

Perancangan Sistem Informasi Monitoring Proyek Pembangunan Software dengan Pendekatan CBSE

Wina Witanti*¹, Falahah²

¹Jurusan Teknik Informatika, Universitas Jenderal Achmad Yani; Cimahi

²Program Studi Sistem Informasi, Universitas Widyatama; Jl. Cikutra no.204 A Bandung

e-mail: *¹witanti@gmail.com, ²falahah@widyatama.ac.id

Abstrak

Saat ini, adopsi metoda Agile dalam pembangunan software sudah banyak dilakukan, baik oleh kelompok developer kecil maupun besar. Salah satu metode Agile yang cukup populer digunakan adalah Scrum. Penggunaan scrum sudah dikenal cukup luas oleh kalangan developer di Indonesia. Namun, dalam pengelolaan dan monitoring proyek, banyak tim pengembang masih menggunakan perangkat yang terpisah-pisah dan aplikasi monitoring proyek yang ada saat ini belum dapat mengadopsi proses atau event dan artifak pada scrum. Atas dasar permasalahan tersebut, dipandang perlu merancang sistem monitoring proyek pembangunan software berbasis scrum, sehingga memudahkan dalam realisasi pembangunan sistem. Perancangan software ini dilakukan dengan pendekatan Componen Based Software Engineering (CBSE), dengan tujuan agar tersedia rancangan komponen sistem yang dapat memenuhi kebutuhan pemantauan proyek pembangunan software sesuai dengan pendekatan scrum, sekaligus komponen yang dapat digunakan ulang (reusable component). Hasil penelitian berupa tersedianya rancangan arsitektur sistem berbasis komponen yang siap diimplementasikan.

Kata kunci— Scrum, Proyek, pemantauan, komponen, CBSE

Abstract

Nowadays, adoption of Agile methods in software development has been widely carried out, both by small and large developer groups. One of the popular Agile methods is Scrum. The use of Scrum is well known by developers in Indonesia. However, in project management and monitoring, many development teams still use separate tools and the current project monitoring application has not been able to adopt processes or events and artifacts on Scrum. Based on these problems, it is deemed necessary to design the components needed in the development of a scrum-based software development project monitoring system, so as to facilitate the realization of system development. The approach for designing components was conducted using Componen Based Software Engineering (CBSE), to produce a system component design that can meet the needs of monitoring software development projects in accordance with the Scrum approach. The results of the study consisted of component-based system designs that were ready to be implemented.

Keywords— Scrum, Project, Monitor, Component, CBSE

1. PENDAHULUAN

Scrum merupakan salah satu pendekatan dalam pengelolaan proyek pembangunan software atau aplikasi yang cukup populer. Saat ini, *Scrum* menempati posisi sebagai metode Agile yang paling banyak digunakan dibandingkan dengan metode-metode agile lainnya seperti *Kanban*, *Crystal*, *XP* dan lain-lain [1]. Di Indonesia sendiri, scrum sudah banyak digunakan, baik pada perusahaan pengembang berskala besar maupun tingkatan *start-up* [2]. Demikian halnya dengan pihak

pengguna, sudah banyak yang meminta pengelolaan proyek pembangunan *software* dilaksanakan dengan pendekatan *scrum*. Waktu pengembangan yang relatif singkat serta sifat proyek yang dianggap lebih adaptif [3] menyebabkan *scrum* menjadi salah satu metoda yang sangat banyak penggemarnya saat ini.

