

Perancangan Dashboard Sistem Informasi Untuk Agile Manajemen Proyek dengan Menggunakan JIRA – Studi Kasus di PT. FLASHiZ Indonesia

Prawido Utomo¹, Setiawan², Franky Wibisono Prayitno³

^{1,2}Dosen STMIK Bina Sarana Global, ³Mahasiswa STMIK Bina Sarana Global

Email : ¹prawidoutomo@stmikglobal.ac.id, ²setiawan@stmikglobal.ac.id, ³greenwibisono@gmail.com

Abstrak— Seiring dengan pesatnya kemajuan teknologi internet dan kemudahan dalam akses untuk penggunaan teknologi tersebut saat ini, memunculkan gagasan untuk pemanfaatannya dalam membantu manajer proyek yang dalam hal mewakili pemilik proyek (owner) agar bisa melakukan pengawasan dan pengendalian terhadap proyek-proyek yang berada dibawah naungannya. Dalam hal waktu kerja, manajer proyek mempunyai keterbatasan, terutama untuk melakukan pengawasan dan pengendalian dikarenakan proyek yang tangani seringkali lebih dari 1 (satu), yang waktu kegiatannya berjalan secara bersamaan. Agar sistem informasi yang ada dapat disimpan dan mudah diakses kapan saja oleh pemilik proyek (owner), maka diperlukan sebuah web server yang dapat menampung data dan informasi yang telah dikerjakan. Informasi dalam database yang ada selain diperuntukkan bagi pemilik proyek (owner) juga dapat diakses oleh pihak-pihak lainnya yang memiliki kepentingan tertentu dalam proyek. Tersedianya sistem laporan kinerja proyek dan informasi status pengerjaan proyek yang terintegrasi, dapat memudahkan dan mempercepat proses pelaporan kepada pemilik proyek (owner). Pengembangan sistem informasi manajemen proyek terbaru dengan menggunakan Metode Agile, yang merupakan salah satu metode didalam proyek manajemen yang banyak digunakan khususnya bagi perusahaan-perusahaan yang bergerak dibidang pengembangan perangkat lunak. Salah satu aplikasi sudah yang banyak dipakai adalah JIRA. Aplikasi ini bisa memberikan data atau informasi secara real-time melalui halaman Dashboard yang disediakan oleh pengelola. Semua history kegiatan baik masalah dan solusi selama pelaksanaan proyek terekam dalam aplikasi ini melalui laporan-laporan yang disediakan, sehingga dapat menjadi acuan bagi perencanaan dan pelaksanaan proyek-proyek yang serupa nantinya.

Kata kunci— Sistem Informasi Manajemen, Manajemen Proyek, Metode Agile, Pengawasan dan Pengendalian, JIRA.

I. PENDAHULUAN

A. Latar Belakang

Pada era globalisasi saat ini, kemajuan teknologi sudah semakin meningkat dan semakin memudahkan segala aspek kegiatan dalam kehidupan manusia. Kemajuan teknologi saat ini merupakan salah satu bentuk dari hasil karya manusia yang semakin berkembang, sesuai dengan tuntutan zaman. Hasil dari kemajuan teknologi dapat membantu mengatasi segala masalah yang ada, sehingga manusia mempunyai pola pikir yang maju. Semakin berkembangnya ilmu pengetahuan dan teknologi saat ini, maka dapat membantu mempertahankan kelangsungan hidup suatu perusahaan. Mengingat semakin ketatnya persaingan di sektor kerja, untuk itu diperlukan alat-

alat yang dapat membantu pihak manajemen perusahaan agar dapat melaksanakan tugasnya dengan baik sehingga efisiensi kerja dapat tercapai. Penggunaan teknologi informasi berbasis web merupakan salah satu cara untuk membantu pihak manajemen perusahaan, karena lebih efisien dan cepat. Karena dengan berbasis web, data yang diperlukan bisa diakses dari mana saja, dengan apa saja apabila terhubung dengan internet.

Selain teknologi informasi, pentingnya manajemen suatu proyek tidak boleh dipandang sebelah mata. Hal ini disebabkan karena dalam melakukan proses produksi, perusahaan harus mampu menghasilkan suatu produk baik berupa barang ataupun jasa yang sesuai dengan kriteria, waktu, dan biaya yang telah ditetapkan. Perubahan terhadap salah satu dari ketiga faktor tersebut dapat mempengaruhi dua faktor lainnya yang tentunya juga akan berdampak pada besar kecilnya nilai keuntungan yang dapat diperoleh perusahaan.

Pada saat ini, sistem pengelolaan proyek di PT. FLASHiZ Indonesia khususnya di bagian DIMO Payment bisa dibilang belum maksimal. PT. FLASHiZ Indonesia merupakan salah satu perusahaan yang bergerak di bidang pengembangan uang digital khususnya dengan sistem pembayaran dengan menggunakan *Quick Response (QR) Code*, dengan slogan “*The preferred way to pay*”, PT. FLASHiZ Indonesia mempunyai tujuan untuk menjadi pionir dalam sistem pembayaran dengan metode *Scan QR Code*.

