

Aplikasi Manajemen *Multi* Proyek menggunakan Metode *Scrum*

Andri Wijaya Kusuma¹, Nurudin Santoso², Arief Andy Soebroto³

Program Studi Teknik Informatika, Fakultas Ilmu Komputer, Universitas Brawijaya
Email: ¹wijayakusuma@student.ub.ac.id, ²nurudin.santoso@ub.ac.id, ³ariefas@ub.ac.id

Abstrak

Dalam suatu *software house*, manajer proyek memegang peranan penting dalam proses pengelolaan dan pembuatan perangkat lunak. Manajer proyek untuk setiap proyeknya diberikan kepada orang yang berbeda. Namun dimungkinkan juga untuk seseorang berperan sebagai manajer proyek pada lebih dari 1 proyek dalam waktu yang bersamaan. Untuk cara pelaporan dari tiap pekerja kepada manajer proyek-pun bervariasi. Ada yang melakukan rapat setiap seminggu sekali, atau pada hari kerja sebelum pulang, atau manajer proyek harus menanyakan sendiri kepada tiap anggotanya. Sehingga manajer proyek tidak dapat memantau perkembangan pengerjaan terkini dari anggota tim setiap waktunya dan juga tidak dapat secara langsung memberikan tugas untuk anggota tim begitu pula anggota tim tidak dapat mengetahui tugas yang harus dikerjakan jika tugas sebelumnya telah selesai dikerjakan. Untuk menangani permasalahan tersebut, penulis mengembangkan aplikasi manajemen multi proyek yang dapat mengelola aktivitas pekerjaan yang dilakukan, dan memantau perkembangan proyek. Aplikasi tersebut menerapkan metode *scrum*, yang didalamnya terdapat iterasi untuk proses pengerjaan proyek. Dalam pengembangan aplikasi penulis menggunakan *waterfall model*. Berdasarkan pengujian yang dilakukan, pengujian unit, pengujian integrasi, dan pengujian validasi menghasilkan nilai *valid* pada kasus uji yang telah dilakukan. Untuk pengujian *compatibility* diperoleh bahwa aplikasi dapat berjalan optimal pada 9 dari 11 versi *browser* yang diujikan.

Kata kunci: manajemen multi proyek, *scrum*, aplikasi, *software house*.

Abstract

In a software house, project managers took an important role in managing and making software. The project manager for each project is given to different people. But it is also possible for someone to act as a project manager on more than 1 project at the same time. For how to report from each worker to the project manager also varies. There are those who hold meetings once a week, or on weekdays before going home, or the project manager must ask to each member. So, the project manager cannot monitor every progress of the latest workmanship of the team members at all times and also cannot directly assign assignments to the team members as well as team members cannot know the tasks that must be done if the previous task has been completed. To solve the problem, the author developed a multi-project management application that can manage work activities carried out, and monitor the progress of the project. The application applies the Scrum method, in which there is an iteration for the process of working on the project. In developing the application the author uses the waterfall model. Based on the testing performed, unit testing, integration testing, and validation testing produced a valid value in the test case that has been done. In compatibility testing shows that the application can run optimal on 9 of the 11 browser versions tested.

Keywords: multi project management, *scrum*, application, *software house*

1. PENDAHULUAN

Pada era teknologi sekarang, perkembangan teknologi informasi semakin cepat, baik dari hal pengadaan proyek, ataupun biaya yang digunakan untuk proyek tersebut. Pertumbuhan

negara Indonesia sebagai negara berkembang dapat dilihat dengan menjamurnya banyak proyek yang memiliki skala besar dan dibangun oleh pemerintah ataupun swasta khususnya teknologi. Pada perusahaan yang bergerak dibidang perangkat lunak atau *software house*, memiliki berbagai macam proyek yang mana

hasil dari proyek tadi dapat dijadikan sebagai produk dari perusahaan tersebut. Produk dari software house dapat berupa perangkat lunak untuk client. Menurut Larson (2006:3), Proyek merupakan upaya yang rumit, tidak rutin, serta dibatasi oleh waktu, biaya, sumber daya, dan spesifikasi dari tiap kinerja yang dirancang dalam memenuhi apa yang diperlukan pelanggan. Suatu proyek terdiri dari beberapa orang anggota yang dipimpin oleh manager proyek.

