Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer e-ISSN: 2548-964X
Vol. 2, No. 12, Desember 2018, hlm. 6627-6634 http://j-ptiik.ub.ac.id

Pengembangan Aplikasi Manajemen Proyek Perangkat Lunak Berbasis Scrum Studi Kasus CV. Nusantara Media Mandiri (CVNMM)

Muhammad Faisal Fahat[1], Bayu Priyambadha[2], Fajar Pradana[3]

Program Studi Teknik Informatika, Fakultas Ilmu Komputer, Universitas Brawijaya
Email: [1]mfaisalfahat@gmail.com, [2]bayu_priyambadha@ub.ac.id, [3]fajar.p@ub.ac.id

Abstrak

Penanggung jawab proyek yang dalam satu waktu dapat menangani tiga sampai lima proyek sekaligus, mempunyai permasalahan ketika memantau pekerjaan proyek, dikarenakan dalam satu proyek terdapat minimal 30 pekerjaan yang harus dikelola. Dengan semua pekerjaan tersebut, ada pekerjaan yang tidak terpantau, sehingga tidak terselesaikan sesuai jadwal. Terlebih CV. Nusantara Media Mandiri (CVNMM) menggunakan *waterfall* didalam pengembangan perangkat lunaknya. Padahal CVNMM memiliki kebijakan dimana perangkat lunak yang selesai dikerjakan, akan mendapatkan garansi (pelatihan penggunaan perangkat lunak, perbaikan, dan perubahan sistem) selama 2 bulan terhitung setelah perangkat lunak diterima oleh klien. Akibat kebijakan tersebut, pada bulan Januari - Maret 2017 saja ada 4 dari 7 kliennya yang meminta adanya perubahan sistem pada perangkat lunak yang sudah selesai dikerjakan. Oleh sebab itu, CVNMM ingin mengembangkan aplikasi manajemen proyek perangkat lunak guna membantu dalam pemantauan pekerjaan didalam proyek, selain itu untuk menangani perubahan sistem, CVNMM dapat menggunakan pendekatan *agile* yang populer, yaitu scrum. Karena kerangka kerja ini, memungkinkan klien untuk melakukan ulasan terhadap potongan atau bagian dari perangkat lunak, ketika pengembang selesai melakukan iterasi, sehingga tidak perlu menunggu hingga perangkat lunak selesai dikerjakan secara utuh. Aplikasi manajemen proyek perangkat lunak CVNMM memiliki 38 kebutuhan fungsional, dimana semua kebutuhan fungsional tersebut telah valid didalam pengujian *black-box.*

Kata kunci: *scrum, agile, waterfall, manajemen proyek, pengembangan perangkat lunak*

Abstract

Project manager who can handle three to five projects at once, has problems to monitor works of the projects, because in one project there are at least 30 works to be managed. With all of those works, there are works that are not monitored, so they are not completed on schedule. Moreover, CV. Nusantara Media Mandiri (CVNMM) still use waterfall approach in software development. Whereas CVNMM has a policy whereby software that has been completed by the project team, it will get a warranty (software usage training, repair, and changes to the system) for 2 months after the software received by the client. Because of this warranty, especially in case of system changes, just in January - March 2017 there are 4 of 7 clients who request a system change on the software that has been completed by the project team. Therefore, CVNMM wants to develop a software project management application to assist in monitoring works of the project, in addition to dealing with system changes, CVNMM can use the popular agile approach, which is scrum. Because this framework allows the client to review a piece of software, when the developer has finished iterating, so there's no need to wait until the software is completely done. CVNMM's software project management application have 38 functional requirements, where all of the functional requirements have valid in black-box testing.

Keywords: *scrum, agile, waterfall, project management, software development*

1. PENDAHULUAN

Mengelola proyek perangkat lunak bukanlah hal yang mudah, seperti halnya membuat sesuatu yang tidak nampak menjadi nampak, selain itu proyek perangkat lunak memiliki 3 karakteristik yang membedakannya dengan proyek lain, yaitu *invisibility*, *complexity*, dan *flexibility* (Hughes & Cotterell, 1999). Menurut survei yang dilakukan oleh Standish Group pada tahun 2014, menunjukkan

bahwa tingkat kesuksesan proyek perangkat lunak sangatlah rendah, yaitu sekitar 16,2% dari 8.380 proyek, dimana 52,7% dari proyek akan menambah biaya sampai 189% dari biaya awal (Munir, 2015). Kegagalan disebabkan oleh beberapa hal, diantaranya: kurangnya rencana dan estimasi, kurangnya standar dan pengukuran kualitas, kesalahan dalam pengambilan keputusan organisasional, kurangnya teknik dalam membuat progres yang nampak, kurang jelasnya pembagian tugas, serta kesalahan dalam memutuskan kriteria sukses (Munir, 2015)