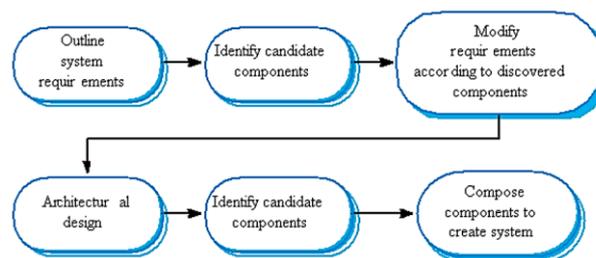
Di lain pihak, proyek pembangunan yang dikelola dengan pendekatan *scrum* juga mengalami kendala, salah satunya adalah kendala untuk memonitoring perkembangan proyek. *Scrum* memiliki acara atau *event* berupa *daily scrum* serta *sprint retrospective* yang dapat digunakan untuk memonitor perkembangan proyek. *Event* ini berupa pertemuan singkat dimana tim *scrum* mendiskusikan hasil pekerjaan serta merancang kegiatan berikutnya. Namun, monitoring kemajuan proyek belum dilengkapi dengan *tools* khusus sehingga pencatatan *progress*, grafik, penugasan dan lain-lain masih dilakukan secara terpisah-pisah pada beberapa *file* misalnya dokumen ataupun kertas kerja elektronik atau *spreadsheet*. Pengelolaan yang terpisah-pisah ini menyulitkan seorang *scrum master* untuk menganalisis dan memberikan laporan yang komprehensif kepada pemilik pekerjaan.

Sejauh ini, belum ada artikel penelitian yang membahas tentang pembangunan aplikasi monitoring proyek menggunakan *scrum*. Pada umumnya aplikasi atau sistem informasi monitoring proyek dibangun tidak spesifik untuk pendekatan *scrum*. Beberapa penelitian terkait misalnya sistem monitoring kegiatan proyek pembangunan pemancar sinyal BTS yang digunakan untuk memonitor pekerjaan pembangunan infrastruktur [4], dan sistem monitoring pengembangan *software* yang dikhususkan pada tahap *development* (tidak sampai tahap pengujian ataupun implementasi) [5]. Sedangkan implementasi kerangka kerja *scrum* sendiri juga pernah diulas pada beberapa artikel seperti Rizaldi dkk [6], Fernando [7], dan Ruseno [8]. Pada umumnya hasil penelitian memaparkan teori *scrum* secara umum serta contoh implementasinya pada pembangunan *software* yang dilengkapi dengan perencanaan proyek berupa tabel *backlog* dan *sprint backlog*.

Atas dasar masalah tersebut, dipandang perlu untuk membangun sebuah aplikasi untuk memonitor proyek pembangunan *software* yang dikelola dengan pendekatan *scrum*. Aplikasi ini diharapkan dapat membantu *scrum master* serta anggota tim *developer* untuk membagi-bagi pekerjaan, melaporkan *progress* serta memonitor *progress* pekerjaan secara keseluruhan.

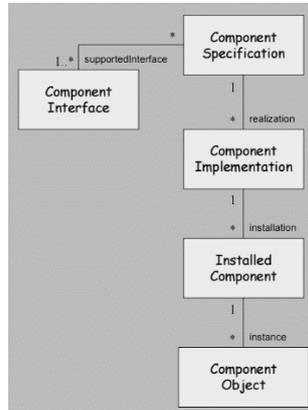
2. COMPONENT BASED SOFTWARE ENGINEERING

Pada penelitian ini, perancangan sistem dilakukan dengan pendekatan *Component Based Software Engineering* (CBSE). Pada CBSE, komponen adalah obyek pada *software* yang dapat berinteraksi dengan komponen lain, dan pada obyek tersebut tercakup sekumpulan fungsionalitas atau fungsionalitas tertentu [9]. Proses CBSE sendiri melibatkan 6 langkah yaitu menentukan kebutuhan sistem, mengidentifikasi komponen, memodifikasi requirement untuk mencari komponen, perancangan arsitektur, mengidentifikasi kandidat komponen dan melakukan komposisi komponen untuk merancang sistem [10]. Gambar 1 menampilkan langkah-langkah dalam pembangunan *software* dengan pendekatan *component-based*.



Gambar 1. Tahapan Pembangunan *Software* berbasis Komponen [10]

Berdasarkan sudut pandang proses rekayasa sistem, komponen dapat dibagi menjadi 5 bentuk yaitu [11]: spesifikasi komponen, komponen antarmuka (interface component), komponen implementasi (component implementation), installed component, dan object component, yang serupa dengan konsep object dan class pada OOP. Keterkaitan antar berbagai bentuk komponen tersebut dapat dilihat pada gambar 2.



Gambar 2. Berbagai bentuk Komponen [12]

Pendekatan adaptif prinsip CBSE pada pembangunan software meliputi 5 langkah yaitu [12]: identifikasi kebutuhan, identifikasi komponen, identifikasi interaksi komponen, menentukan spesifikasi komponen dan provisioning atau memilih teknologi untuk implementasi hasil rancangan spesifikasi komponen.

