

OPTIMALISASI GITHUB UNTUK SOFTWARE PROJECT MANAGEMENT DENGAN MEMANFAATKAN NOTIFIKASI SMS

Syarif Hidayatulloh

Fakultas Teknik, Universitas BSI

Jalan Sekolah Internasional No.1-6 Antapani Bandung 40282

syarif.sfq@bsi.ac.id

Abstract - software project management is the art and science of planning and supervision in software projects. On project management software many emerging constraints that could result in his development and software development. One of the key points in a project the software repository is the time it takes to comment on, add and merge source code. Rapid feedback to make the members of a software project team was pleased to contribute in a software project. Problems on Github repository is when the notification was delivered via the web and email are not directly addressed because the members of the project management software rarely check emails. The methods used in this study is a literature study and experiments of different cases in the article, book or paper that discusses how to implement project management software quality so the software project goals can be achieved. Conclusion the results of the research that has been done is to apply the SMS notification on Github is expected to further speed up the interaction and communication between members in a software project and makes it easier for an IT manager in conducting management on Github project.

Keywords: Software project management, GitHub, notification, SMS

Abstrak - Manajemen proyek Perangkat lunak adalah seni dan ilmu perencanaan dan pembimbingan dalam proyek perangkat lunak. Pada manajemen proyek perangkat lunak banyak muncul kendala-kendala yang dapat mengakibatkan lamanya pembangunan dan pengembangan perangkat lunak. Salah satu poin penting dalam sebuah repositori proyek perangkat lunak adalah waktu yang dibutuhkan untuk mengomentari, menambahkan dan menggabungkan source code. Feedback yang cepat membuat anggota-anggota tim proyek perangkat lunak merasa senang untuk berkontribusi dalam sebuah proyek perangkat lunak. Permasalahan pada repositori Github adalah ketika notifikasi yang disampaikan lewat web dan email tidak langsung ditanggapi karena anggota-anggota proyek manajemen perangkat lunak jarang memeriksa email. Metode yang digunakan dalam penelitian ini adalah eksperimen dan studi pustaka dari berbagai kasus dalam artikel, buku atau paper yang membahas bagaimana melaksanakan manajemen proyek perangkat lunak yang berkualitas sehingga tujuan proyek perangkat lunak dapat dicapai. Kesimpulan hasil penelitian yang telah dilakukan adalah dengan menerapkan notifikasi SMS pada Github diharapkan dapat lebih mempercepat interaksi dan komunikasi antar anggota dalam sebuah proyek perangkat lunak serta mempermudah seorang IT manager dalam melakukan manajemen pada Github project.

Keywords: Software project management, GitHub, notifikasi, SMS

PENDAHULUAN

Manajemen proyek Perangkat lunak adalah seni dan ilmu perencanaan dan pembimbingan dalam proyek perangkat lunak (Stellman & Greene, 2006).

Perangkat lunak merupakan produk tidak berwujud sehingga dibutuhkan pengetahuan membuat perangkat lunak yang baik dan efisien dengan pendekatan teknik. Manajemen proyek perangkat lunak merupakan bagian yang penting dalam pembangunan perangkat lunak (Pressman, 2001).

Pada manajemen proyek perangkat lunak banyak muncul kendala-kendala yang dapat mengakibatkan panjangnya lead time yang dibutuhkan untuk menerbitkan sebuah dokumen. Kendala-kendala tersebut adalah sebagai berikut:

- *Delay* dalam hal pengumpulan hasil uji terima dari lapangan yang disebabkan oleh jarak dan pengiriman, yang dapat mengakibatkan terjadinya penumpukan (*bursting*) dokumen di belakang
- Banyaknya dokumen yang harus diproses dalam waktu yang hampir bersamaan.
- Proses dokumentasi masih bersifat manual, baik dalam penyusunan maupun pemeriksaan awal.

