

Pengembangan Aplikasi Manajemen Proyek Perangkat Lunak Kolaboratif Menggunakan Scrum

Bayu Kharisma¹, Nurudin Santoso²

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

Abstrak

Metode manajemen proyek yang sering digunakan saat ini adalah agile dengan kerangka kerja scrum. Agile adalah metode iteratif pengerjaan proyek yang dimana sebuah kebutuhan dari perangkat lunak itu sendiri belum jelas sesuai dengan karakter kebanyakan klien. Scrum adalah metode kolaborasi pengerjaan, yang dimana setiap kebutuhan atau permasalahan dikerjakan dalam bentuk backlog secara tim dalam sirkulasi sprint yang telah ditentukan. Pada PT. Debox Indonesia sebagai objek studi kasus sebagai perwakilan manajemen proyek agile dengan scrum, banyaknya proyek yang diterima membuat manajemen proyek terbengkalai dikarenakan perusahaan ini hanya menggunakan Google Spreadsheet untuk mengelola masing-masing proyek, sehingga sistem informasi yang bisa mencakup manajemen proyek secara agile beserta proses scrum yang meliputi sirkulasi sprint dalam lingkup sebuah organisasi atau perusahaan merupakan solusi dari permasalahan tersebut. Harapan dari sistem informasi ini adalah mudahnya proses manajemen karena semua tercatat dan dapat dilacak dari segi manajemen proyek, scrumming proyek, manajemen klien, dan manajemen organisasi. Hasil pengujian sistem informasi ini adalah seluruh kebutuhan fungsional telah lulus uji dari segi verifikasi dan validasi, dan seluruh kebutuhan non-fungsional telah lulus uji secara manual.

Kata kunci: Manajemen Proyek, Agile, Scrum, Sprint

Abstract

The project management method that is often used today is agile with the Scrum framework. Agile is an iterative method of project work in which a requirement of the software itself is not yet clear according to the character of most clients. Scrum is a collaborative workmanship method, where every requirement or issue is done in the form of a team backlog in a predetermined sprint circulation. At PT. Debox Indonesia as the object of the case study as a representative of the agile project management with Scrum, the number of projects received made the project management neglected because the company only uses Google Sheets to manage each project, so the information system that can include agile project management and scrum processes that covering circulation sprints within the scope of an organization or company is a solution to these problems. The expectation of this information system is the easy management process because everything is recorded and can be tracked in terms of project management, project scrumming, client management, and organizational management. The results of testing this information system are all functional requirements have passed the test in terms of verification and validation, and all non-functional requirements have passed the test manually.

Keywords: Project Management, Agile, Scrum, Sprint

1. PENDAHULUAN

Metode Agile dengan Scrum merupakan hasil dari adaptasi lingkungan proyek perangkat lunak yang cenderung bersifat dinamis, karena pada lingkungan ini dibutuhkan fleksibilitas dan kelincahan yang tinggi. Lingkungan proyek yang bersifat dinamis membuat kebutuhan proyek

berevolusi, harus diselesaikan dengan cepat, kinerja desain yang dikompromikan, dan usaha signifikan untuk memperbarui rencana dan adaptasi terhadap perubahan. Sehingga, metode ini menerapkan proses iterasi pada pengerjaan proyek yang dikerjakan dari awal hingga selesai, dan diulang kembali jika hasil tinjauan pemangku kepentingan dirasa kurang. Scrum

sendiri merupakan kerangka kerja proses Agile untuk pekerjaan kompleks yang tidak bisa diprediksi dengan menerapkan kolaborasi kerja dalam pelaksanaannya (Azanha, et al., 2017).