Berdasarkan analisa yang ada, maka penulis mencoba untuk membuat suatu sistem yang mempermudah pengawasan dari pengembangan suatu produk di PT. FLASHiZ Indonesia dari awal hingga akhir.

II. LANDASAN TEORI

A. Scrum

Penggunaan metode Scrum untuk pengembangan perangkat lunak menandai perubahan yang cukup dramatis dari metode manajemen yang bersifat *waterfall*. Scrum dan metode lain ini terinspirasi dari kekurangan-kekurangan yang dimiliki oleh metode yang bersifat *waterfall*. Scrum menekankan pada kolaborasi, software yang berfungsi dengan baik, manajemen tim yang baik (*self-management*), dan kemampuan beradaptasi terhadap perubahan yang ada sesuai dengan realitas bisnis yang muncul.

1. Metode Scrum

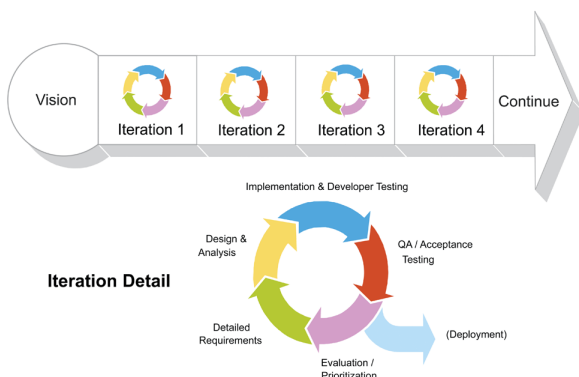
Scrum merupakan bagian dari Agile yang merupakan

respon terhadap kegagalan dari manajemen proyek bersifat waterfall. Pada tahun 2001, 17 pelopor metode yang serupa bertemu di Resort Snowbird Ski di Utah dan menulis Agile Manifesto, deklarasi empat nilai dan dua belas prinsip dari Agile. Nilai-nilai dan prinsip-prinsip berdiri dengan kontras dengan tradisional *Project Manager's Body of Knowledge* (PMBOK).

2. Scrum Roles

Scrum memiliki tiga peranan penting: *Product Owner*, *Scrum Master*, dan *Development Team*.

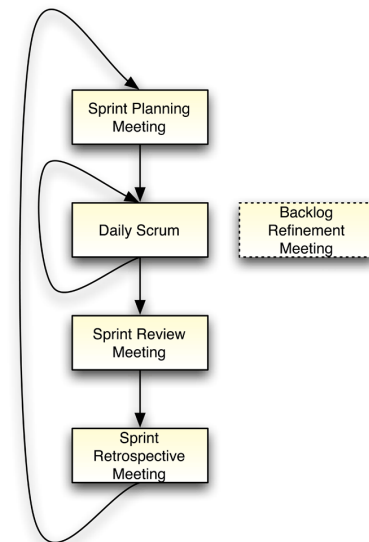
- a. *Product Owner*, seseorang yang memiliki visi, otoritas, dan ketersediaan waktu. Pemilik produk bertanggung jawab untuk terus mengkomunikasikan visi dan prioritas untuk tim pengembangan.
- b. *Scrum Master*, seseorang yang bertindak sebagai fasilitator untuk pemilik produk dan tim pengembangan yang terdiri dari *Developer* dan *Tester* (*Quality Assurance*). Scrum Master tidak bertanggung jawab dengan pengelolaan tim. Fungsi dari Scrum Master untuk menghilangkan hambatan yang menghalangi tim dari mencapai tujuannya. Hal ini membantu tim tetap kreatif dan produktif sambil memastikan keberhasilan yang terlihat ke pemilik produk. Scrum Master juga bekerja untuk memberikan nasihat kepada pemilik produk tentang bagaimana memaksimalkan *Return On Investment* (ROI) untuk tim.
- c. *Scrum Development Team*, menurut pendiri Scrum “*the team is utterly self managing*”. Tim pengembangan bertanggung jawab untuk mengatur diri untuk menyelesaikan pekerjaannya. Sebuah tim pengembangan Scrum memiliki sekitar tujuh anggota yang penuh dedikasi (resminya tiga sampai sembilan orang), ideanya dalam satu ruangan tim dilindungi dari gangguan dari luar. Untuk proyek pengembangan perangkat lunak, tim umum meliputi insiyur perangkat lunak (*software engineers*), arsitek (*architects*), programmer, analis (*analyst*), ahli QA (*Quality Assurance*), penguji (*testers*), dan UI *designers*. Setiap sprint, tim bertanggung jawab untuk menentukan bagaimana ia akan menyelesaikan pekerjaan yang harus diselesaikan. Tim memiliki otonomi dan tanggung jawab untuk memenuhi tujuan dari sprint.