CV. Nusantara Media Mandiri (CVNMM) merupakan badan usaha di bidang pengembangan perangkat lunak. CVNMM memiliki kebijakan dimana perangkat lunak yang telah selesai dikerjakan oleh tim proyek dan kemudian diserahkan kepada klien, akan mendapatkan garansi selama 2 bulan terhitung setelah perangkat lunak diterima oleh klien. Garansi ini mencakup pelatihan penggunaan perangkat lunak, perbaikan, dan perubahan pada sistem. Namun dengan diberlakukannya kebijakan ini, dimana CVNMM menjamin adanya pelayanan terhadap perubahan sistem, pada bulan Januari - Maret 2017 saja sudah ada 4 dari 7 kliennya yang meminta adanya perubahan sistem pada perangkat lunak.

Terlebih saat ini CVNMM masih menggunakan pendekatan *waterfall* dalam pengembangan proyeknya. Apabila ada perubahan yang terjadi pada sistem, maka harus dilakukan perubahan pada semua fase yang ada pada model *waterfall* (Adenowo & Adenowo, 2013). Dengan menggunakan pendekatan ini klien hanya akan mengetahui bagaimana perangkat lunak yang mereka pesan, ketika perangkat lunak sudah melalui tahap implementasi (Datyal, 2015). Hal ini juga menjadi salah satu pemicu adanya perubahan kebutuhan, karena apa yang dibayangkan oleh klien, ternyata tidak tertangkap pada saat pengumpulan kebutuhan sistem.

Penanggung jawab proyek yang dalam satu waktu dapat menangani tiga sampai lima proyek, mengalami permasalahan ketika ingin memantau perkerjaan dimasing-masing proyek, karena tidak adanya alat bantu guna memantau pekerjaan, maka ada pekerjaan yang tidak terpantau sehingga pekerjaan tersebut tidak terselesaikan tepat waktu dan mempengaruhi jadwal pengerjaan proyek. Padahal dalam satu proyek terdapat minimal 30 pekerjaan yang harus dikelola. Oleh sebab itu, untuk kedepannya CVNMM ingin menerapkan aplikasi pengelola proyek perangkat lunak, guna membantu mengorganisir semua proyek yang sedang mereka kerjaan. Menurut Maatita (Maatita, et al., 2011) dalam penelitiannya, aplikasi manajemen proyek perangkat lunak dapat dikatakan baik, apabila secara fleksibel dapat sesuai dengan kebutuhan dari tim proyek. Hal tersebut turut mendorong CVNMM untuk mulai mengembangkan aplikasi manajemen proyek perangkat lunaknya sendiri.

Dalam rangka memenuhi keinginan pengguna, khususnya dalam hal perubahan sistem, maka CVNMM dapat menggunakan pendekatan *Agile* didalam mengembangkan perangkat lunak. Metodologi *Agile* merupakan alternative terhadap manajemen proyek secara tradisional, metodologi ini berfokus pada kaloborasi antara pengembang dan klien, mendukung pengiriman awal produk, meningkatkan kualitas proyek, dan meningkatkan kepuasan klien (Vijay & Ganapathy, 2014). Pada saat ini, scrum merupakan pendekatan *agile* yang populer untuk mengelola proyek perangkat lunak, kerangka kerja ini memungkinkan klien untuk melakukan ulasan terhadap potongan atau bagian dari perangkat lunak yang dipesan pada saat pengembang selesai melakukan iterasi, dan apabila klien menginginkan adanya perubahan, maka tidak perlu menunggu perangkat lunak telah diimplementasi secara utuh, karena perubahan tersebut dapat dilakukan pada potongan perangkat lunak tersebut di iterasi selanjutnya (Sultana, 2006).

