasi nirlaba yang bertujuan memulihkan kawasan hutan yang hancur akibat kebakaran serta berbagai industri seperti minyak

sawit. Misi One Tree Planted adalah memudahkan orang-orang untuk membantu menjaga lingkungan dengan cara menanam pohon dalam lingkup global.

Paperlust mengkampanyekan gerakan ini melalui *banner* yang dipasang di halaman *browse* desain dan halaman informasi khusus.



Gambar 7 *Tooltip* informasi di halaman Paperlust



Gambar 8 *Banner* One Tree Planted pada halaman *browse design* Paperlust



Gambar 9 Halaman informasi One Tree Planted

Banner seperti yang ditunjukkan pada **Gambar** diatur hanya muncul ketika memenuhi aturan momen tertentu. Model ini menerapkan prinsip *A/B testing* yang mempunyai tujuan untuk membandingkan dua model pada momen yang tertentu untuk melihat apakah setiap perubahan komponen membawa peningkatan terhadap persentase tindakan pengunjung atau justru sebaliknya.

V. KESIMPULAN

Berdasarkan uraian yang telah dikemukakan maka dapat disimpulkan bahwa proses pengembangan menggunakan kerangka kerja *scrum* sangat tepat bila diterapkan di Paperlust. Fleksibilitas dan kecepatan eksekusi dari setiap pemecahan masalah terlaksana dengan sangat efektif dan efisien. Adanya kolaborasi sebagai tim yang kuat dan proses iterasi pengembangan produk sebagai tumpuan, membuat hasil akhir mempunyai nilai yang tinggi. Disamping itu, pendekatan bertahap dan berkelanjutan yang diterapkan di *scrum* mampu mengoptimalkan kemampuan prediksi dan mengendalikan risiko [1].

REFERENSI

- [1] R. S. Pressman, dalam *Rekayasa Perangkat Lunak: Pendekatan Praktisi Edisi 7*, Yogyakarta, ANDI, 2012.
- [2] S. Ken dan S. Jeff, *Panduan Scrum*, 2017.
- [3] A. I. Agrawal S, "Mobile Application Development: A Developer Survey," 2010.
- [4] R. C. Martin, dalam *Agile Software Development: Principles, Patterns, and Practice*, Upper Saddle River, Pearson Education, 2003.
- [5] S. L. Pfleeger dan J. M. Atlee, dalam *Software Engineering: Theory and Practice*, Upper Saddle River, Pearson Prentice Hall, 2006.
- [6] A. Pham, dalam *Scrum in Action Agile Software Project Management and Development*, Boston Mass, Course Technology PTR, 2011.
- [7] R. S. Kenett dan E. Baker, *Process Improvement and CMMI® for Systems and Software*, Francis: CRC Press, 2010.
- [8] "Transactional Email for MailChimp Users," MailChimp: The Rocket Science Group, 2018. [Online]. Available: <https://mandrill.com/>. [Diakses 19 November 2019].

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

ORIGINALITY REPORT

11%

SIMILARITY INDEX

9%

INTERNET SOURCES

3%

PUBLICATIONS

6%

STUDENT PAPERS

PRIMARY SOURCES

1	ejournal.poltektegal.ac.id Internet Source	3%
2	Adi, Putu. "Scrum Method Implementation in a Software Development Project Management", International Journal of Advanced Computer Science and Applications, 2015. Publication	2%
3	scrum.or.id Internet Source	1%
4	luthfinaufal18.blog.upi.edu Internet Source	1%
5	docobook.com Internet Source	1%
6	Oscar Franco-Bedoya, Dolores Costal, Soraya Hidalgo, Ron Ben-Jacob. "Expert Mining for Evaluating Risk Indicators Scenarios", 2014 IEEE 38th International Computer Software and Applications Conference Workshops, 2014 Publication	1%

7

Submitted to Forum Komunikasi Perpustakaan
Perguruan Tinggi Kristen Indonesia (FKPPTKI)

Student Paper

1%

8

Submitted to Universitas Brawijaya

Student Paper

1%

9

journal-isi.org

Internet Source

1%

10

id.123dok.com

Internet Source

<1%

Exclude quotes Off

Exclude matches Off

Exclude bibliography On