Manajer Proyek memegang peranan yang penting pada proses pembuatan produk. Dari hasil wawancara yang dilakukan oleh penulis, diperoleh bahwa proses pelaporan progress pekerjaan biasanya dilakukan seminggu sekali, atau pada waktu hari kerja sebelum pulang, atau manajer proyek harus menanyakan sendiri kepada tiap anggotanya. Sehingga manajer proyek tidak dapat memantau perkembangan pengerjaan terkini dari anggota tim setiap waktunya. Selain tidak dapat memantau perkembangan terkini, manajer proyek juga tidak dapat secara langsung memberikan tugas untuk anggota tim begitu pula anggota tim tidak dapat mengetahui tugas yang harus dikerjakan jika tugas sebelumnya telah selesai dikerjakan. Masalah lain yang timbul dari software house adalah adanya miscommunication yang terjadi antar pekerja. Untuk penanggung jawab setiap proyek diberikan kepada orang yang berbeda. Namun dalam waktu bersamaan, seorang pekerja dimungkinkan berperan sebagai manajer proyek pada dua atau lebih pekerjaan.

Seiring berkembangnya teknologi informasi, semakin ketat pula kompetisi dalam penyelenggaraan proyek pada software house dalam memenuhi apa yang diperlukan *client*, sehingga perlu adanya cara pengelolaan, metode dan teknik yang terbaik. Hal tersebut diharapkan supaya penggunaan dari sumber daya yang dimiliki akan sepenuhnya efektif dan efisien, maka diperlukan suatu manajemen proyek yang berguna dalam mengatur keseluruhan proyek perangkat lunak yang dikerjakan. Manajemen Proyek berisi proses perencanaan, mengarahkan dan menggunakan sumber daya yang dimiliki perusahaan guna mencapai target dalam waktu yang telah ditentukan. Manajemen proyek menempatkan anggotanya pada tugas tertentu. Sedangkan perangkat lunak ialah suatu perintah program di dalam suatu komputer, yang mana jika dijalankan oleh user-nya akan memberikan fungsi dan unjuk kerja sesuai yang telah diharapkan oleh user-nya (Roger S. Pressman,

2002).

Pesatnya teknologi informasi, menuntut manusia dalam menggunakan komputer sebagai perangkat untuk proses perhitungan dan manajemen proyek. Perkembangan perangkat lunak ikut serta mendorong perubahan manajemen dari tradisional kedalam suatu perangkat lunak manajemen proyek. Dalam satu waktu, suatu perusahaan dapat menjalankan lebih dari satu proyek yang dikerjakan oleh tiap tim. Perangkat lunak ini nantinya dapat membantu manajer proyek pada hal perencanaan, pengorganisasian, serta mengelola sumber daya, sehingga dapat mencapai tujuan proyek. Untuk mengatasi masalah yang kompleks dari suatu proyek, maka diperlukan metode dalam penyelesaiannya. Contoh metode yang mana dapat diterapkan pada manajemen proyek adalah metode scrum.

Scrum adalah proses dalam hal pengembangan perangkat lunak Agile dimana akan berfokus pada praktik manajemen proyek. *Scrum* ialah kerangka kerja ringan serta memuat langkah-langkah untuk mengelola dan mengontrol proses pengembangan suatu produk. Metode Scrum mulai populer dalam beberapa tahun terakhir dan terbukti sangat berguna walaupun bukan metode yang selalu digunakan (Schwaber, K, 2002). Akhir-akhir ini Scrum telah mendapatkan popularitas besar dan diadopsi oleh perusahaan besar seperti Yahoo (G. Cloke, 2007), Microsoft, Intel, dan Nokia.

2. LANDASAN KEPUSTAKAAN

2.1. Kajian Pustaka

Terdapat tiga penelitian untuk dijadikan sebagai referensi pada penelitian kali ini. Pada penelitian pertama ditulis oleh Marchenko yang berjudul "*Scrum in a Multiproject Environment: An Ethnographically-Inspired Case Study on the Adoption Challenges*" (Marchenko, 2008) diperoleh bahwa saat itu, departemen kasus memiliki kesan positif dari proses Scrum. Masalah prioritas terhadap kebutuhan dianggap menjadi salah satu elemen kesuksesan dari Scrum. Terdapat sepuluh tantangan khusus yang diidentifikasi selama penelitian. Selanjutnya, aplikasi kasus Scrum dalam penelitian meningkatkan pemahaman tentang hambatan dalam situasi praktis yang berbeda. Dalam penelitian tersebut terdapat beberapa keterbatasan dalam penelitian, seperti periode waktu yang relatif singkat. Jika tujuannya adalah

untuk berhasil dalam lingkungan pada penelitian, maka hal terbaik yang harus dilakukan adalah membuat persiapan terlebih dahulu.