Banyaknya lembar dokumen yang harus melalui proses *approval* dan inisiasi pejabat terkait sehingga membutuhkan waktu dan effort yang tidak sedikit. Untuk mengatasi masalah tersebut dibuatlah suatu system yang dapat membuat simplifikasi proses dengan aplikasi berbasis otomatisasi *IT Platform* untuk peningkatan akurasi dan akuntabilitas pada setiap titik proses.

Tool yang akan digunakan dalam hal ini adalah GIT yang diharapkan dapat meminimalisir terjadinya hambatan-hambatan di atas.

Sistem ini akan memonitor dan mengintegrasikan proyek sejak awal proyek sehingga diharapkan dengan adanya system ini akan mempercepat *lead time* dari proses *eksisting* yang berjalan dan akan meningkatkan efisiensi dalam hal penggunaan kertas, tinta printer dan waktu yang dibutuhkan oleh para pejabat berwenang untuk melakukan approval (Rizal, 2014).

Git

Git adalah perangkat lunak pengontrol versi atau proyek manajemen kode perangkat lunak yang diciptakan oleh Linus Torvalds, yang pada awalnya ditujukan untuk pengembangan kernel Linux. Desain Git terinspirasi oleh BitKeeper dan Monotone. Git pada awalnya hanya dirancang sebagai mesin tingkat rendah yang dapat digunakan oleh tampilan muka (*front end*) lain seperti Cogito atau StGIT. Namun selanjutnya proyek inti Git telah berkembang menjadi pengontrol revisi lengkap yang dapat digunakan langsung. Saat ini, beberapa perangkat lunak terkenal menggunakan Git sebagai pengontrol revisinya, antara lain kernel Linux, Server X.org, pengembangan inti OLPC (*One Laptop per Child*), serta kerangka kerja web Ruby on Rails. Pemeliharaan perangkat lunak Git saat ini diawasi oleh Junio Hamano. Dirilis di bawah Lisensi Publik Umum GNU versi 2, Git adalah suatu perangkat lunak bebas (Wikipedia, 2013).

Github

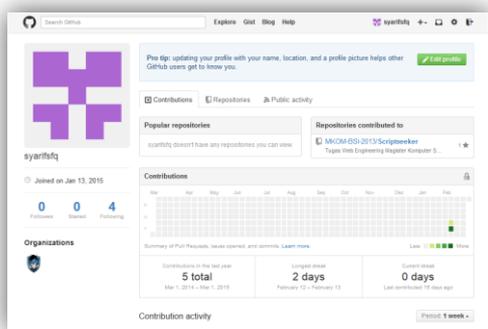
Github adalah *software Hosting* untuk proyek *open source* yang menggunakan *Tool System* revisi kontrol Git. Jadi Git adalah *tool* untuk melakukan revisi code, sedangkan github adalah *webhostingnya*. Mudah-mudahan Github adalah *Webhosting* untuk proyek-proyek *software* seperti Google code atau sourceforge.net. Dalam bahasa gaul, Github adalah jejaring sosial untuk *software developer* (Putra, 2014).

Yang membedakan Github dari *Project hosting* yang lain bisa kita lihat di bawah ini :

1. Github menggunakan Git sebagai *tool* utama dalam merevisi code.
2. Github menyediakan *free hosting* untuk proyek *open source*, namun Github juga menawarkan *hosting* berbayar untuk proyek perusahaan atau pribadi yang sifatnya privat.
3. Github ditargetkan untuk para *developer* yang bekerja secara tim dan tidak disatu tempat.
4. Github mempunyai fitur Social seperti Twitter seperti follow atau favorit (*Star*).
5. Github mendukung semua bahasa pemrograman, anda bebas memakai bahasa dan *tool* yang biasa anda pakai.

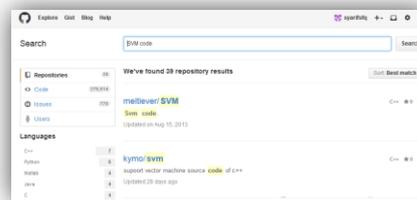
Beberapa fitur Sosial di Github:

1. Github *user profile*.
2. Follow, Kita bisa memfollow profil *programmer* lain di github.
3. Star, fungsinya sama dengan *Bookmark*.
4. Watch, memonitor *repository* tertentu. Setiap ada perubahan di Repo, maka kita akan dikirim notifikasi.
5. Fork, mengcopy keseluruhan *code repository* ke repo kita sendiri.