Berdasarkan hasil observasi penulis terhadap PT. Debox Indonesia, semakin banyak proyek yang diterima dari klien maka semakin susah alokasi waktu serta pembagian tugas kepada anggotanya, dan kebanyakan proyek dari klien masih belum jelas kebutuhannya. Sehingga, organisasi ini menggunakan metode manajemen proyek Agile untuk iterasi berjalannya proyek hingga seluruh kebutuhan dari klien terpenuhi, dan menggunakan kerangka kerja Scrum untuk kolaborasi kerja pelaksanaan proyek. Proses yang dilakukan pada organisasi tersebut masih sederhana, yaitu semua proses dicatat secara manual menggunakan aplikasi web Google Spreadsheet yang bersifat *shared files* agar bisa dibuka bersama antara pimpinan proyek dengan staf yang terlibat dalam proyek, dan pembuatan laporan masih manual menggunakan aplikasi Microsoft Word dengan data terkait laporan dikumpulkan dari Google Spreadsheet.

Permasalahan yang ada dari hasil observasi penulis adalah; pertama, kesulitannya proses pemantauan oleh pimpinan organisasi terhadap setiap proyek yang ada dikarenakan setiap proyek mempunyai file Google Spreadsheet sendiri; kedua, penggunaan waktu yang tidak efektif seperti ada staf yang mengerjakan tugas sudah selesai jauh sebelum batas waktu atau bahkan belum selesai, yang dimana hal ini bisa berakibat fatal seperti penyampaian hasil akhir produk tidak sesuai dengan batas waktu yang sudah dijanjikan kepada klien; ketiga, laporan yang tidak bisa didapatkan secara langsung dan akurat dikarenakan data diambil secara manual dari Google Spreadsheet, sehingga berpengaruh terhadap tingkat kepercayaan klien jika ada data yang tidak sesuai.

Berdasarkan permasalahan yang didapatkan dari hasil observasi penulis, diperlukan solusi berupa sistem informasi manajemen proyek yang menggunakan metode Agile sebagai dasar proses bisnis iterasi menghadapi lingkungan proyek yang dinamis, dan kerangka kerja Scrum sebagai dasar proses bisnis kolaborasi kerja untuk menghadapi pekerjaan kompleks yang tidak bisa diprediksi. Sehingga, tahap pada proses manajemen proyek dari perencanaan, koordinasi, pelaksanaan, dan pemantauan bisa tercatat dan terlacak dengan baik, laporan proyek bisa didapatkan secara langsung dan akurat, dan alokasi sumber daya beserta pemanfaatan waktu

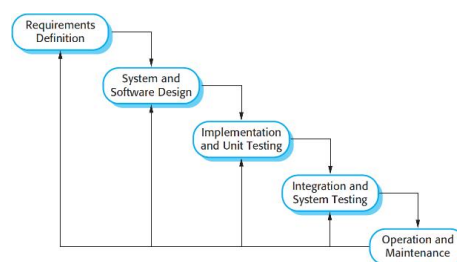
bisa menjadi lebih efektif dan efisien.

2. LANDASAN KEPUSTAKAAN

2.1 Metode Pengembangan

Penggunaan metode pengembangan sistem informasi disesuaikan karakteristik *stakeholder* terhadap proyek itu sendiri. Tiga metode yang umum digunakan diantaranya Agile, V-Model, dan Waterfall. Setiap metode tersebut menjadi efektif jika karakter dari *stakeholder* sesuai dengan metode yang digunakan. Pada metode Agile, metode ini merupakan teknik progresif yaitu mengerjakan proyek secara iteratif sampai dirasa cukup oleh *stakeholder*, sehingga cocok untuk yang tidak tahu akan kebutuhan terhadap sistem informasi itu sendiri. Pada metode V-Model, metode ini terdapat proses validasi dan verifikasi *stakeholder* di setiap fase pengerjaan untuk peninjauan sebelum pengembang lanjut ke fase selanjutnya, sehingga cocok untuk yang masih ragu akan kebutuhan terhadap sistem informasinya. Pada metode Waterfall, metode ini merupakan pengerjaan sekuensial, dimana setiap fase harus valid terlebih dahulu sebelum lanjut ke fase selanjutnya, sehingga cocok untuk yang sudah tahu akan kebutuhan terhadap sistem informasinya (Balaiji & Murugaiyan, 2012).