Menurut romano dalam penelitiannya yang berjudul “*Project management using the Scrum agile method: A case study within a small enterprise*” (Romano, 2015), yang mana terdapat perbandingan sebelum dan sesudah penyebaran metode Scrum. Dalam semua kasus ada perubahan positif, bahkan partisipasi tim dalam item dinilai memuaskan. Hasil juga menunjukkan peningkatan kualitas dan waktu yang signifikan. Pada penelitian dengan judul “*SCRUM Model for Agile Methodology*” menjelaskan tentang scrum yang memiliki siklus yang telah terbukti bahwa dapat mengakomodasi perubahan pada tiap iterasinya (Srivastava, et al., 2017).

2.2. Manajemen Proyek

Manajemen merupakan salah satu ilmu pengetahuan tentang memimpin suatu organisasi yang didalamnya mengandung kegiatan perencanaan, pengorganisasian, pelaksanaan, serta pengendalian kepada sumber daya yang terbatas dalam upaya untuk mencapai tujuan serta sasaran yang efektif dan efisien (Husen, 2009). Sedangkan Proyek ialah gabungan dari beberapa sumber daya yang meliputi manusia material, dan modal atau biaya yang dikumpulkan pada sebuah organisasi sementara yang nantinya untuk menggapai target dan tujuan (Husen, 2009).

Dari definisi diatas dapat ditarik kesimpulan mengenai sebagian arti dari manajemen proyek. Manajemen proyek ialah penerapan dari suatu ilmu pengetahuan, keahlian, cara terbaik dalam mengelola sumber daya yang terbatas, untuk dapat mencapai target dan tujuan yang telah diuraikan (Husen, 2009). Sebagai suatu proses, manajemen dapat memiliki arti bahwa proyek dilaksanakan berdasarkan langkah yang telah disepakati dan sesuai target, tujuan, serta keterbatasan yang dimiliki.

Peran dari manajemen proyek sangatlah penting dalam mendukung kegiatan pengembangan perusahaan. Dalam skala yang lebih luas pastinya manajemen multi proyek sangat diperlukan guna hasil yang diperoleh dari tiap-tiap proyek mulai dari perintisan proyek hingga tahap akhir dapat berjalan dengan lancar dan sesuai target menurut batas waktu

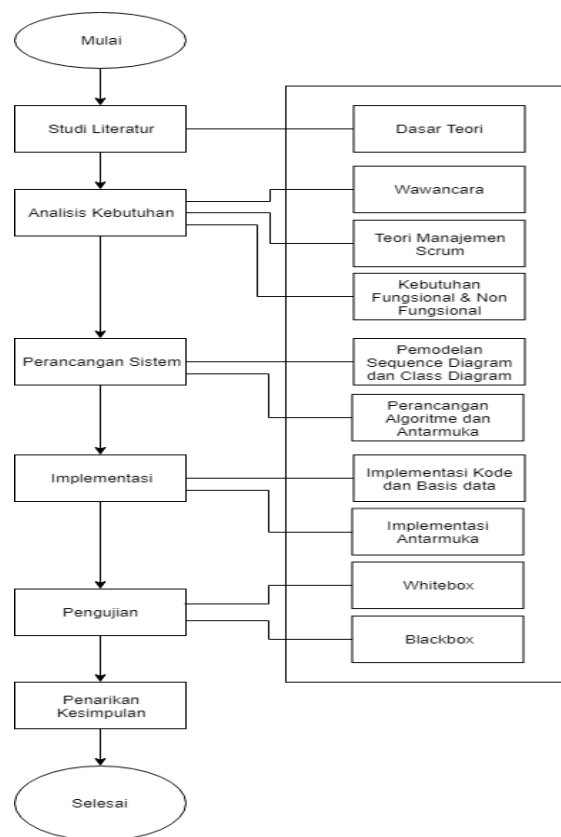
pengerjaan serta hasilnya sesuai dengan tujuan awal yang ingin dicapai.

2.3. Scrum

Scrum adalah suatu *framework* dimana penggunaannya akan dapat mengatasi masalah yang kompleks dan adaptif, serta diwaktu yang bersamaan, penggunaanya juga menghantarkan produk disertai nilai tinggi secara produktif dan kreatif (Ken & Jeff, 2017). Scrum memecahkan ketidak-efektifan dari proses manajemen produk dan teknik dalam melakukan kerja, sehingga pengguna dapat terus-menerus untuk selalu meningkatkan kinerja dari produk, tim, dan lingkungan kerja (Ken & Jeff, 2017).

3. METODOLOGI PENELITIAN

Pada bagian berikut penulis menguraikan urutan dalam proses perencanaan untuk penelitian yang hendak dilakukan dalam menyelesaikan masalah penelitian. Metode ini nantinya digunakan sebagai landasan dalam mengerjakan penelitian. Diagram langkah langkahnya berada pada Gambar 1.