Gambar 1. Scrum

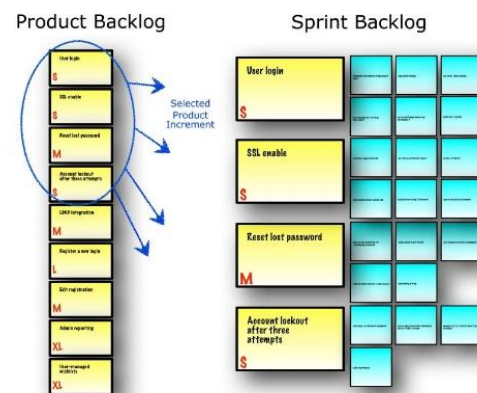
3. Scrum Meetings

Scrum memiliki beberapa rapat-rapat didalam prosesnya. Semua rapat di fasilitasi oleh Scrum Master, yang tidak memiliki otoritas pengambilan keputusan pada setiap rapat/pertemuan.



Gambar 2. Scrum Flow

- a. *Sprint Planning Meeting*, setiap awal Sprint, pemilik produk dan tim akan mengadakan rapat untuk menegosiasikan *Product Backlog Item* mana saja yang akan diusahakan untuk dikerjakan selama sprint. Pemilik produk bertanggung jawab untuk memberitahukan item mana yang paling penting untuk bisnis. Menjelang akhir dari rapat, tim akan membagi item yang terpilih ke daftar *Sprint Tasks*, dan membuat komitmen final untuk melakukan pekerjaannya. Maksimum waktu yang diberikan (*timebox*) untuk merencanakan Sprint adalah 14 – 30 hari.



Gambar 3. Sprint Planning Meeting

- b. *Daily Scrum* (*Sprint Execution*), setiap hari pada waktu dan tempat yang sama, anggota *Scrum Development Team* akan menghabiskan total 15 menit untuk melaporkan hasil pekerjaannya. Setiap anggota tim

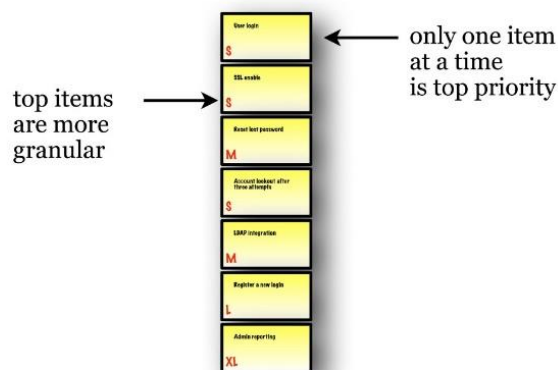
melaporkan apa saja yang dikerjakan dihari sebelumnya, apa yang akan dilakukan hari ini, dan hambatan apa saja yang dihadapi.

- c. *Sprint Review Meeting*, rapat ini dilakukan untuk menunjukkan (demonstrasi) fungsi dari produk yang sudah selesai dikerjakan kepada pemilik produk dan orang lain yang tertarik dengan produk. Pertemuan ini harus dilakukan secara langsung (demonstrasi langsung), bukan laporan. Setelah demonstrasi, pemilik produk akan memberikan komentar dan memberikan keputusan item mana yang dianggap benar-benar telah selesai dikerjakan. *Scrum Master* akan membantu pemilik produk dan stakeholder untuk mengkonversi *feedback* yang mereka berikan menjadi *Product Backlog Item* yang baru.
- d. *Sprint Retrospective Meeting*, setiap *Sprint* berakhir dengan retrospektif. Pada pertemuan ini, tim melihat kembali proses yang sudah dilalui. Mereka memeriksa perilaku mereka dan mengambil tindakan untuk beradaptasi untuk *Sprint* yang akan datang.
- e. *Backlog Refinement Meeting*, kebanyakan *Product Backlog Item* (PBI) awalnya perlu perbaikan karena mereka terlalu besar dan kurang dipahami. Karena hal ini, tim mengambil waktu sedikit dari *Sprint Execution* untuk membantu perencanaan *Sprint* berikutnya.

4. *Scrum Artifacts*

a. *Product Backlog*

1. Menentukan peringkat dari fungsi yang diinginkan.
2. Bisa dilihat dan ditambahkan oleh semua stakeholder (termasuk tim).
3. Pemilik produk menentukan prioritas kerja setiap saat.
4. Item yang berada dipaling atas leih rinci dari yang bawah.
5. Dipertahankan selama *Backlog Refinement Meeting*.



Gambar 4. *Product Backlog*

b. *Sprint Backlog*

1. Terdiri dari item yang sudah dinegosiasikan antara tim dan pemilik produk.

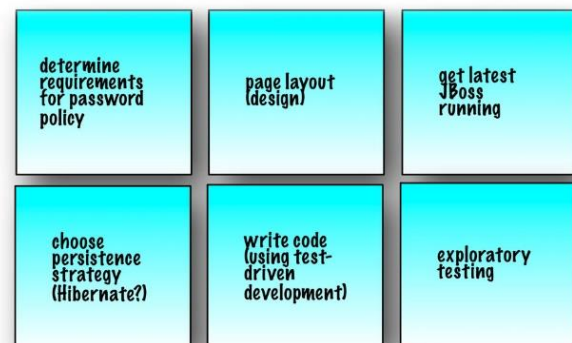
2. Lingkup komitmen tidak berubah selama *Sprint Execution*.
3. Semua tugas awal sudah diidentifikasi selama *Sprint Planning Meeting*.
4. Tim akan menemukan tugas tambahan yang diperlukan untuk memenuhi komitmen yang ada.
5. Semua anggota tim bisa melihat tugas-tugas apa saja yang ada dalam *Sprint Execution*.