1. Identifikasi kebutuhan meliputi kegiatan pendefinisian model konsep bisnis, proses bisnis dan identifikasi use case. Output dari langkah ini adalah tersedianya model konsep bisnis dan diagram usecase.
2. Identifikasi komponen, meliputi kegiatan mengidentifikasi model tipe bisnis, identifikasi antarmuka bisnis, identifikasi antarmuka sistem dan operasional, dan membuat spesifikasi awal komponen beserta arsitekturnya.
3. Identifikasi interaksi komponen, meliputi kegiatan mengidentifikasi operasional bisnis, menyempurnakan interaksi dan operasional yang sudah didapat pada fase sebelumnya, dan merevisi atau menyempurnakan spesifikasi dan arsitektur komponen.
4. Spesifikasi komponen, merupakan langkah mendefinisikan spesifikasi komponen yang akan dibuat dengan memperhatikan perilaku dan informasi yang dipertukarkan dan relasi antar komponen.
5. Provisioning, yaitu langkah mengimplementasikan komponen pada lingkungan dan teknologi yang dipilih.

3. HASIL DAN PEMBAHASAN

Tahapan yang dilakukan pada perancangan komponen sistem ini mengikuti langkah-langkah CBSE adaptasi CBSE seperti yang dikemukakan oleh Daniels [12]. Namun, pada artikel ini bahasan dibatasi hingga contoh spesifikasi komponen. Contoh kasus yang akan digunakan adalah pembangunan sistem informasi monitoring progress proyek perangkat lunak di PT.X [13]. Proses pembangunan software di PT X sudah menggunakan pendekatan Scrum, dan hingga saat ini PT.X belum memiliki sistem untuk memonitor perkembangan proyek-proyek tersebut.

3.1. Requirement Analysis

Pada fase ini dilakukan analisis atas kebutuhan sistem yang akan dibangun, sebagai berikut:

Obyek penelitian adalah sebuah perusahaan pembangunan software yaitu PT.X, yang berfokus pada pengembangan aplikasi berbasis web dengan menggunakan berbagai kerangka kerja dan teknologi yang terkini. Semua proyek pembangunan software dilakukan dengan pendekatan *scrum*.

Berdasarkan pengamatan pada kondisi manajemen proyek pembangunan software di PT.X, ditemukan beberapa kondisi antara lain:

- a. Pembangunan *software* pada umumnya dilakukan menggunakan pendekatan *Scrum*.
- b. Satu *programmer* dapat terlibat pada beberapa proyek pembangunan *software* sehingga *programmer* harus membuat laporan yang akurat atas *progress* pekerjaannya untuk masing-masing proyek, kepada masing-masing manajer proyek. Untuk proyek yang dikerjakan menggunakan pendekatan *scrum*, manajer proyek dilaksanakan oleh seorang *Scrum Master*.
- c. *Product Owner* melakukan proses pembagian pekerjaan, pembagian kelompok tim developer dan penentuan fitur utama aplikasi melalui *product backlog*.
- d. *Scrum master* memonitoring hasil pekerjaan dan memberikan laporan *progress* pekerjaan menggunakan tools atau teknologi aplikasi yang umum seperti pengolah dokumen (*word processor*), pengolah kertas kerja (*spreadsheet*), dan aplikasi pengelola proyek pada umumnya.
- e. Teknologi dan aplikasi umum tersebut belum dapat memenuhi kebutuhan pelaporan berkala dan intensif seperti pada metode *scrum* yaitu memberikan laporan berupa kegiatan harian (*daily scrum*), feedback di setiap akhir sprint (*sprint retrospective*), dan menelusuri perubahan *backlog*.