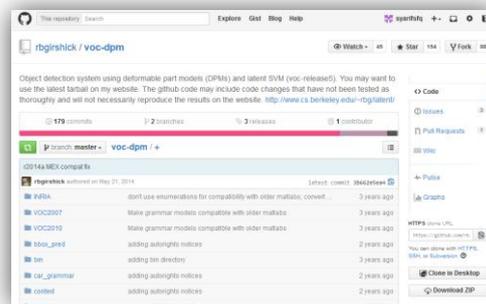
Gambar 1
Github Profile

Sama seperti Twitter, Github juga mempunyai profil. Profil berisi foto, jumlah repo yang dimiliki, jumlah *follower*, jumlah *user* yang kita *follow* dan *Star* (jumlah proyek yang kita beri tanda bintang).



Gambar 2
Fitur Pencarian

Github mempunyai lebih dari 10 juta repo *software*. Jika anda ingin mencari proyek *open source* tertentu, gunakan fitur pencarian, hasil pencarian tampak seperti di gambar 2 diatas. Untuk memperkecil hasil pencarian bisa di filter berdasarkan Repo, *code*, *issue* atau nama *user* atau bahasa pemrograman. Hasil pencarian bisa diurutkan berdasarkan jumlah bintang tau jumlah "*fork*". Biasanya makin banyak bintangnya, makin berkualitas *source codenya*.



Gambar 3
Struktur Halaman proyek di Github

Contoh di gambar 3 adalah struktur proyek dari *Object detection system using deformable part models (DPMs) and latent SVM (voc-release5)*. Mari kita analisa satu persatu struktur halamannya.

Bagian Statistik

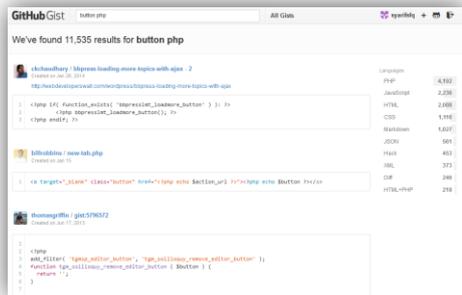
Bagian pertama adalah statistik, kita bisa melihat jumlah Watch, star dan fork dibagian atas halaman.

Bagian Code

Bagian utama dari struktur proyek ini adalah *code* itu sendiri. Kode ini juga berisi penjelasan kapan dan apa file terakhir yang diedit beserta penjelasan perubahan apa yang terjadi.

Bagian Clone dan Download

Clone in desktop di pakai untuk mengkopi *source code* proyek tersebut ke komputer kita, namun komputer kita harus sudah terinstall Github for Windows atau Mac. Bagian paling penting dan paling sering dipakai adalah menu DOWNLOAD. jika anda ingin langsung mencoba *code* di komputer, cukup klik download, maka *source code* otomatis akan di-zip dan diunduh ke komputer anda.



Gambar 4
Contoh pencarian snippet code di Github Gist

Github Gist adalah layanan *share snippet* dari github. Biasanya hanya potongan *code* pendek untuk memecahkan solusi di bahasa pemrograman tertentu. Sebagai contoh, anda ingin tahu bagaimana cara membuat tombol di php, anda cukup buka <https://gist.github.com>. lalu masukan *query* “button php”.