Committed Backlog Items	Tasks Not Started	Tasks In Progress	Tasks Completed

Gambar 5. *Sprint Backlog*

c. *Sprint Task*

1. Menentukan bagaimana cara untuk memenuhi/mencapai *Product Backlog Item* (PBI).
2. Membutuhkan satu hari atau kurang dari satu hari kerja.
3. Sisa dari upaya (waktu) pengerjaan akan diestimasi ulang setiap harinya, biasanya dalam bentuk jam.
4. Selama *Sprint Execution*, anggota tim bisa menjadi relawan untuk jadi penanggung jawab atas sebuah tugas.
5. *Sprint Task* menjadi tanggung jawab seluruh anggota tim, jadi seluruh anggota tim harus bisa berkolaborasi.



Gambar 6. *Sprint Task*

B. Atlassian JIRA

1. Konsep JIRA

a. Proyek (*Project*), sebuah project di JIRA adalah sebuah kumpulan dari isu/tugas, yang didefinisikan sesuai dengan keperluan perusahaan. Project di JIRA bisa berupa:

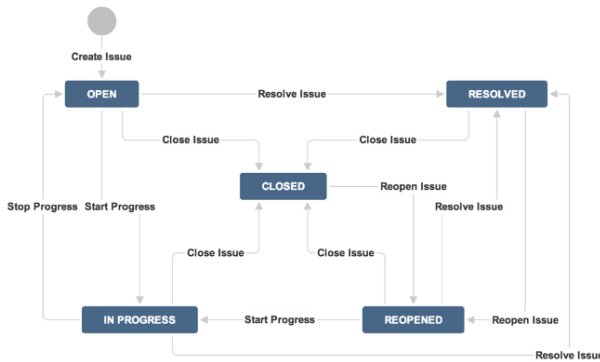
1. Proyek pengembangan perangkat lunak (*software*).
2. Kampanye marketing.
3. Sistem *helpdesk*.
4. Sistem manajemen cuti.
5. Sistem permintaan penambahan fungsi website.

Setiap isu memiliki proyek, setiap proyek memiliki nama (misal: *Website Issues*) dan kunci (misal: *WEB*). Kunci dari proyek menjadi bagian pertama dari isu (misal: *WEB-101, WEB-102*).



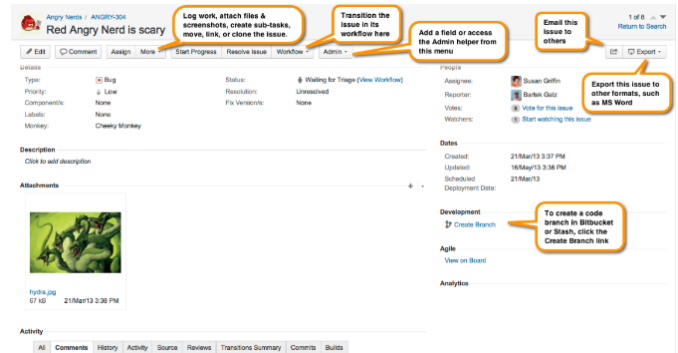
Gambar 7. JIRA Project

b. Alur Kerja (*Workflow*), adalah sekumpulan Status (*Statuses*) dan Transisi (*Transitions*) yang akan dilalui oleh sebuah isu dalam siklus hidup pengembangannya.



Gambar 8. JIRA Workflow

c. Isu/Tugas (*Issue*), setiap organisasi yang berbeda menggunakan JIRA untuk mengawasi berbagai jenis isu. Tergantung pada bagaimana organisasi menggunakan JIRA, isu bisa menjadi *software bug*, tugas proyek, tiket *helpdesk*, formulir permintaan cuti, dan sebagainya.



Gambar 9. JIRA Issue

2. Kebutuhan Untuk Instalasi JIRA

Kebutuhan perangkat lunak (*software*) untuk JIRA *Client* dan *Server*

- a. Internet Browser.
- b. Java Platform (*Java Developers Kit (JDK)/Java Runtime Environment (JRE)*).
- c. Operating Systems.
- d. *Application Server* (jika menggunakan JIRA WAR).
- e. Basis Data (*Database*).
- f. Kebutuhan perangkat keras (*hardware*) untuk JIRA *Server*.