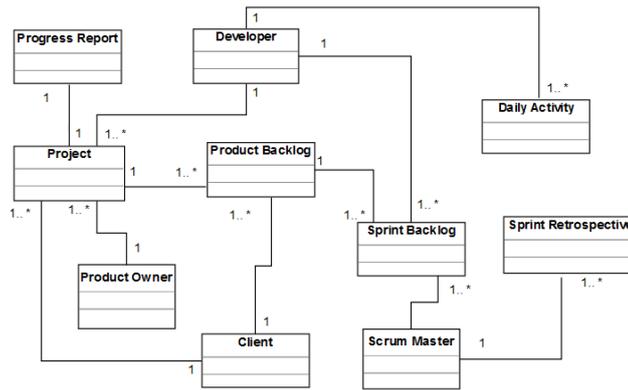
Stakeholder yang terlibat:

Pada kasus ini, stakeholder yang terlibat adalah: pimpinan utama proyek yang mewakili *product owner*, *Scrum Master*, yang bertanggung jawab terhadap terlaksananya proyek pengembangan software dan *Developer* yang bertanggung jawab membuat dan menguji software atau komponen software.

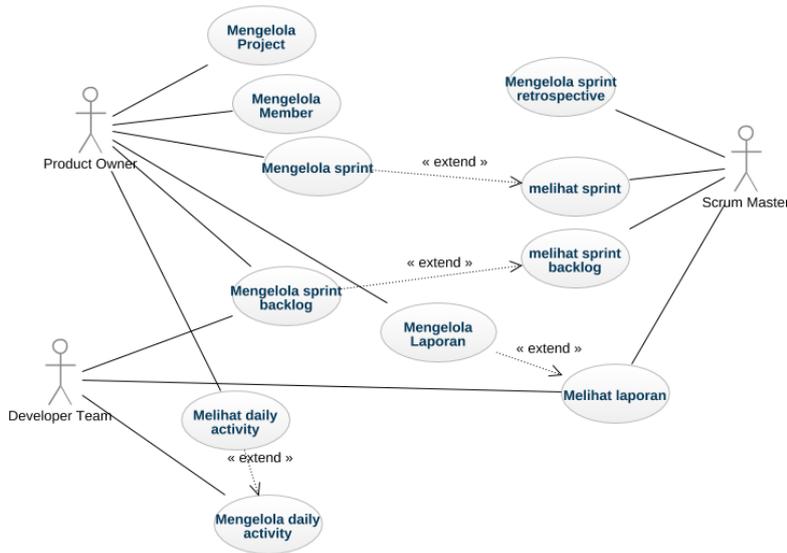
Berdasarkan asumsi, definisi dan analisis atas proses yang sudah berjalan, serta usulan solusi yang diajukan, maka dapat disusun rancangan komponen sistem informasi yang akan dikembangkan. Rancangan komponen ini mengacu pada fitur-fitur yang perlu disediakan pada sistem tersebut, yaitu: Manajemen proyek, Manajemen *sprint* dan pembagian tugas, Laporan kegiatan harian *developer*, Laporan *sprint planning*, Laporan *daily scrum*, Laporan *sprint review*, Laporan *sprint retrospective*, dan *Dashboard*. Manajemen Proyek meliputi: pengisian identitas proyek baru, penambahan anggota tim yang meliputi *scrum master* dan tim developer, mendefinisikan *product backlog* dan *sprint backlog*. Manajemen *sprint* dan pembagian tugas meliputi: mendefinisikan *sprint backlog*, mendistribusikan penugasan *backlog* pada *sprint*, mengisi laporan *daily activity*, dan mengelola *sprint retrospective*.

Berdasarkan identifikasi kebutuhan dan fungsionalitas sistem di atas, dapat digambarkan model konsep bisnis yang akan dibuat, seperti pada gambar 3. Sedangkan hasil identifikasi kebutuhan atas fungsionalitas sistem digambarkan dalam bentuk use case diagram seperti pada gambar 4.