2. SCRUM

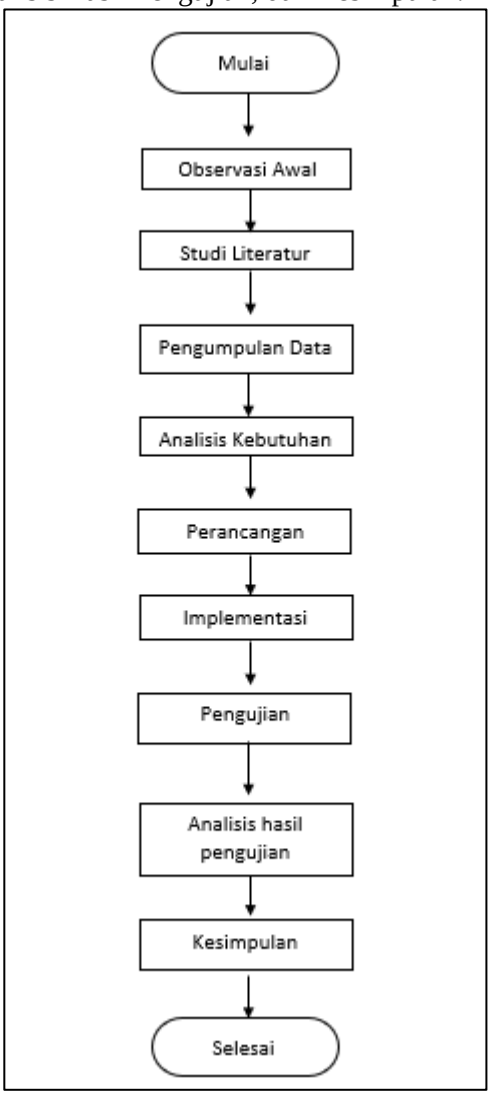
Scrum merupakan suatu kerangka kerja yang memungkinkan penyelesaian masalah yang berubah-ubah dan kompleks, disisi lain produk yang dihasilkan akan tetap memiliki nilai yang tinggi baik secara produktif maupun kreatif (Ken & Jeff, 2013). Pada sekitar tahun 1990 Scrum sudah mulai digunakan untuk mengembangkan produk yang kompleks. Scrum bukan merupakan proses atau teknik didalam mengembangkan produk, tetapi sebuah kerangka kerja yang dapat memuat berbagai proses dan teknik didalamnya. Scrum memiliki 3 sifat, yaitu: ringan, mudah dipahami, namun sulit dikuasai.

Teori kontrol proses empiris merupakan dasar dari Scrum, atau dapat disebut dengan empirisme yang berfokus pada pengetahuan

yang didapat dari pengalaman dan pembuatan keputusan berdasar pada pengetahuan yang dimiliki. Selain itu, untuk menekan resiko dan meningkatkan prediktabilitas Scrum menerapkan pendeketan *Iterative* (berkala) dan *Incremental* (bertahap). Ada tiga hal penting didalam implementasi kontrol proses empiris yaitu: transparansi, inspeksi, dan adaptasi (Ken & Jeff, 2013).

3. METODOLOGI PENELITIAN

Dalam melakukan penelitian ini, tahapan-tahapan yang digunakan ditunjukkan oleh Gambar 1. Dimana ada beberapa tahapan, diantaranya: Observasi Awal, Studi Literatur, Pengumpulan Data, Analisis Kebutuhan, Perancangan, Implementasi, Pengujian, Analisis Hasil Pengujian, dan Kesimpulan.



Gambar 1. Diagram Alir Metodologi Penelitian

3.1. Observasi Awal

Didalam observasi awal, dilakukan untuk menggali permasalahan yang terdapat didalam CV. NUSANTARA MEDIA MANDIRI (CVNMM) terutama dalam hal aplikasi manajemen proyek perangkat lunak. Selain itu dengan melakukan pengamatan secara langsung, maka peneliti dapat mengetahui bagaimana mekanisme manajemen proyek perangkat lunak di CVNMM. Sehingga, permasalahan dan informasi yang mendukung terkait topik penelitian akan dikumpulkan pada tahap ini.

3.2. Studi Literatur

Pada tahap ini, dilakukan pemahaman terhadap kepustakaan yang mendukung penyelesaian masalah. Studi literatur dilakukan dengan cara membaca dan mempelajari buku, jurnal, dan artikel ilmiah. Studi ini bertujuan untuk mendapatkan informasi terkait pengembangan aplikasi manajemen proyek perangkat lunak. Selain itu, dilakukan pula studi literatur terkait kerangka kerja dan teknologi yang diperlukan didalam mengembangkan aplikasi manajemen proyek perangkat lunak. Sehingga fokus atau pokok bahasan didalam studi literatur adalah sebagai berikut: proyek, manajemen proyek, rekayasa perangkat lunak, software development life cycle (SDLC) dengan pendekatan waterfall, kerangka kerja SCRUM, pemodelan Unified Modeling Language (UML), dan pengujian perangkat lunak.