III. ANALISA SISTEM YANG BERJALAN

A. Tata Laksana Sistem yang Berjalan

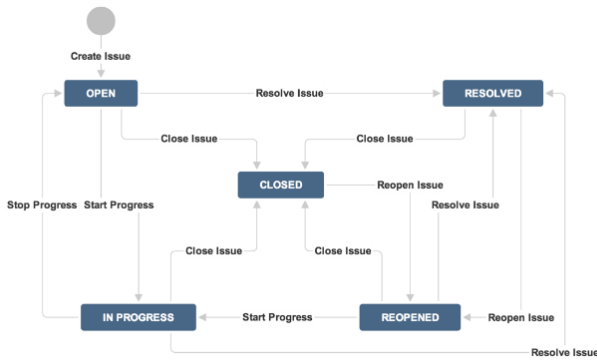
Sistem yang berjalan, tim produk akan meminta tim pengembang untuk membuat suatu produk berdasarkan *requirement* yang dibuat. *Requirement* tersebut dibuat dalam bentuk dokumen bernama *Functional Specification Document (FSD)*. Berdasarkan dokumen tersebut, maka tim pengembang akan memulai membangun produk. Permasalahannya, terkadang ada permintaan pengembangan produk baru tanpa dokumentasi yang jelas, semua dilakukan secara manual.

Pengaturan jadwal kerja dan tugas masing-masing anggota tim, dilakukan secara langsung setiap minggunya, tanpa rencana detil, dengan menggunakan metode agile. Tiap minggu akan diadakan rapat untuk membahas hasil kinerja dari proyek, apa saja yang sudah dikerjakan, apa saja yang belum dikerjakan, masalah-masalah apa saja yang ditemukan selama pengembangan, dan rencana kerja berikutnya.

Setelah produk selesai dibangun, akan dilakukan pengetesan oleh tim *Quality Assurance* untuk memastikan tidak akan ada masalah ketika produk diluncurkan ke pasaran. Sebelum produk diluncurkan, anggota dari tim implementasi project akan memastikan bahwa produk bisa diintegrasikan ke sisi pengguna, misal pemilik POS (*Point of Sale*) sistem dan e-commerce. Pengerjaan produk akan selalu dilakukan setiap kali ada permintaan perubahan dari pihak-pihak yang terkait didalamnya.

Hasil dari produk pengembangan ini bisa diintegrasikan dengan produk-produk yang berhubungan dengan *mobile banking* dan e-money.

1. JIRA Workflow yang Berjalan

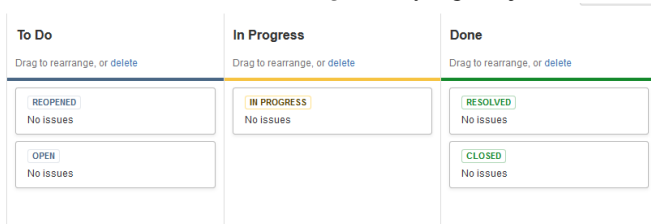


Gambar 10. JIRA Workflow yang Berjalan

Berdasarkan gambar diatas:

- JIRA *workflow* adalah sekumpulan dari *Statuses* dan *Transitions* yang akan dilalui oleh *Issue* sesuai dengan siklus hidup yang sudah ada.
- Reporter akan membuka atau membuat *Issue* baru di aplikasi JIRA, sesuai dengan proyeknya.
- Tim pengembangan bisa menentukan untuk memulai proses pengembangan, melakukan tes terhadap *Issue*, atau menutup *Issue*.
- Issue* yang sudah diselesai dikerjakan akan di tes sebelum ditutup.
- Setelah ditutup, *Issue* bisa dibuka kembali jika dibutuhkan.

2. Dashboard Column Management yang Berjalan



Gambar 11. Dashboard Column Management yang Berjalan

Berdasarkan gambar diatas:

- Masing-masing *Statuses* akan dikelompokan kedalam tahapan pengembangan produk yang ada.
- Kolom pertama menandakan hal-hal yang harus dikerjakan.
- Kolom kedua sampai n, menandakan hal-hal yang sedang dikerjakan.
- Kolom terakhir, menandakan hal-hal yang sudah dikerjakan.

B. Masalah Yang Dihadapi

Beberapa masalah yang dihadapi oleh PT. FLASHiZ Indonesia diantaranya adalah:

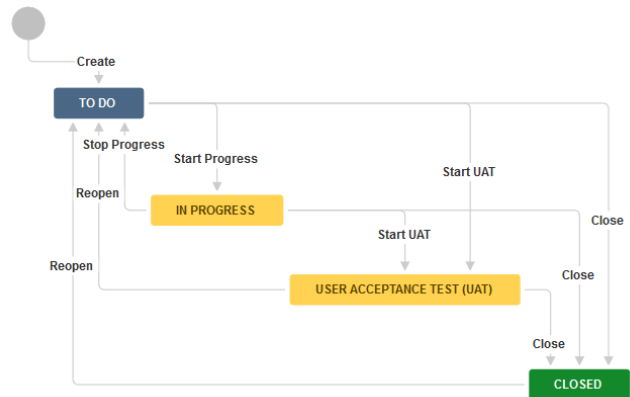
- Tidak adanya proses yang standar dalam pengembangan produk atau manajemen proyeknya.
- Pengembangan yang tidak sesuai spesifikasi, dikarenakan (kadang) tidak adanya spesifikasi dari desain yang digunakan.

- Kurangnya sumber daya manusia untuk pengembangan, akibatnya ada anggota dari tim pengembang bisa memegang beberapa proyek sekaligus tanpa adanya pengawasan terhadap hasil kerjanya.
- Aplikasi JIRA yang ada tidak digunakan sepenuhnya atau kurang maksimalnya penggunaannya. Penggunaan dari aplikasi ini hanya bersifat standar (*built-in*) yang tidak sesuai dengan proses yang berjalan sekarang.