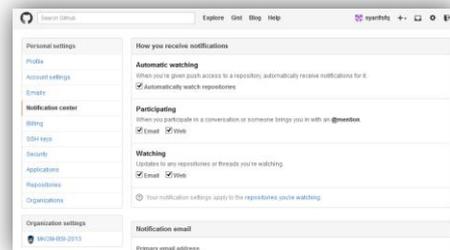
METODE PENELITIAN

Metode yang digunakan dalam penelitian ini adalah eksperimen dan studi pustaka dari berbagai kasus dalam artikel, buku atau paper yang membahas bagaimana melaksanakan manajemen proyek perangkat lunak yang berkualitas sesuai dengan tujuan yang ditetapkan sehingga menghasilkan analisis deskriptif untuk menentukan masalah. Dan pendekatan pada penelitian ini adalah pendekatan secara general empiris berdasarkan pengalaman dan teori-teori yang berkaitan dengan manajemen proyek khususnya pada proyek perangkat lunak (Primawati & Mustari, 2013).

PEMBAHASAN

Pengembangan perangkat lunak merupakan suatu yang penuh dengan tantangan. Sistem dibangun dengan kolaborasi antar anggota tim. Satu kunci penting bagi *developer* dalam kolaborasi membangun aplikasi adalah sesegera mungkin membuat comment dan menggabungkan permintaan ke dalam *project* pada Github Project.

Notifikasi *Feedback* dalam project sangat penting bagi *developer* atau *IT manager* namun sayangnya seringkali notifikasi yang dikirim lewat email terlewatkan karena jarang memeriksa *email*.



Gambar 5
Settings notifications

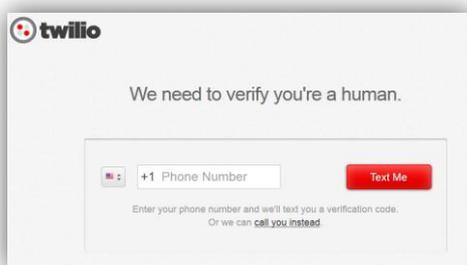
Dalam setting notifikasi Github hanya terdapat *option* pilihan email dan web. Dengan mengkombinasikan Zapier dengan Twilio kita dapat membuat

notifikasi SMS untuk Github tanpa harus menulis *source code* sendiri.

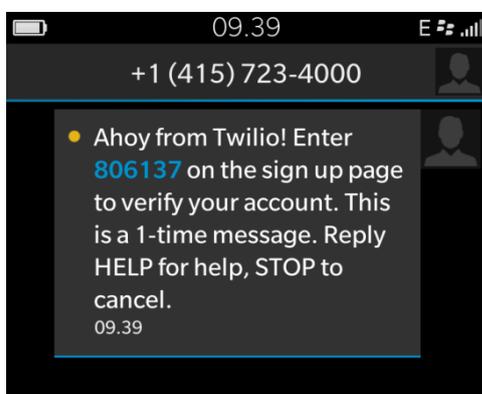
Daftar Twilio untuk mengirim SMS



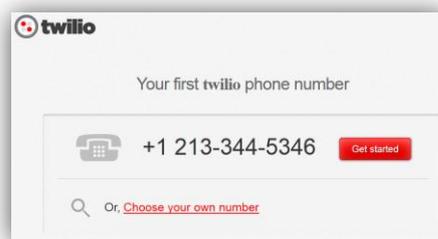
Gambar 6
Daftar Twilio



Gambar 7
Verifikasi Nomor Telepon



Gambar 8
Pesan verifikasi SMS dari Twilio



Gambar 9
Nomor telepon Twilio kita

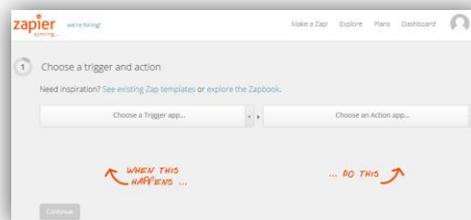
Twilio akan memberikan nomor telepon secara *random* setelah proses pendaftaran selesai.

Setting Zap dengan Zapier

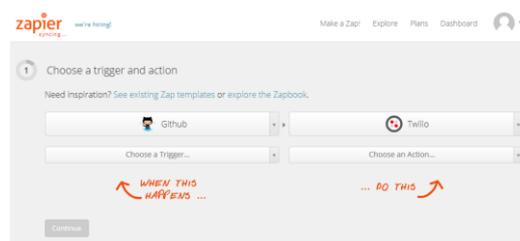


Gambar 10
Membuat Zap

Tekan tombol make new zap untuk membuat zap.