3.3. Pengumpulan Data

Pengumpulan data kebutuhan untuk aplikasi manajemen proyek perangkat lunak cvnmm, dilakukan dengan cara wawancara. Wawancara itu sendiri, dilakukan dengan tujuan untuk mendapatkan jawaban secara langsung dari responden melalui proses tanya jawab, selain itu wawancara yang dilakukan secara langsung atau tatap muka dengan responden akan meningkatkan kerjasama, serta jawaban dari responden dapat langsung diklarifikasi (Hasibuan, 2007). Pada penelitian ini, akan dilakukan tanya jawab kepada penanggung jawab proyek perangkat lunak di CV. NUSANTARA MEDIA MANDIRI dengan tujuan untuk melakukan penggalian terhadap kebutuhan dari aplikasi yang akan dikembangkan.

3.4. Analisis Kebutuhan

Berdasarkan hasil pengumpulan data

kebutuhan yang telah dilaksanakan sebelumnya, akan dilakukan proses penerjemahan kebutuhan dari aplikasi kedalam pemodelan kebutuhan perangkat lunak. Pemodelan yang digunakan adalah pemodelan kebutuhan berorientasi objek yang memanfaatkan Diagram UML. Dan diagram yang digunakan adalah *Usecase Diagram* dan *Usecase Scenario*. *Usecase Diagram* digunakan untuk mengambarkan apa saja aktor yang ada pada Aplikasi Manajemen Proyek Perangkat Lunak pada CV. NUSANTARA MEDIA MANDIRI, serta dapat menggambarkan bagaimana hak atau otorisasi yang dimiliki oleh aktor terhadap aplikasi. Diagram ini dapat memberikan gambaran perbedaan hak yang dimiliki oleh tiap aktor, atau bagaimana tiap aktor berinteraksi dengan aplikasi.

3.5. Perancangan

Pada tahap ini akan dilakukan perancangan yang nantinya digunakan sebagai landasan dari implementasi. Perancangan sendiri berdasar pada hasil analisis kebutuhan. Dan berikut adalah perancangan yang akan dilakukan:

1. *Sequence Diagram*
   Diagram ini digunakan untuk mengambarkan bagaimana suatu objek berinteraksi dengan objek lainnya dan bagiamana antar objek saling bertukar pesan pada waktu tertentu.
2. *Class Diagram*
   Diagram ini digunakan untuk memetakan *class-class* yang ada pada aplikasi manajemen proyek perangkat lunak CV. NUSANTARA MEDIA MANDIRI. Dari diagram ini, akan terlihat apa saja perilaku dan atribut yang dimiliki oleh tiap-tiap *class* yang ada.
3. Perancangan Komponen
   Pada tahap ini, dilakukan perancangan algoritme method pada sebuah class yang nantinya akan digunakan sebagai landasan didalam melakukan implementasi. Algoritme yang telah dirancang, dituangkan kedalam bentuk *pseudocode*.
4. Perancangan Data
   Untuk dasar dari database, maka dibuat perancangan data yang memanfaatkan *Physical Data Model* yang menunjukkan relasi antar tabel serta kolom yang dimiliki oleh setiap tabel.
5. Perancangan Antarmuka
   Perancangan antarmuka direpresentasikan menggunakan mockup, sehingga dapat menunjukkan apa saja komponen yang ada pada antarmuka, selain itu perancangan antarmuka juga dapat menggambarkan bagaimana tata letak dari masing-masing komponen.

3.6. Implementasi

Pada tahap ini, dilakukan proses penerjemahan perancangan kedalam bahasa pemrograman, sehingga nantinya dapat dimengerti oleh komputer. Bahasa pemrograman yang dipakai pada penelitian ini adalah bahasa pemrograman Java. Sehingga algoritme yang sudah dirancang didalam perancangan, nantinya akan diterjemahkan kedalam *syntax* bahasa java. Perancangan antarmuka yang telah direpresentasikan menggunakan *mockup*, akan diimplementasikan kedalam *graphical user interface* (GUI) milik Bahasa Java yaitu Java Swing. Untuk *database*, DBMS yang akan digunakan adalah MySQL, implementasi *database* didasarkan pada perancangan *physical data model* (PDM).