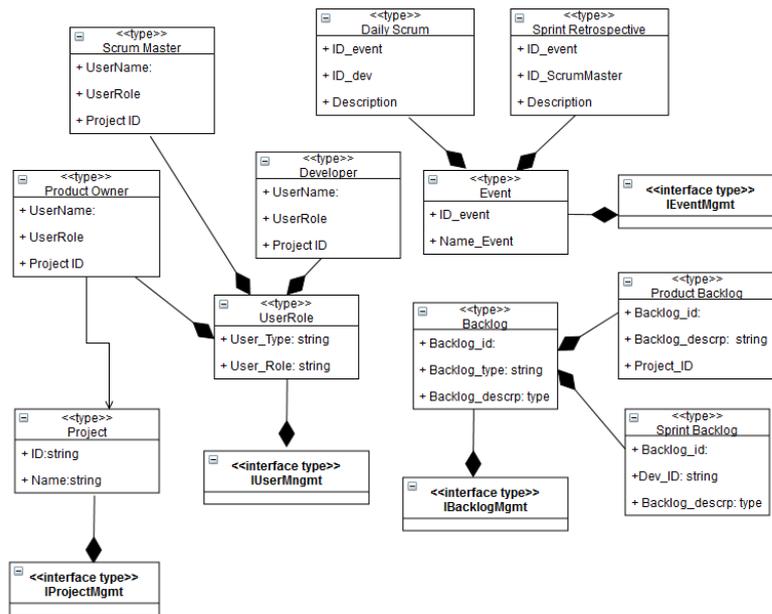
Berdasarkan interaksi antar komponen pada model konsep bisnis di atas, dapat didefinisikan model tipe bisnis, yaitu dengan mengekstraksi komponen-komponen utama yang harus ada pada sistem (*core type*). Hasil identifikasi *core type* digambarkan dalam bentuk model konsep bisnis seperti pada gambar 5.



Gambar 3. Model Konsep Bisnis



Gambar 4. Use Case Diagram



Gambar 5. Diagram Interface responsibility

3.2. Identifikasi Komponen

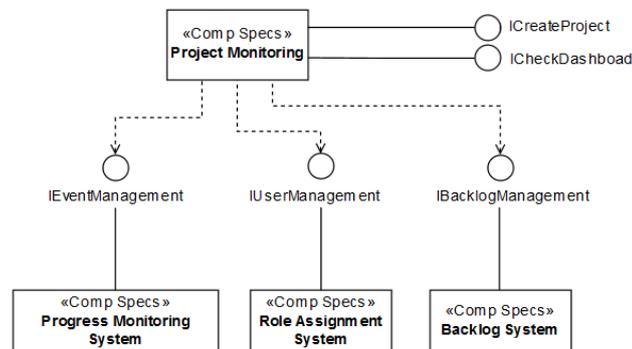
Pada fase ini dilakukan beberapa perencanaan tindakan untuk menyelesaikan masalah yang sudah teridentifikasi sebelumnya. Langkah-langkah tersebut meliputi: menentukan asumsi dan definisi, menganalisis interaksi antar komponen dan membuat arsitektur awal komponen.

Definisi dan asumsi yang digunakan pada perancangan solusi ini adalah:

1. *Product owner*, yaitu pihak manajemen perusahaan yang mewakili klien pemberi pekerjaan. Hal ini dikarenakan pada proyek pembangunan *software* ini, klien pemberi pekerjaan tidak selalu dapat berinteraksi langsung dengan tim pengembang.
2. *Scrum master*, dalam hal ini adalah karyawan yang ditunjuk sebagai penanggung jawab proyek yang bertugas membagikan pekerjaan, memonitor hasil dan mengambil tindakan jika terjadi kendala atau masalah
3. *Developer* adalah karyawan atau anggota tim yang diberi tugas sesuai item pada *product backlog*.
4. *Product backlog* adalah daftar pekerjaan yang harus dipenuhi, yang dibuat berdasarkan kesepakatan dengan klien pemberi pekerjaan
5. *Sprint* adalah periode pengerjaan yang dibagi berdasarkan kesepakatan, misalnya 1 minggu, 2 minggu atau periode lain sesuai hasil analisis.
6. *Sprint backlog* adalah pekerjaan yang harus diselesaikan pada masa satu sprint
7. *DailyReport* adalah laporan progress harian pekerjaan yang akan diisi oleh masing-masing *developer* sesuai penugasan.
8. *Sprint retrospective* adalah rewiu terhadap hasil dari satu sprint untuk mengidentifikasi kemajuan proyek, sekaligus mengumpulkan feedback dan mengambil tindakan antisipasi atas permasalahan

Selanjutnya dapat diidentifikasi komponen-komponen utama yaitu *scrum master*, *product owner*, *developer*, *daily scrum*, *sprint retrospective*, *product backlog*, dan *sprint backlog*. Komponen-komponen tersebut perlu dianalisis lebih lanjut untuk mengetahui interaksi dan tanggung jawab masing-masing komponen. Hasil pengelompokan komponen berdasarkan tanggung jawab masing-masing dapat dilihat pada gambar 6.