Gambar 11
Memilih trigger and action

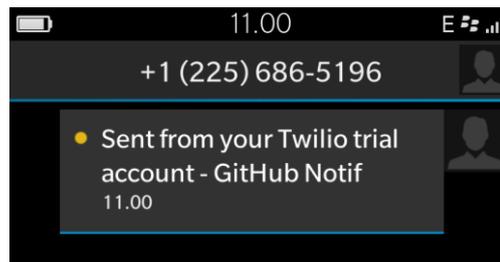


Gambar 12
Memilih trigger dan action

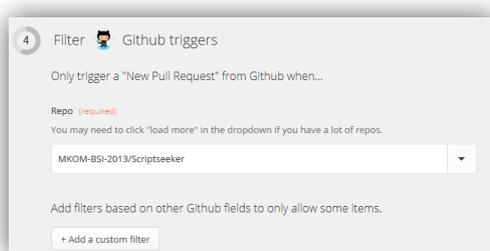


Gambar 13
Zapier berhasil terkoneksi dengan
Github dan Twilio

tambahan namun tidak lebih dari 160 karakter.



Gambar 16
Hasil notifikasi SMS

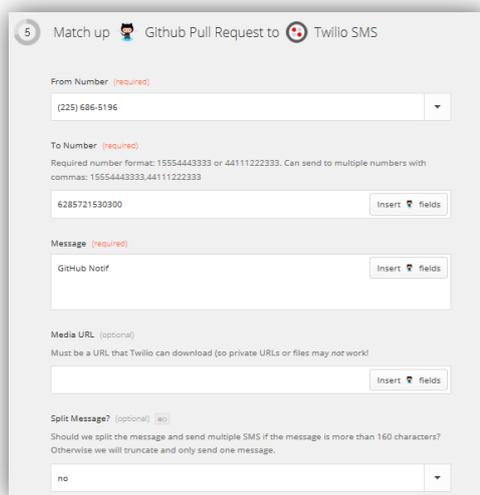


Gambar 14
Menentukan repositori Github
project

KESIMPULAN

Berdasarkan penelitian yang telah dilakukan, maka dapat diambil beberapa kesimpulan sebagai berikut.

1. Fitur Notifikasi pada Github kurang variatif sehingga perlu adanya penambahan fitur.
2. Mempermudah seorang IT manager dalam melakukan manajemen pada Github project.
3. Dengan menerapkan notifikasi SMS pada Github diharapkan dapat lebih mempercepat interaksi dan komunikasi antar anggota dalam sebuah *Project*.



Gambar 15
Menambahkan pesan tambahan pada
notifikasi SMS

REFERENSI

Pressman, R. S. (2001). *Software Engineering : A Practioner's Approach.* 5th. McGrawHill.

Primawati, A., & Mustari, D. (2013). ANALISIS MANAJEMEN PROYEK PERANGKAT LUNAK SISTEM INFORMASI AKUNTANSI PADA BIRO SISTEM INFORMASI PT. X . *Journal of Information Systems Vol.9, Issue.2*, 101-106.

Putra, C. A. (2014, Januari 25). *Pengenalan Github untuk pemula.* Retrieved from candra.web.id: <http://www.candra.web.id/2014/>

Terakhir, pilih nomor telepon Twilio untuk mengirim notifikasi. Kita juga dapat menambahkan pesan

01/25/pengenalan-github-untuk-pemula/

Rizal, A. (2014). Analisis Penerapan Project Management Information System (PMIS) Menggunakan Metode Technology Acceptance Model (TAM) Studi Kasus PT. INDOSAT, Tbk . *IncomTech Vol.5, No.1 ISSN 2085-4811*, 1-24.

Stellman, A., & Greene, J. (2006). *Applied Software Project Management*. california: O'Reilly Media, Inc.

Wikipedia. (2013, April 6). *Git*. Retrieved from wikipedia: <http://id.wikipedia.org/wiki/Git>