Gambar 1 Diagram Alir Penelitian

3.1. Studi Literatur

Pada studi literatur berisi hasil dari studi kepustakaan yang dapat menunjang penulisan untuk penelitian. Sumber yang dipakai dapat berasal dari buku yang berisi teori ilmu, jurnal publikasi ilmiah, situs, serta tulisan yang memiliki kaitan dengan analisis dan perancangan aplikasi. Selain daripada itu, dilakukan juga studi literatur mengenai kerangka kerja serta teknologi yang akan digunakan dalam pengembangan Aplikasi Manajemen *Multi* Proyek.

3.2. Analisis Kebutuhan

Pada tahap ini, analisis kebutuhan dilakukan dengan dengan mengidentifikasi seluruh kebutuhan fungsional dan non fungsional oleh pengguna. Selain itu, dapat diperoleh suatu kebutuhan dari metode yang digunakan yaitu Scrum. Untuk pemodelan yang akan diterapkan merupakan pemodelan kebutuhan yang berorientasi kepada objek, dimana akan menggunakan diagram UML. Selain itu, digunakan pula *Usecase Diagram* dan *Scenario*. *Usecase Diagram* memberikan gambaran umum aplikasi dari tampak luar, yang berisi aktor pengguna sistem disertai fungsi yang dapat dilakukan.

3.3. Perancangan Sistem

Pada tahap perancangan, akan dilakukan sebagai landasan dari implementasi aplikasi. Perancangan ini bersumber pada analisis kebutuhan. Dari kebutuhan yang diperoleh kemudian akan dilakukan pemodelan dalam bentuk *Unified Modeling Language* (UML). Pemodelan UML yang nantinya hendak dilakukan adalah perancangan *Sequence Diagram*, perancangan pada diagram kelas, perancangan pada Basis Data, serta perancangan terhadap antarmuka.

3.4. Implementasi

Pada bagian ini, penulis akan mengubah hasil perancangan yang sebelumnya dirancang kedalam bentuk kode program, sehingga kode tersebut akan dapat dipahami dan dijalankan oleh komputer. Bahasa pemrograman yang akan digunakan adalah bahasa PHP dibantu dengan framework Laravel. Untuk tampilan antarmuka menggunakan HTML dan CSS dibantu dengan *framework* Bootstrap. Untuk basis data menggunakan MySQL.

3.5. Pengujian

Tahap pengujian dilakukan dalam upaya menemukan kesalahan pada Aplikasi seperti menguji apakah keseluruhan kebutuhan telah tepat dan sesuai dengan analisis kebutuhan. Pengujiannya akan dikerjakan menggunakan metode *Black box Testing* dan *White box Testing*. Pada *White box Testing* akan dilakukan pengujian jalur dasar dari algoritma yang telah dirancang. Tujuannya untuk mengukur kompleksitas algoritma. Untuk *Black Box Testing* dilakukan dengan *Requirement Testing* yang bertujuan dalam mengetahui jika aplikasi telah dikembangkan sesuai dengan kebutuhan atau tidak.

3.6. Penarikan Kesimpulan

Penarikan kesimpulan dilakukan ketika tahapan tahapan dalam metode penelitian telah dilakukan. Kesimpulan diperoleh dari hasil perancangan serta pengujian. Kesimpulan yang didapatkan harus mampu menjawab rumusan masalah yang dipaparkan diawal penelitian.

4. ANALISIS KEBUTUHAN

4.1. Deskripsi Aplikasi

Aplikasi ini berguna untuk memudahkan tim dalam pengembangan perangkat lunak pada suatu software house. Aplikasi ini dapat mengelola proyek yang sedang berjalan. Setiap anggota dapat berkontribusi dalam proses pengelolaan suatu proyek. Semua aktivitas dari anggota tim juga akan terekam dan dapat diketahui perubahan apa saja yang telah dilakukan. Karena menggunakan Scrum, maka beberapa kebutuhan juga berasal dari teori tentang Scrum. Kebutuhan tersebut meliputi Tugas Backlog yang berisi tugas tugas yang harus diselesaikan tim, Sprint yang berjalan pada rentang waktu yang telah ditetapkan, Scrum Board untuk mengubah status tugas