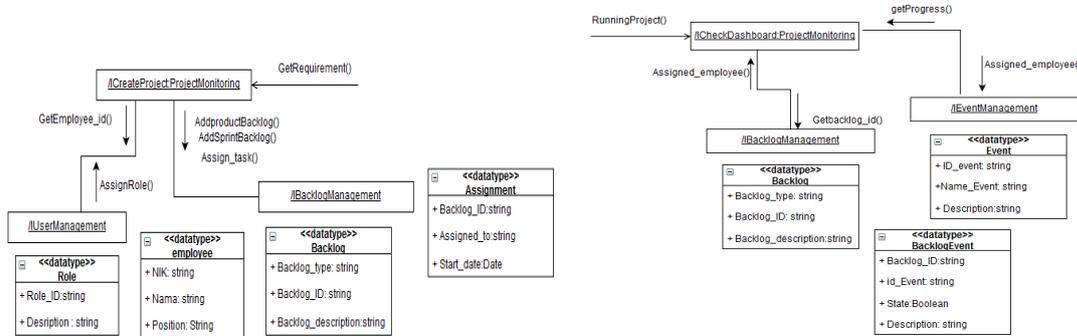
Beraskan hasil analisis interface dan responsibility masing-masing komponen, dapat didefinisikan arsitektur awal komponen seperti pada gambar 6. Pada gambar terlihat bahwa berdasarkan pendekatan berbasis komponen, maka sistem informasi monitoring proyek ini dapat dibagi menjadi 3 interface utama yaitu pengelolaan event (*ieventmanagement*), pengelolaan pengguna (*IUsermanagement*), dan Manajemen backlog (*IBacklog Management*).



Gambar 6. Arsitektur awal komponen

3.3. Analisis Interaksi antar Komponen

Setelah interface masing-masing komponen dapat didefinisikan, selanjutnya dilakukan analisis interaksi antar komponen. Analisis ini dapat dilakukan dengan menelusuri proses pada masing-masing interface. Gambar 7.a. menampilkan model interaksi antar komponen untuk interface iCreateProject, iUserManagement dan IBacklogManagement. Interaksi yang sama juga dapat dianalisis dan ditelusuri untuk interface CheckDashboard, dan hasilnya dapat dilihat pada gambar 7.b.

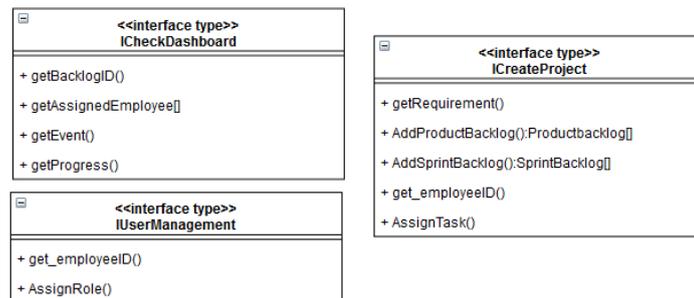


Gambar 7. Diagram Interaksi Interface iCreateProject

3.3. Component Specification

Berdasarkan hasil identifikasi komponen dan interaksi antar komponen, dapat didefinisikan spesifikasi komponen. Spesifikasi ini didefinisikan melalui model informasi yang dibutuhkan pada setiap interface, data yang diperlukan, dan proses yang harus dikerjakan. Gambar 8 menampilkan model informasi untuk tiga interface yang dimaksud.