IV. RANCANGAN SISTEM YANG DIUSULKAN

A. Alur Proses (*Workflow*) di JIRA

Alur proses (*workflow*) di JIRA adalah kumpulan *Statuses* dan *Transitions* yang akan dilalui oleh *Issues* selama siklus hidupnya.



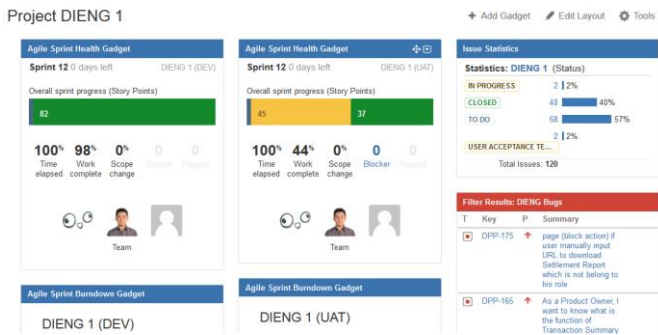
Gambar 12. Alur Proses (*Workflow*) di JIRA

Berdasarkan gambar diatas:

- Workflow* yang diusulkan terdiri dari empat *Statuses* utama yaitu: *TO DO*, *IN PROGRESS*, *USER ACCEPTANCE TEST (UAT)*, dan *CLOSED*.
- Status *TO DO* akan menampung semua *Issues* yang teridentifikasi pada saat atau selama proyek berjalan.
- Status *IN PROGRESS* akan menampung semua *Issues* yang sedang dikerjakan oleh tim pengembangan di Sprint yang sedang berjalan.
- Status *USER ACCEPTANCE TEST (UAT)* akan menampung semua *Issues* yang telah selesai dikerjakan oleh tim pengembang, *Issues* yang siap di tes oleh tim *Quality Assurance (QA)*.
- Status *CLOSED* akan menampung semua *Issues* yang sudah lolos tes di *Quality Assurance (QA)*.

B. Rancangan Project Dashboard

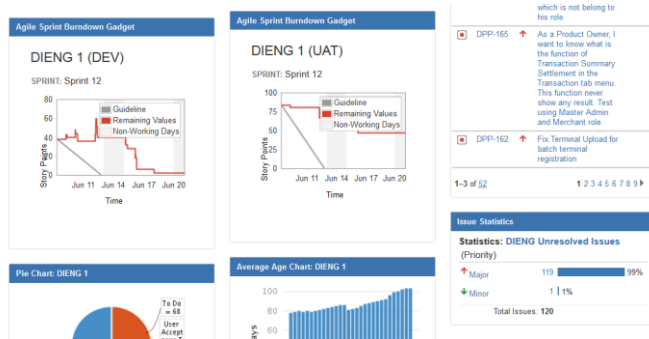
Project Dashboard adalah halaman yang pertama kali akan dilihat oleh *User* pada saat mereka login untuk pertama kali. Berikut rancangan yang dibuat oleh penulis.



Gambar 13. Rancangan Project Dashboard 1

Berdasarkan gambar diatas:

- Agile Sprint Health Gadget* yang pertama akan digunakan untuk mengawasi atau menganalisa status dari kinerja Developer setiap Sprint.
- Agile Sprint Health Gadget* yang kedua akan digunakan untuk mengawasi atau menganalisa status dari kinerja *Quality Assurance* setiap *Sprint*.
- Issue Statistic gadget* akan digunakan untuk menampilkan statistik dari Issue yang ada didalam proyek berdasarkan dari statusnya.
- Filter Results gadget* akan digunakan untuk menampilkan data-data masalah (*bugs*) yang ditemukan pada saat pengerjaan proyek.



Gambar 14. Rancangan Project Dashboard 2

Berdasarkan gambar:

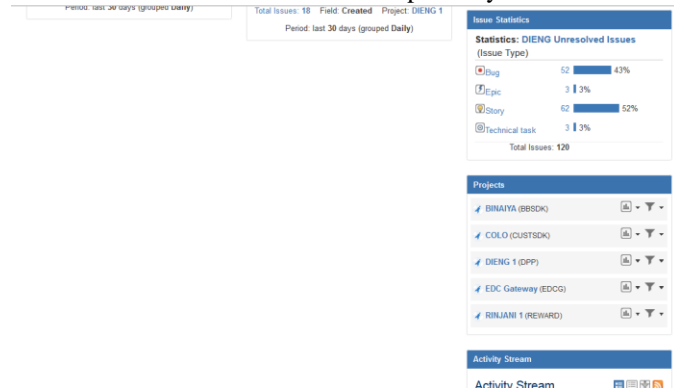
- Agile Sprint Burndown Gadget* yang pertama akan digunakan untuk menganalisa nilai dari Issue yang sudah diselesaikan oleh Developer di setiap *Sprint*.
- Agile Sprint Burndown Gadget* yang pertama akan digunakan untuk menganalisa nilai dari Issue yang sudah diselesaikan oleh *Quality Assurance* di setiap *Sprint*.
- Issue Statistic gadget* akan digunakan untuk menampilkan statistik dari Issue yang belum di selesaikan berdasarkan dari prioritasnya.