Dari setiap karakter metode pengembangan yang ada, metode pengembangan Waterfall merupakan metode yang paling cocok untuk penelitian ini, dikarenakan dari pihak PT. Debox Indonesia sudah mengetahui kebutuhan sistem informasi yang dibutuhkan.



Gambar 2.1 Metode Pengembangan Waterfall (Sommerville, 2011)

Tahap metode pengembangan ini dibagi menjadi beberapa fase diantaranya spesifikasi, implementasi, validasi, dan evolusi. Pada fase spesifikasi, kebutuhan dan tujuan yang ada pada sistem ditetapkan berdasarkan konsultasi kepada *stakeholder* sebagai spesifikasi sistem. Pada fase pengembangan, hasil dari spesifikasi diterapkan terhadap perangkat lunak. Pada fase validasi, perangkat lunak diuji setiap spesifikasi unitnya

sebagai verifikasi fungsional dan persyaratan perangkat lunaknya untuk validasi fungsional. Pada fase evolusi, sistem dijalankan *stakeholder* dan diperbaiki jika ada kesalahan (Sommerville, 2011).

2.2 Manajemen Proyek

Metode manajemen proyek tradisional mempunyai kekurangan terkait desain perangkat lunak dan pengembangannya jika diterapkan pada proyek berskala besar karena tidak empiris, tidak terduga, dan tidak bisa iteratif. Metode yang melakukan pendekatan pada karakteristik pemangku kepentingan adalah solusi untuk tujuan proyek lebih jelas sekalipun mereka tidak mengetahui secara persis perangkat lunak apa yang mereka butuhkan. Metode tersebut adalah Agile Project Management, dimana interaksi individu diutamakan daripada proses pengerjaan proyek, proses pengerjaan diutamakan daripada dokumentasi, dan pada pemangku kepentingan dilibatkan sehingga permintaan perubahan bisa langsung direspon daripada mengikuti rencana awal inisialisasi proyek hingga selesai (Azanza, Junior, & Argoud, 2017).

Scrum adalah metode pengerjaan tugas secara kolaborasi yang dimana setiap tugas disebut dengan *backlog*, dan tim yang terlibat terhadap *backlog* mengerjakan dalam beberapa minggu berdasarkan penentuan manajer proyek yang disebut dengan *sprint*.

2.3 Analisis Kebutuhan

Unified Model Language atau UML adalah standar pemodelan sistem berorientasi objek. UML mengandung mekanisme model statis dan dinamis. Atribut beserta operasi merupakan model statis, dan karakteristik waktu terkait bagaimana objek berkomunikasi dengan objek lainnya untuk menyelesaikan tugas merupakan model dinamis. Pada penerapan pengembangan perangkat lunak, seluruh kebutuhan fungsional dimodelkan dalam bentuk diagram Use Case beserta aktornya. Setelah pemodelan kebutuhan fungsional dan aktor, dilanjutkan dengan model statis yaitu kelas yang berisi atribut beserta metodenya dalam bentuk diagram Class beserta hubungan antar kelasnya. Setelah pemodelan statis, dilanjutkan dengan model dinamis yaitu interaksi antar objek dari *event* dipanggil hingga *output* yang dihasilkan dalam bentuk diagram Sequence (Zheng, Feng, & Zhao, 130).

Dari hasil analisis kebutuhan sistem pada penelitian ini, seluruh kebutuhan fungsional

dimodelkan dalam bentuk diagram Use Case. Lalu selanjutnya sistem dilakukan perancangan dengan pemodelan statis untuk kelasnya, dan pemodelan dinamis untuk alur kerja fungsional melewati kelas yang telah dibuat.

2.4 Implementasi Sistem

Bahasa yang sering pada saat ini untuk pengembangan web adalah JavaScript dengan dua *framework* yang populer berbasis bahasa ini yaitu AngularJS dan ReactJS. ReactJS adalah *framework* yang lebih unggul secara performa karena konsep virtual DOM yang bisa *rendering* dari sisi klien maupun dari sisi server sekaligus melakukan komunikasi, dibandingkan dengan AngularJS yang masih menggunakan konsep MVC (Model-View-Controller) yang hanya mengandalkan konsep *two-way binding*. Konsep *framework* AngularJS efektif untuk aplikasi web yang interaktif, tetapi secara performa jauh lebih cepat ReactJS (Kumar & Singh, 2015).