4.2. Analisis Pengguna Aplikasi

Tabel 1 Analisis Pengguna

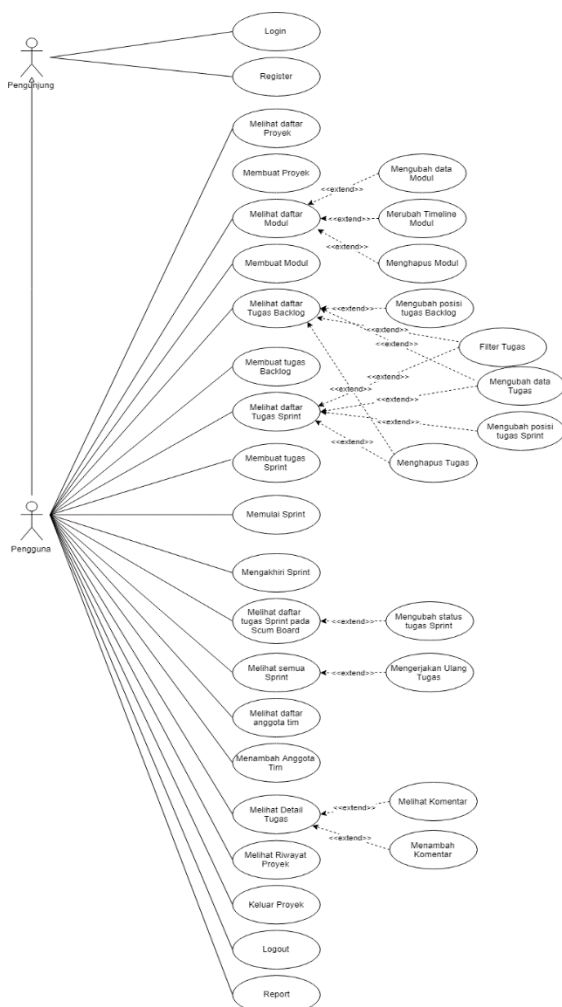
No	Jenis Pengguna	Penjelasan
1.	Pengunjung	Merupakan pengguna aplikasi yang belum login pada aplikasi. Pengunjung hanya dapat menggunakan fitur login dan register

2. Pengguna Adalah pengunjung aplikasi yang berhasil daftar dan login ke dalam aplikasi Manajemen Multi Proyek. Pengguna dapat menjalankan atau memiliki hak akses terhadap semua fungsi yang ada pada aplikasi Manajemen Multi Proyek.

Pada aplikasi Manajemen Multi Proyek ini terdapat beberapa pengguna antara lain, pengguna dan pengunjung. Untuk penjelasan dari pengguna tersebut akan dipaparkan pada Tabel 1.

4.3. Use Case Diagram

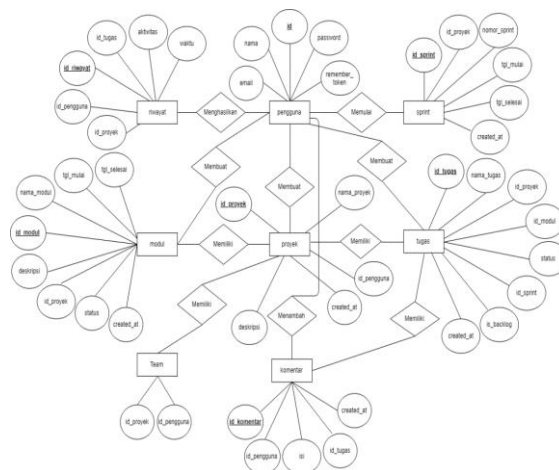
Hasil dari keseluruhan kebutuhan akan dibentuk sebagai suatu *Usecase diagram*. Yang berisi *case*. *Usecase diagram* berisi gambaran aplikasi dari sisi luar yang berisi aktifitas tiap aktor pada aplikasi, dan ditampilkan pada Gambar 2.



Gambar 2 *Usecase Diagram*

4.3. Analisis Data

Bagian ini dilakukan guna mendapatkan identifikasi entitas yang diperlukan pada aplikasi manajemen multi proyek menggunakan metode scrum. Analisis data akan direpresentasikan pada bentuk *Entity Relationship Diagram*.



Gambar 3 *Entity Relationship Diagram*

Dalam gambar ERD tersebut terdapat delapan entitas meliputi pengguna, proyek, modul, tugas, komentar, riwayat, sprint, dan team. Semua entitas tersebut memiliki relasi antar entitas.