3.7. Pengujian

Pada tahap ini, dilakukan pengujian yang memiliki tujuan untuk memverifikasi dan memvalidasi sistem, sehingga sistem yang dibuat dapat sesuai dengan kebutuhan. Pengujian ini dilakukan dengan dua pendekatan pengujian, yaitu : *White Box Testing* dan *Black Box Testing*. Pada *White Box Testing* akan dilakukan pengujian jalur dasar (*Basis Path Testing*) pada algoritme yang telah dirancang. Tujuan pengujian jalur dasar adalah untuk mengukur kompleksitas dari algoritme. Sedangkan *Black Box Testing* akan dilakukan *Requirement Testing*. Tujuan dari pengujian *Requirement Testing* adalah untuk mengetahui bahwa aplikasi sudah sesuai dengan daftar kebutuhan, serta aplikasi sudah berjalan dengan valid.

3.8. Analisis Hasil Pengujian

Setelah dilakukan pengujian terhadap Aplikasi Manajemen Proyek Perangkat Lunak pada CV. NUSANTARA MEDIA MANDIRI, maka selanjutnya akan dilakukan analisis terhadap hasil pengujian *basis path testing* dan *requirement testing*. Hasil dari analisis pengujian akan menjadi salah satu bahan didalam menyimpulkan hasil penelitian yang telah dilakukan.

3.9. Kesimpulan

Pada tahap ini didapatkan suatu kesimpulan yang didapat dari pengujian serta analisis terhadap aplikasi manajemen proyek perangkat lunak pada CV. NUSANTARA MEDIA MANDIRI. Selain itu, peneliti juga akan memberikan saran untuk penelitan kedepannya.

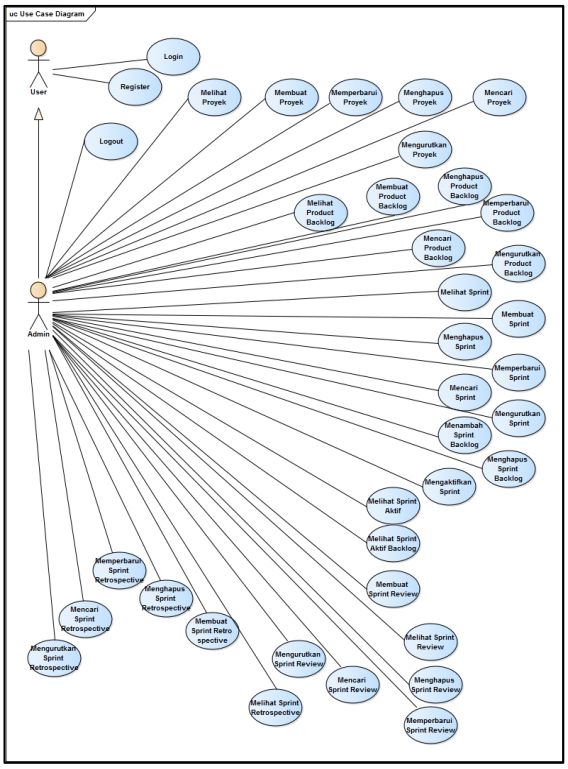
4. ANALISIS KEBUTUHAN

Didalam mengembangkan perangkat lunak, perlu dilakukan penggalian informasi terkait kebutuhan dari pengguna. Kebutuhan yang terkumpul nantinya akan menjadi dasar dari perancangan perangkat lunak. Dan Tabel 1. berikut adalah aktor-aktor yang ada pada sistem manajemen proyek perangkat lunak:

Tabel 1. Tabel identifikasi pengguna

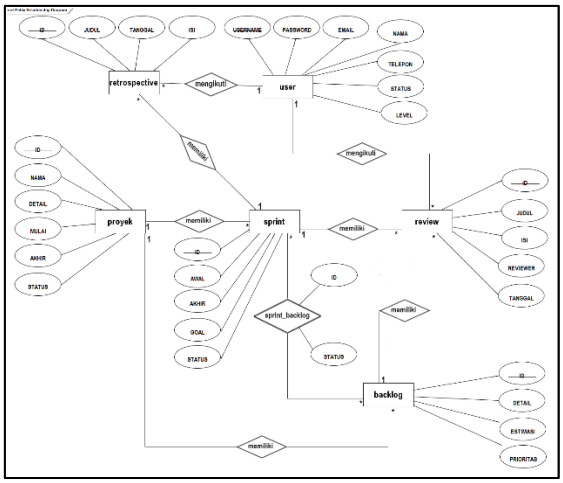
No.	Identifikasi Pengguna	Karakteristik
1	User	Merupakan pengguna yang belum memiliki hak akses untuk menggunakan fitur-fitur dari aplikasi, agar dapat mengakses fitur maka user perlu untuk melakukan registrasi terlebih dahulu, setelah terdaftar maka user dapat *login* kedalam aplikasi, dan dapat pula *logout* dari aplikasi.
2	Admin	Admin merupakan pengguna yang dapat membuat proyek, memperbarui data proyek, menghapus proyek, mencari data proyek, mengurutkan data proyek berdasar pada kategori pengurutan. Admin berperan dalam mendefinisikan bagaimana produk yang akan dihasilkan, menyusun item-item yang ada pada product backlog. Selain itu, product owner juga berperan dalam penyusunan sprint, serta memilih sprint mana yang akan aktif dikerjakan. Memberikan review kepada kepada sprint, serta dapat menambahkan sprint retrospective.