Teknologi *web service* terbaru pada saat ini adalah GraphQL, yang dimana berfungsi sebagai jembatan antara klien dengan server. GraphQL secara efektif mengambil dan mengembalikan data sesuai dengan apa yang diminta. Basis data yang digunakan adalah MongoDB, karena basis data ini berbasis cloud dengan URI yang sudah disediakan sehingga konektivitas dengan web service cukup mudah (Nugraha, 2016).

Dari hasil sumber yang didapatkan, sistem ini akan dikembangkan menggunakan ReactJS untuk antarmuka pengguna, mempertimbangkan performa *rendering* antarmuka menggunakan virtual DOM yang cepat. Lalu basis data yang digunakan untuk sistem ini adalah MongoDB dengan jembatan layanan web GraphQL yang berfungsi sebagai *middleware* antar klien dengan server, mengingat bahwa layanan web tersebut dapat mengambil sekaligus mengembalikan data secara efektif sesuai dengan apa yang diminta dan konektivitas MongoDB yang mudah karena URI yang sudah disediakan.

2.5 Pengujian Sistem

Perangkat lunak bisa dikatakan berkualitas bebas dari bug jika sudah melalui tahap dan proses pengujian, dimana pengujian yang dimaksud disini merupakan prosedur eksekusi program dengan tujuan untuk mencari kesalahan sebanyak mungkin agar program segera diperbaiki sebelum diberikan ke klien.

Pada proses verifikasi dan validasi, metode White Box dilakukan untuk menguji apakah

perangkat lunak dibangun dengan benar, dan metode Black Box dilakukan untuk menguji apakah perangkat lunak yang dibangun adalah benar. Metode White Box adalah proses verifikasi struktural terkait apakah kode pada program tidak ada kesalahan seperti pengetikan kode dan alur algoritma kode, dilakukan dari tahap implementasi kode. Metode Black Box adalah proses validasi fungsional sistem berdasarkan kebutuhan untuk memeriksa apakah sudah sesuai semua, dan menguji antarmuka dengan input valid dan tidak valid berdasarkan perspektif pengguna, dilakukan dari tahap implementasi antarmuka (Nidhra & Dondeti, 2012).

Mengingat perkembangan dari kebutuhan dari pengguna bukan hanya terkait fungsional saja, diperlukan juga pengujian bisa memastikan bahwa perangkat lunak layak dikatakan berkualitas dengan dilakukannya pengujian terhadap kebutuhan non-fungsional. Pengujian ini mencakup seluruh aspek yang tidak ada hubungannya dengan fungsionalitas perangkat lunak. Sehingga, pengujian ini hanya cukup berdasarkan yang dibutuhkan oleh pengguna saja (Manish & Kaur, 2016).

Dari hasil sumber yang didapatkan, sistem ini akan dilakukan pengujian dengan metode Black Box sebagai verifikasi fungsional untuk memastikan bahwa tujuan dari fungsional itu benar adanya, dan metode White Box sebagai validasi algoritma setiap fungsional untuk memastikan bahwa kode program benar adanya. Pengujian diluar aspek fungsionalitas sistem disesuaikan dengan kebutuhan pengguna agar menghasilkan sistem yang berkualitas sebagai pelengkap dari pengujian dasar.

3. METODOLOGI PENELITIAN

Metodologi penelitian disini merupakan tahapan dalam melaksanakan penelitian, dan dapat dilihat pada Gambar 3.1.

Langkah pertama adalah studi literatur, yang dimana informasi terkait sistem ini dari manajemen proyek itu sendiri hingga teknis pengembangan sistem dikumpulkan. Semua informasi yang dikumpulkan dari sumber resmi diantaranya jurnal, buku, artikel, dan penelitian terdahulu yang serupa.