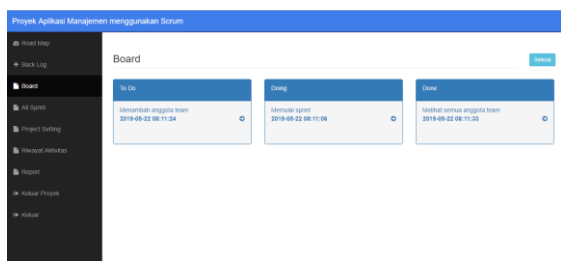
5. PERANCANGAN DAN IMPLEMENTASI

5.1. Perancangan

Pada bagian perancangan akan diuraikan tentang perancangan yang akan dilakukan dari hasil analisis untuk sebelum diteruskan pada tahapan implementasi. Dalam tahap perancangan terdiri dari perancangan data, *sequence diagram*, *class diagram*, perancangan pada algoritme, serta perancangan terhadap antarmuka. Pemodelan *sequence diagram* berguna untuk melakukan pemodelan perilaku antar objek dengan objek lainnya menggunakan diagram alir. Pemodelan *sequence diagram* berdasarkan *use case diagram* dan *use case scenario*. Perancangan pada *class diagram* berguna dalam menunjukkan objek apa saja yang terdapat pada aplikasi dan juga menampilkan atribut serta fungsi yang dimiliki oleh setiap objek. Selain daripada itu, dalam perancangan sistem menghasilkan perancangan terhadap algoritme dan antarmuka.

5.2. Implementasi

Implementasi berjalan dan dilakukan sesuai dengan hasil perancangan yang dibuat, yang mana meliputi implementasi pada kode program, antarmuka, dan data. Masing masing implementasi disesuaikan dengan hasil perancangannya. Pada implementasi terhadap kode program dilakukan menggunakan PHP sebagai bahasa pemrograman disertai dengan metode *Object-Oriented Programming* dan menggunakan *framework* Laravel. Untuk implementasi pada bagian antarmuka menggunakan HTML, CSS, serta JavaScript. Sedangkan pada implementasi data dilakukan menggunakan basis data MySQL.



Gambar 4 Implementasi Halaman Board

6. PENGUJIAN

Pada bagian pengujian, akan dilakukan pemeriksaan kepada hasil implementasi aplikasi yang telah dilakukan untuk mengetahui bahwa sistem sudah sesuai dengan analisis kebutuhan dan perancangan atau belum sesuai. Pengujian adalah bagian yang dilakukan setelah melakukan implementasi. Terdapat 4 bagian pengujian, yaitu berupa pengujian unit, integrasi, pengujian validasi, dan pengujian compatibility.

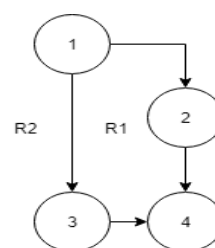
6.1. Pengujian Unit

Pengujian unit pada kasus ini diterapkan pada fungsi `buat_proyek()` diujikan berdasarkan pseudocode telah dibuat yang mana berisi algoritma pada implementasi kode program aplikasi. Untuk pseudocodenya tertulis pada Tabel 2, kemudian pseudocode tersebut digambarkan dalam suatu *flow graph* yang menjelaskan alur dari kode tersebut untuk menghitung basis path pengujian, seperti yang dapat dilihat pada Gambar 5. Selanjutnya, untuk hasil pengujian berada pada Tabel 3.

Tabel 2 Pseudocode Method `buat_proyek`

```

Start method buat_proyek. (1)
If nilai form nama_proyek == null
    Kembali kehalaman semua proyek
End if
Simpan proyek baru dalam database
Kembali kehalaman semua proyek
End (4)
  
```



Gambar 5 Flow Graph Pengujian `buat_proyek`

Berdasarkan gambar diatas dapat dilakukan perhitungan terhadap total *cyclomatic complexity* yang diperoleh untuk menentukan banyaknya kasus uji yang nantinya akan dilakukan untuk memeriksa unit dari fungsional.

- $V(G) = 2$, terdapat 2 region
 - $V(G) = 4 \text{ edges} - 4 \text{ nodes} + 2 = 2$
 - $V(G) = 1 \text{ predicate nodes} + 1 = 2$.
- Selanjutnya akan diperoleh *independent path*
- Jalur ke-1 = 1 – 2 – 4
 - Jalur ke-2 = 1 – 3 – 4

Hasil pengujian dari method `buat_proyek()` untuk tiap jalurnya ditampilkan pada Tabel 3.