Gambar 7. Diagram Interaksi Interface iCreateProject



Gambar 8. Model Informasi Interaksi

Berdasarkan model informasi tersebut, dapat didefinisikan spesifikasi komponen seperti contoh berikut:

Component Specification

Context ICreateProject (in:Requirement, in:Employee_Id, out:ProjectPlan: string):Boolean

Pre:

Employee is available
Employee → available

Post:

Result implies
Let emp = emp → select (e | e.stat <>full) → asSequence
If e.stat <> full then
Set e.role = dev
...
Let Req = list_of_req()
While Req.exist
Create_Backlog
Assign_employee

4. KESIMPULAN

Penelitian ini memaparkan proses identifikasi kebutuhan dan perancangan sistem informasi monitoring proyek pembangunan software, dengan menggunakan pendekatan component-based software engineering (CBSE). Sistem informasi monitoring proyek ini dirancang agar dapat mengadopsi kebutuhan pelaporan dan dokumentasi proyek menggunakan scrum, yang agak berbeda dengan model pelaporan, penelusuran kemajuan proyek dan dokumentasi proyek. Berdasarkan hasil penelitian, diperoleh 4 komponen utama pada sistem ini yang bertanggung jawab sebagai antarmuka (interface) fungsi-fungsi utama pada sistem yaitu `icreateProject()`, `IcheckDashboard()`, `IUserManagement`, dan `IBacklogManagement`. Interface ini kemudian terhubung dengan komponen utama sistem yang meliputi komponen project, backlog, Role dan event. Komponen dasar ini dapat memuat sistem terdekomposisi dengan lebih baik dan masing-masing komponen dapat diimplementasikan dengan tingkat kebergantungan yang rendah. Hal ini membuat sistem yang dikembangkan lebih fleksibel terhadap pemilihan teknologi implementasi.

DAFTAR PUSTAKA

- [1] A. Lamelas, "Top 5 main Agile methodologies: advantages and disadvantages," 2018. [Online]. Available: <https://www.xpand-it.com/2018/10/11/top-5-agile-methodologies/>. [Accessed: 10-Aug-2019].
- [2] "Kenapa menggunakan Scrum, kalau semua sudah baik-baik saja." [Online]. Available: <http://www.scrum.co.id/why-scrum>. [Accessed: 12-Aug-2019].
- [3] C. Kulbacki, "What's More Popular Than Scrum?," 2018. [Online]. Available: <https://scrumstar.com/articles/the-most-popular-agile-methodologies>. [Accessed: 12-Aug-2019].
- [4] D. Megawati, A., Gustina, "Membangun Sistem Informasi Monitoring Kegiatan Proyek Pemancar Sinyal BTS Berbasis Web Pada PT. Swatama Mega Teknik," *J. Ilm. Fifo*, vol. Mei 2018, 2018.
- [5] P. . Herliana, A., Rasyid, "Sistem Informasi Monitoring Pengembangan Software Pada Tahap Development Berbasis Web," *J. Inform.*, vol. III, no. 1, 2016.
- [6] H. Rizaldi, T, Sarwo, D.P, Yufit, "Implementasi Metodologi SCRUM dalam Pengembangan Sistem Pembayaran Elektronik Pada Usaha Mikro Kecil Menengah," in *Seminar Hasil Penelitian dan Pengabdian Masyarakat Dana BOPTN Tahun 2016*, 2016.
- [7] F. Fernando, D, Anharudin, "Rancang Bangun Aplikasi E-Portofolio Hasil Karya Mahasiswa UNSERA Menggunakan Metode Scrum," *J. Sist. Inf.*, vol. 5, no. 1, 2018.
- [8] D. Ruseno, "Implementasi Scrum pada Pengembangan Aplikasi Sistem Reservasi Online Menggunakan PHP," *J. Gerbang*, vol. 9, no. 1, 2019.
- [9] "Component-Based Architecture." [Online]. Available: https://www.tutorialspoint.com/software_architecture_design/component_based_architecture.htm. [Accessed: 10-Aug-2019].
- [10] I. Sommerville, *Software Engineering, 7th Ed.* 2004.

- [11] A. J. A. Wang and K. Qian, *Component Oriented Programming*. John Wiley & Sons, 2005.
- [12] J. Daniels, "Component-Based Design : A Complete Worked Example," 2002. [Online]. Available: <http://www.syntropy.co.uk/papers/CBDtutorial.pdf>.
- [13] H. Nurafifah, "Sistem Informasi Monitoring Progress Proyek Perangkat Lunak Berbasis Scrum," 2019.