Sistem ini memiliki 38 Kebutuhan Fungsional dan 1 Kebutuhan Non-Fungsional. Dan pada Gambar 2. berikut ini, merupakan pemodelan use case dari sistem:



Gambar 2. *Use Case Diagram*

Untuk analisis data dimodelkan menggunakan *Entity Relationship Diagram* atau disebut juga dengan ERD, diagram analisis data ditunjukkan pada Gambar 3. berikut ini:



Gambar 3. *Entity Relationship Diagram*

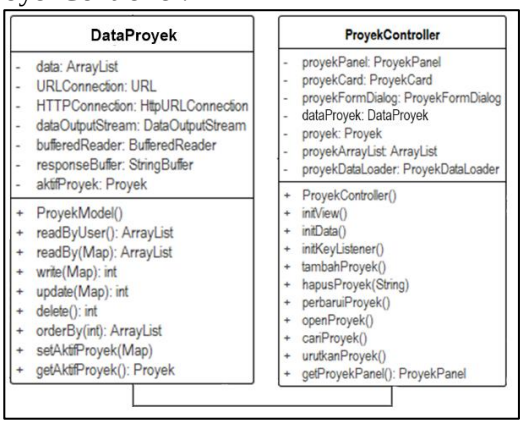
5. PERANCANGAN

4.1. Perancangan *Sequence Diagram*

Terdapat 3 *Sequence Diagram* yang dicantumkan didalam penelitian ini, diantaranya: Membuat Proyek *Sequence Diagram*, Memperbarui Proyek *Sequence Diagram*, Menghapus Proyek *Sequence Diagram*. Dimana setiap *Sequence Diagram* memiliki *boundary* yang berhungan dengan aktor, *controller* yang bertugas sebagai jembatan antara *boundary* dengan *entity*, dan *entity* yang merepresantikan model. *Sequence Diagram* sendiri memiliki kegunaan untuk menampilkan bagaiaman interaksi antar objek pada satu waktu tertentu

4.2. Perancangan *Class Diagram*

Class pada penelitian ini digolongkan menjadi 3 bagian, yaitu: *view*, *controller*, dan *model*. Dan Gambar 4. berikut adalah class diagram yang menggambarkan hubungan antara *class* DataProyek dengan *class* ProyekController:



Gambar 4. Relasi antara DataProyek dan ProyekController

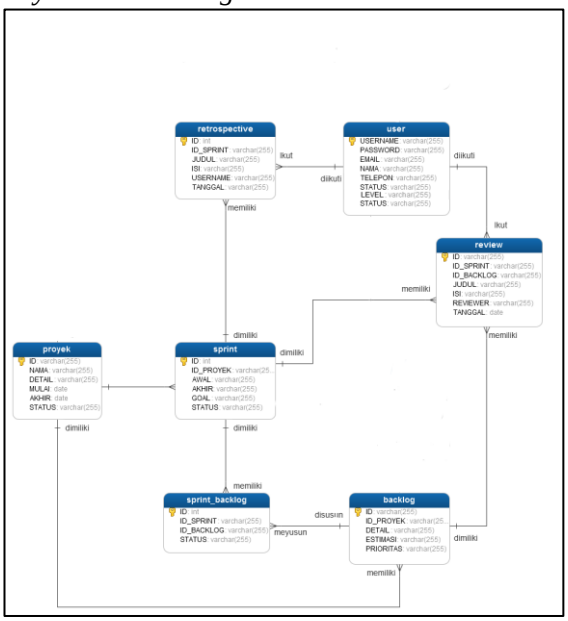
4.3. Perancangan Komponen

Terdapat 5 perancangan algoritme yang dicantumkan dalam penelitian ini, diantaranya:

1. Algoritme method tambahProyek pada Class ProyekController yang digunakan untuk menambahkan proyek.
2. Algoritme perbaruiProyek pada Class ProyekController yang digunakan untuk memperbarui data proyek.
3. Algoritme method hapusProyek pada Class ProyekController digunakan untuk menghapus proyek berdasar pada id proyek.
4. Algoritme method write pada Class DataProyek yang digunakan untuk menyimpan data proyek pada database.
5. Algoritme method update pada Class DataProyek yang digunakan untuk memperbarui data proyek pada database.