Tabel 3 Hasil Pengujian Unit method `buat_proyek`

No. Jalur	Prosedur Uji	Expected Result	Result	Status
1	Field <code>nama_proyek</code> pada form tidak berisi nilai.	Aplikasi akan kembali memuat halaman semua proyek.	Aplikasi kembali memuat halaman semua proyek.	Valid
2	Field <code>nama_proyek</code> pada form memiliki suatu nilai.	Aplikasi akan menyimpan proyek dan	Aplikasi berhasil menyimpan proyek dan	Valid

kembali kehalam an semua proyek.	kembali kehalam an semua proyek.
---	---

6.2. Pengujian Integrasi

Pengujian ini dijalankan dengan memeriksa unit atau komponen yang melakukan interaksi dalam sistem guna menghasilkan suatu fungsional. Pengujian integrasi ini dilakukan dengan cara memakai sampel dari kebutuhan fungsional menambah komentar. Untuk menjalankan kebutuhan fungsional ini, method `beri_komentar` pada kelas `ProyekController` terintegrasi dengan method `create` yang terdapat pada kelas `M_komentar`.

Tabel 4 Hasil Pengujian Integrasi

Input Pertama	Komentar tentang suatu tugas
Method dari kelas <code>ProyekController</code>	<code>beri_komentar()</code>
Output	Komentar tentang suatu tugas
Method dari kelas <code>M_komentar</code>	<code>create()</code>
Expected Result	Sistem menyimpan komentar dan memuat ulang halaman detail tugas yang dikomentari oleh pengguna
Result	Sistem berhasil menyimpan komentar dan memuat ulang halaman detail tugas yang dikomentari oleh pengguna
Status	Valid

6.3. Pengujian Validasi

Pengujian ini berfungsi dalam hal memastikan apakah perangkat lunak yang telah dikembangkan sudah benar dan sesuai berdasarkan daftar kebutuhan yang sebelumnya telah didefinisikan. Pada pengujian ini menggunakan metode *black-box* yaitu jenis *scenario-based testing*.

Tabel 5 Hasil Pengujian Validasi Register

Nama Kasus	Kasus Uji untuk Mendaftar
Kode	MMPS_F_01
Prosedur	1. Mengunjungi halaman register

2.	Mengisikan kolom nama = "Andri Wijaya"
3.	Mengisikan kolom email = "wijayakusuma@gmail.com"
4.	Memasukkan password = "andri123"
5.	Memasukkan confirm password= "andri123"
6.	Klik tombol Register
Hasil yang diharapkan	Aplikasi akan memunculkan pesan bahwa pengguna berhasil daftar
Hasil	Aplikasi memunculkan pesan bahwa pengguna berhasil daftar
Status	Valid

6.4. Pengujian Compatibility

Pengujian *Compatibility* merupakan pengujian yang dilakukan guna mendapatkan informasi dan memeriksa apakah aplikasi manajemen multi proyek menggunakan metode scrum dapat berjalan atau diakses pada beberapa browser dan versi *browser*-nya, serta pada beberapa platform. Dalam pengujian berikut ini, dilakukan dengan menggunakan bantuan pihak ketiga, yaitu aplikasi *SortSite* versi 5. Pengujian dilakukan kepada 8 browser yang total keseluruhan mencapai 11 versi browser, dan 3 platform. Dari hasil pengujian didapatkan jika aplikasi dapat beroperasi dan diakses dengan optimal pada 9 dari 11 versi browser. Terdapat tiga tingkatan untuk permasalahan yang muncul, yaitu *Minor*, *Major*, dan *Critical Issues*. Untuk tingkatan *Critical* terjadi pada 2 browser, yaitu *firefox* dan *chrome* yang terjadi karena pada kedua browser tersebut tidak menggunakan *HTTPS* pada aplikasi. Namun, jika aplikasi diakses secara langsung, akan tetap berjalan optimal tanpa timbul masalah terhadap fungsionalitas yang tidak berjalan semestinya walaupun tidak menggunakan *HTTPS*. Selain itu terdapat satu masalah pada tingkatan *Critical* serta delapan masalah pada tingkatan *Minor* dari hasil pengujian aplikasi yang berkaitan dengan penulisan sintaks pada kode *CSS*.

7. KESIMPULAN

Dari hasil penelitian tentang aplikasi manajemen multi proyek menggunakan metode *scrum*, untuk kebutuhan didapatkan total 33 kebutuhan untuk fungsional dan 1 untuk *non fungsional*. Kebutuhan non fungsional yang dimaksud berkaitan dengan aspek *compatibility* dari aplikasi. Pada proses analisis kebutuhan,

dilakukan dengan memodelkan dalam diagram *use case* dan dijelaskan pada skenario *use case*.