Gambar 15. Rancangan Project Dashboard 3

Menurut gambar diatas:

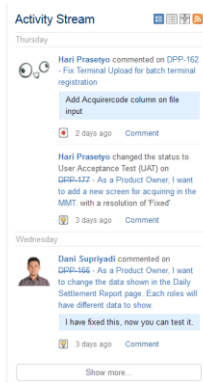
- Pie Chart gadget* akan digunakan untuk menampilkan Issue berdasarkan statusnya dengan menggunakan diagram.
- Average Age Chart gadget* akan digunakan untuk menampilkan berapa lama waktu yang dihabiskan untuk Issue yang belum diselesaikan.
- Created vs. Resolved Chart gadget* akan digunakan untuk menampilkan grafik dari Issue yang dibuat dengan Issue yang sudah diselesaikan.
- Time Since Chart gadget* akan digunakan untuk menampilkan grafik Issue yang dibuat dari awal proyek berjalan.
- Issue Statistic gadget* akan digunakan untuk menampilkan statistik dari Issue yang belum di selesaikan berdasarkan dari Assignee-nya.
- Issue Statistic gadget* akan digunakan untuk menampilkan statistik dari Issue yang belum di selesaikan berdasarkan dari komponennya.



Gambar 16. Rancangan Project Dashboard 4

Menurut gambar diatas:

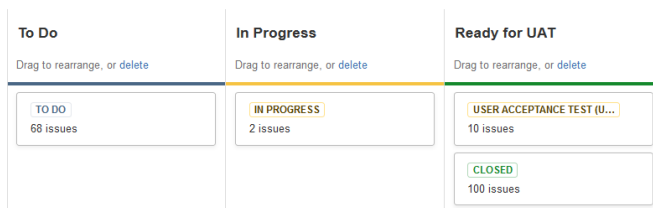
- Issue Statistic gadget* akan digunakan untuk menampilkan statistik dari Issue yang belum di selesaikan berdasarkan dari tipenya.
- Project gadget* akan digunakan untuk menampilkan proyek-proyek apa saja yang terdapat dalam sistem.



Gambar 17 Rancangan Project Dashboard 5

Berdasarkan gambar diatas: *Activity Stream gadget* akan digunakan untuk menampilkan *Issue* yang sedang dikerjakan oleh Developer atau *Quality Assurance*.

C. Rancangan Agile Development Board

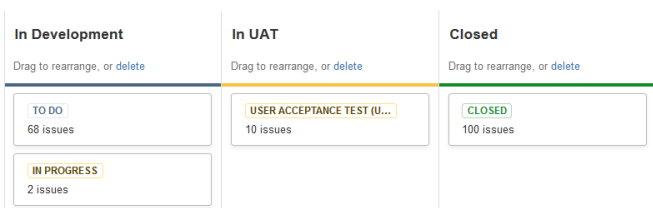


Gambar 18. Rancangan Agile Development Board

Berdasarkan gambar diatas:

- a. Terdapat 3 kolom proses yang akan digunakan oleh Developer.
- b. *To Do*, kolom ini akan digunakan untuk menampilkan *Issue* apa saja yang harus dikerjakan oleh Developer.
- c. *In Progress*, kolom ini akan digunakan untuk menampilkan *Issue* apa saja yang sedang dikerjakan oleh Developer.
- d. *Ready for UAT*, kolom ini akan digunakan untuk menampilkan *Issue* apa saja yang akan diberikan ke tim *Quality Assurance* untuk di testing.

D. Rancangan Agile Testing Board



Gambar 19. Rancangan Agile Testing Board

Berdasarkan gambar diatas:

- a. Terdapat 3 kolom proses yang akan digunakan oleh Developer.
- b. *In Development*, kolom ini akan digunakan untuk menampilkan *Issue* apa saja yang sedang dikerjakan oleh Developer.
- c. *In UAT*, kolom ini akan digunakan untuk menampilkan

Issue apa saja yang sedang dikerjakan oleh *Quality Assurance*.

- d. *Closed*, kolom ini akan digunakan untuk menampilkan *Issue* apa saja yang sudah diselesaikan dalam proyek.