4.4. Perancangan Data Model

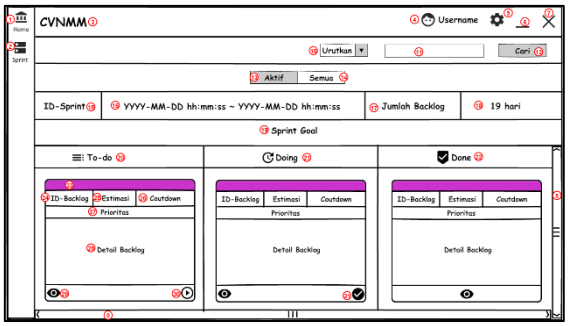
Pada penelitian ini terdapat 7 Tabel diikuti dengan relasi-relasinya. Kesembilan tabel meliputi: user, proyek, backlog, sprint, sprint_backlog, review, retrospective. Dan Gambar 5. berikut ini merupakan diagram dari *Physical Data Diagram*:



Gambar 5. *Physical Data Diagram*

4.5. Perancangan Antarmuka

Perancangan antarmuka merupakan dasar dari implementasi antarmuka. Perancangan antarmuka menjelaskan komponen-komponen yang ada pada rancangan antarmuka. Dan pada Gambar 6. berikut merupakan rancangan antarmuka untuk Sprint Aktif:



Gambar 6. Perancangan Antarmuka Sprint Aktif

Gambar 7. Implementasi Antarmuka Sprint Aktif

6. IMPLEMENTASI

5.1. Implementasi Komponen

Implementasi komponen didasarkan pada perancangan algoritme. Dimana algoritme yang telah dirancang akan diubah kedalam *syntax* Bahasa Java. Sehingga pada penelitian ini, dilakukan implementasi pada algoritme:

1. Implementasi method tambahProyek dalam Class ProyekController yang didasarkan pada perancangan algoritme method tambahProyek.
2. Implementasi dari method perbaruiProyek dalam Class ProyekController yang didasarkan pada perancangan algoritme method perbaruiProyek.
3. Implementasi dari method hapusProyek dalam Class ProyekController yang didasarkan pada perancangan algoritme method hapusProyek.
4. Implementasi dari method write dalam Class DataProyek yang didasarkan pada perancangan algoritme method write
5. Implementasi dari method update dalam Class DataProyek yang didasarkan pada perancangan algoritme method update

5.2. Implementasi Antarmuka

Implementasi antarmuka berdasar pada *mock-up* yang telah dirancang pada saat tahap perancangan. Dan pada Gambar 7. berikut adalah implementasi dari antarmuka Sprint Aktif yang disarkan pada perancagan antarmuka Sprint Aktif Gambar 6.

7. PENGUJIAN

Pada tahap ini, dilakukan pengujian yang memiliki tujuan untuk memverifikasi dan memvalidasi sistem, sehingga sistem yang dibuat dapat sesuai dengan kebutuhan. Pengujian ini dilakukan dengan dua pendekatan pengujian, yaitu : *White Box Testing* dan *Black Box Testing*. Pada *White Box Testing* dilakukan pengujian jalur dasar (*Basis Path Testing*) pada fungsional – fungsional utama. Tujuan pengujian jalur dasar adalah untuk mengukur kompleksitas dari algoritme yang ada pada method.Dari pengujian ini didapatkan nilai *cyclomatic complexity* sebesar 2 untuk method tambahProyek pada Clas ProyekController, untuk pengujian yang dilakukan pada method perbaruiProyek didapatkan nilai *cyclomatic complexity* sebesar 2, dan method hapusProyek juga mendapatkan nilai *cyclomatic complexity* sebesar 2. Sedangkan *Black Box Testing* dilakukan pengujian pada fungsional yang ada. Tujuan dari pengujian tersebut adalah untuk mengetahui bahwa fungsional yang dibuat telah valid. Dari hasil pengujian *black-box testing*, didapatkan 39 pengujian bernilai valid dari 39 test case, dimana 38 pengujian untuk kebutuhan fungsional dan 1 pengujian untuk kebutuhan non-fungsional. Dan pada Tabel 2. berikut adalah hasil pengujian *black-box* pada kebutuhan mencari proyek dengan kode kebutuhan MPPL_F_08.