Berdasarkan hasil dari perancangan aplikasi manajemen multi proyek menggunakan metode *scrum*, diperoleh beberapa perancangan meliputi, pemodelan *sequence diagram* yang berisi penjelasan tentang urutan dari proses aplikasi serta interaksi yang terjadi pada objek di dalam tiap fungsionalitas aplikasi. Pemodelan *class diagram* di dalamnya memuat kelas kelas yang digunakan pada proses pengembangan aplikasi, dalam kasus ini menggunakan arsitektur MVC. Perancangan algoritme yang menghasilkan pseudocode untuk nantinya diimplementasikan pada kode program. Perancangan antarmuka yang berisi desain dari antarmuka dan penjelasan pada setiap komponennya. Sedangkan untuk perancangan data dilakukan pemodelan dalam bentuk *Entity Relationship Diagram*.

Berdasarkan hasil dari implementasi aplikasi, diperoleh spesifikasi pengembangan aplikasi pada bagian perangkat lunak dan perangkat keras yang dipakai penulis dalam melakukan proses implementasi berdasar perancangan dan pemodelan yang telah dilakukan. Untuk implementasi dari kode berasal dari proses dari proses melakukan perancangan algoritme, implementasi antarmuka berasal dari proses melakukan perancangan terhadap antarmuka, dan implementasi data berasal dari proses melakukan perancangan terhadap data.

Untuk hasil pengujian aplikasi, diperoleh hasil valid untuk pengujian unit menggunakan *white box testing* untuk menguji keseluruhan jalur uji. Untuk pengujian bagian integrasi didapatkan hasil bernilai valid pada proses pengujian. Untuk pengujian validasi dilakukan dengan metode *black box testing*, diperoleh hasil valid terhadap keseluruhan kasus uji. Untuk pengujian *compatibility* diperoleh hasil jika aplikasi dapat beroperasi dan diakses dengan baik pada 9 dari 11 versi browser.

Saran untuk pengembangan selanjutnya adalah penambahan fitur notifikasi kepada pengguna untuk aktivitas yang terjadi, dan memberi informasi jika terdapat pekerjaan yang akan mencapai *deadline* pengerjaan serta membuat tampilan antarmuka menjadi lebih interaktif.

8. DAFTAR PUSTAKA

Carneiro, L. B., Silva A. C. C. L. M. dan

Alencar, L. H, 2018. *Scrum Agile Project Management Methodology Application for Workflow Management: A Case Study*, [e-journal]. Tersedia di: <<https://ieeexplore.ieee.org/document/8607356>> [Diakses 28 Februari 2019].

Farrel-Vinay. (2008). *Manage Software Testing*. Auerbach Publications.

G. Cloke, "Get Your Agile Freak On! Agile Adoption at Yahoo! Music", in *AGILE 2007*, 2007, pp. 240-248.

Husen, Abrar. 2009, *Manajemen Proyek*. Yogyakarta: Andi Offset.

K. Schwaber and M. Beedle, "Agile software development with Scrum", in *Series in agile software development* Upper Saddle River, NJ: Prentice Hall, 2002, pp. 31-56.

Ken, S. & Jef, S., 2017. *Panduan Scrum*. s.I.:ScrumOrg & ScrumInc.

Marchenko, A dan Abrahamsson, P, 2008. *Scrum in a Multiproject Environment: An Ethnographically-Inspired Case Study on the Adoption Challenges*, [e-journal]. Tersedia di: <<https://ieeexplore.ieee.org/document/4599449>> [Diakses 13 Januari 2019].

Oza, N., Fagerholm, F. dan Münch, J, 2013. *How Does Kanban Impact Communication and Collaboration in Software Engineering Teams?*, [e-journal]. Tersedia di: <<https://ieeexplore.ieee.org/document/6614747>> [Diakses 28 Februari 2019].

Parsons, D., Thorn, R., Inkila, M., dan MacCallum, K, 2018. *Using Trello to Support Agile and Lean Learning with Scrum and Kanban in Teacher Professional Development*, [e-journal]. Tersedia di: <<https://ieeexplore.ieee.org/document/8615399>> [Diakses 28 Februari 2019].

Romano B.L., and Silva A.D, 2015. *Project management using the Scrum agile method: A case study within a small enterprise*, [e-journal]. Tersedia di: <<https://ieeexplore.ieee.org/document/7113578>> [Diakses 13 Januari 2019].

Sommerville, I., 2011. *Software engineering*. 9th ed. London: Addison-Wesley.

Srivastava A., Saraswat S., dan Bhardwaj S, 2017. *SCRUM model for agile*

methodology, [e-journal]. Tersedia di:
<<https://ieeexplore.ieee.org/document/8229928>> [Diakses 13 Januari 2019].

Wagh, R, 2012. *Using Scrum for Software Engineering Class Projects*, [e-journal]. Tersedia di:
<<https://ieeexplore.ieee.org/document/6170011>> [Diakses 28 Februari 2019]