V. PENUTUP

A. Kesimpulan

Berdasarkan hasil penelitian yang telah dilakukan dapat diambil beberapa kesimpulan sebagai berikut :

1. Memiliki suatu sistem manajemen proyek adalah suatu keharusan yang harus ada disetiap perusahaan yang ingin memiliki suatu sistem pengawasan dan pengendalian yang baik. Sistem manajemen proyek yang masih bersifat manual tidak bisa dibilang sistem yang baik, dikarenakan kurangnya pengawasan dan pengendalian dari proses yang berjalan. Penggunaan sistem manajemen proyek yang menggunakan teknologi informasi merupakan pilihan yang terbaik.
2. Berdasarkan hasil penelitian yang dilakukan penulis, sistem yang berjalan di PT. FLASHiZ Indonesia, khususnya di bagian produk DIMO Payment, bisa dibilang masih berjalan secara manual. Sistem yang ada hanya digunakan untuk mengawasi atau mendaftar proyek apa saja yang ada di perusahaan tersebut, tetapi pengawasan kinerja dari proyek tidak ada. Beberapa masalah yang muncul seperti ketidakjelasan produk yang dikembangkan, waktu pengerjaan yang mundur atau telat, laporan yang tidak tepat waktu, perubahan-perubahan taanpa dokumentasi, dan lain sebagainya termasuk kekurangan sumber daya manusia dalam pengembangannya.
3. Untuk menjawab kekurangan dari sistem yang berjalan di PT. FLASHiZ Indonesia adalah dengan memaksimalkan salah satu aplikasi yang sudah dimiliki oleh perusahaan tersebut, yaitu Atlassian JIRA. Aplikasi sudah terbukti fungsinya sebagai salah satu aplikasi terbaik untuk manajemen proyek yang banyak digunakan oleh perusahaan-perusahaan yang sejenis (misal, *software house*). Aplikasi ini bisa digunakan lebih dari sekedar manajemen proyek, tergantung kebutuhan dari perusahaan itu sendiri. Khusus untuk PT. FLASHiZ Indonesia, penulis merancang *workflow* standar yang bisa digunakan diberbagai macam produk yang ada di perusahaan, beberapa *dashboard* yang bisa menampilkan informasi-informasi yang dibutuhkan secara *real-time*, kapanpun, dan dimanapun, dan beberapa laporan yang bisa digunakan sebagai bahan analisa untuk perencanaan proyek berikutnya.

B. Saran

Selama ini proses yang sudah berjalan di PT. FLASHiZ Indonesia terbilang cukup manual, dikarenakan banyak tugas-tugas yang dikerjakan tidak terdaftar dalam sistem yang sudah ada, mengakibatkan banyak salah komunikasi, kurang pengawasan, pengembangan yang tidak maksimal karena kurangnya informasi dan sumber daya manusia. Oleh karena itu penulis menyarankan kepada pihak manajemen PT.

FLASHiZ Indonesia untuk memanfaatkan secara maksimal aplikasi yang sudah dimiliki, supaya kinerja bisa lebih cepat dan efisien. Selain itu, juga melakukan pembiasaan diri untuk menggunakan aplikasi maupun standar prosedur dalam proses kerja sehari-hari. Agar tujuan sistem manajemen proyek yang diinginkan bisa tercapai.

DAFTAR PUSTAKA

- [1] A. Kadir, *Pengenalan Sistem Informasi*, Penerbit Andi, Yogyakarta 2003.
- [2] Atlassian Developer, Website, <https://developer.atlassian.com/index.html>
- [3] Atlassian Documentation, Website, <https://confluence.atlassian.com/display/ALLDOC/Atlassian+Documentation>
- [4] D. Darmawan dan K. N. Fauzi, *Sistem Informasi Manajemen*, PT. Remaja Rosdakarya, Bandung, 2013.
- [5] S. George M, *Prinsip-Prinsip Sistem Informasi Manajemen*, Raja Grafiika Persada, Jakarta, 2004.
- [6] I. Heryanto dan T. Triwibowo, *Manajemen Proyek Berbasis Teknologi Informasi*, Informatika, Bandung, 2013.
- [7] JIRA 101, Website, <https://confluence.atlassian.com/display/JIRA064/JIRA+101>
- [8] JIRA Administrator's Guide, Website, <https://confluence.atlassian.com/display/JIRA064/JIRA+Administrator%27s+Guide>
- [9] JIRA Agile Documentation, Website, <https://confluence.atlassian.com/display/AGILE/JIRA+Agile+Documentation>
- [10] JIRA Developer Documentation, Website, <https://developer.atlassian.com/jiradev/>
- [11] JIRA Documentation, Website, <https://confluence.atlassian.com/display/JIRA064/JIRA+Documentation>
- [12] JIRA Installation and Upgrade Guide, Website, <https://confluence.atlassian.com/display/JIRA064/JIRA+Installation+and+Upgrade+Guide>
- [13] JIRA User's Guide, Website, <https://confluence.atlassian.com/display/JIRA064/JIRA+User%27s+Guide>
- [14] Manifesto for Agile Software Development, Website, <http://agilemanifesto.org/>
- [15] M. Leod, *Information System Concepts*, Macmilian Publishing Company, New York, 1994.
- [16] M. Leod, *Management Information System*, Macmilian Publishing Company, New York, 2004.
- [17] Project Management Institute, *A Guide to the Project Management Body of Knowledge*, 1996.
- [18] S. S. Hamidjoyo, *Ciri-Ciri Informasi*, Bahan Perkuliahan PPs Unpad BKU Ilmu Komunikasi UNPAD, Bandung, 2000.
- [19] Scrum Methodology – Learn Scrum, Website, <http://scrummethodology.com/>
- [20] Yakub, *Pengantar Sistem Informasi*, Graha Ilmu, Yogyakarta, 2012.