Tabel 2. Tabel hasil pengujian membuat proyek

Test Case	Expected Result	Result	Status
1.Pilih Menu Proyek 2.Masukkan keyword pencarian 3.Tekan tombol	Aplikasi dapat menampil-kan proyek sesuai dengan	Proyek berhasil ditampil-kan sesuai dengan keyword pencarian	Valid

Cari	keyword masukkan

8. KESIMPULAN

Kesimpulan yang didapat dari penelitian ini, adalah sebagai berikut:

Hasil analisis kebutuhan menunjukkan adanya 2 aktor yang nantinya berinteraksi dengan aplikasi, diantaranya: User dam Admin. Dimana aktor user memiliki 2 kebutuhan fungsional, Admin memiliki 36 kebutuhan fungsional. Sehingga total kebutuhan adalah 39 kebutuhan, terdiri dari 38 kebutuhan fungsional ditambah dengan 1 kebutuhan non fungsional.

Pada perancangan menghasil *Sequence Diagram* yang berguna untuk menampilkan bagaimana interaksi antar objek pada satuan waktu tertentu. Didalam perancangan juga didapatkan adanya 66 Class dimana secara garis besar class akan dikelompokkan menjadi 3, yaitu class view, class controller, dan class model. Selain itu dari perancangan juga dihasilkan perancangan komponen yang bertujuan untuk membuat rancangan algoritme, perancangan antarmuka yang berbentuk *mockup*. Pada tahap implementasi, dilakukan implementasi algoritme yang sudah dirancang pada perancangan komponen, serta dilakukan implementasi antarmuka yang berdasar pada perancangan antarmuka. Selain itu, untuk implementasi basis data didasarkan pada perancangan data yang dimodelkan dalam bentuk *physical data model*.

Pengujian dilakukan menggunakan pendekatan *white-box testing* dan *black-box testing*. Didalam *white-box testing*, dilakukan pengujian terhadap method tambahProyek pada Class ProyekController yang menghasilkan *Cyclomatic Complexity* bernilai 2, pada pengujian yang dilakukan terhadap method perbaruiProyek pada Class ProyekController menghasilkan nilai *Cyclomatic Complexity* sebesar 2, untuk pengujian yang dilakukan terhadap method hapusProyek pada Class ProyekController menghasilkan nilai Cyclomatic Complexity sebesar 2. Sedangkan pada *black-box testing*, didapatkan 39 pengujian bernilai valid dari 39 test case, dimana 38 pengujian untuk kebutuhan fungsional dan 1 pengujian untuk kebutuhan non-fungsional. Dari keseluruhan pengujian maka didapatkan tingkat keberhasilan sebesar 100%.

DAFTAR PUSTAKA

Adenowo, A. A. A. & Adenowo, B. A., 2013. Software Engineering Methodologies: A Review of the Waterfall Model and Object-Oriented Approach. *Inertnational Journal of Scientific & Engineering Research,* 4(7), pp. 427-434.

Datyal, D., 2015. Proposed Model to Overcome the Problems in Waterfall Model. *Recent Innovation in Electronics, Electrical & Computer Engineering (RIEECE) 2015,* 2(2), pp. 29-32.

Hughes, B. & Cotterell, M., 1999. *Software Project Management.* 2nd ed. Berkshire: McGraw-Hill.

Ken, S. & Jeff, S., 2013. *Panduan Scrum.* s.l.:ScrumOrg & ScrumInc.

Maatita, G., Samopa, F. & Wibowo, R., 2011. *PENGEMBANGAN APLIKASI MANAJEMEN PROYEK PERANGKAT LUNAK BERBASIS SPRING: MODUL CORE SYSTEM DAN MANAJEMEN SOURCE CODE.* Surabaya, Institut Teknologi Sepuluh Nopember (ITS).

Munir, 2015. *Manajemen Proyek Perangkat Lunak.* Bandung: UPI Press.

Srivastava, A., Bhardwaj, S. & Saraswat, S., 2017. *SCRUM Model for Agile Methodology.* Greater Nodia, IEEE.

Sultana, J., 2006. *A STUDY ON APPLICABILITY OF THE SCRUM FRAMEWORK FOR LARGE SOFTWARE PROJECTS,* Toronto: Ryerson University.

Vijay, D. & Ganapathy, G., 2014. GUIDELINES TO MINIMIZE THE COST OF SOFTWARE QUALITY IN AGILE SCRUM PROCESS. *International Journal of Software Engineering & Applications (IJSEA),* 5(3), pp. 61-69.