Lalu dilanjutkan ke analisis kebutuhan, yang dimana garis besar sistem dijelaskan beserta diagramnya. Aktor diidentifikasi beserta kebutuhan fungsionalnya, kemudian dimodelkan dengan diagram Use Case. Basis

data diidentifikasi terkait entitas dan atribut apa saja yang perlukan.



Gambar 3.1 Metodologi Penelitian

Setelah analisis kebutuhan dirasa cukup, dari pihak *stakeholder* maupun pengguna, dilakukanlah perancangan sistem pada basis data, model statis, model dinamis, algoritma fungsional, dan antarmuka. Pada basis data, tabel dan kolom dibuat dengan isi entitas dan atributnya, untuk membantu peneliti meraba arsitektur basis data sebelum implementasi. Pada model statis dan dinamis, komponen sistem digambarkan dengan diagram Class, dan alur berjalannya fungsional melalui komponen digambarkan dengan diagram Sequence, untuk membantu peneliti meraba arsitektur sistem sebelum implementasi.

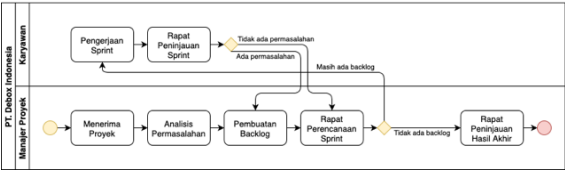
Pada antarmuka, pandangan secara garis besar pada tampilan didesain menggunakan *mockup* dengan jenis *low fidelity*, untuk membantu peneliti meraba antarmuka yang bagus sekaligus mudah digunakan sebelum implementasi. Pada algoritma fungsional, pseudocode metode struktural dibuat untuk membantu peneliti meraba kode sebelum tahap implementasi dan sekaligus berfungsi sebagai dasar pengujian.

4. ANALISIS KEBUTUHAN

4.1 Deskripsi Umum Sistem

Sistem ini dikembangkan dengan basis web yang bertujuan untuk memudahkan proses pada manajemen proyek yang bersifat kolaborasi kerja, dan secara spesifik menggunakan metode Agile dengan kerangka kerja Scrum.

4.2 Proses Bisnis



Gambar 4.1 BPMN Manajemen Proyek

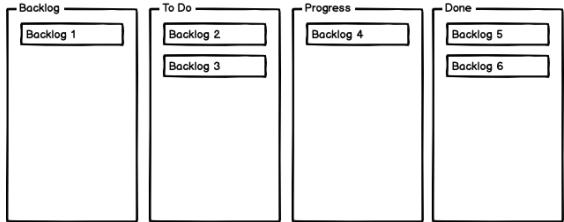
Tahap pertama pada pengerjaan proyek setelah proyek diterima dari pihak klien atau *stakeholder* adalah analisis permasalahan, yaitu mencatat garis besar proyek dengan formulir tinjauan yang berisi lima informasi utama untuk menentukan apa yang harus bisa dilakukan sistem, dan dijelaskan pada tabel dibawah ini.

Tabel 4.1 Formulir Tinjauan Proyek

Latar Belakang	Latar belakang atau permasalahan yang dihadapi oleh klien saat menjalankan kegiatannya hingga klien membutuhkan sistem untuk mempermudah kegiatan tersebut.
Tujuan	Tujuan yang ingin dicapai klien pada sistem terhadap kegiatan tersebut.
Objektivitas	Abstrak dari sistem yaitu apa yang harus bisa dilakukan oleh sistem informasi dalam rangka memudahkan permasalahan klien
Kriteria Sukses	Parameter untuk penentu apakah sistem yang akan dibuat nantinya sesuai dengan yang ekspetasi dan keinginan klien.
Hambatan	Asumsi, resiko, atau hambatan yang akan atau diperkirakan akan ada dalam pengerjaan sistem informasi. Poin formulir ini bisa terkait dari proses bisnis kegiatan klien atau bahkan hambatan dalam pengerjaan proyek ini.

Setelah analisis permasalahan, scrumming proyek dilakukan. Kebutuhan sistem didaftarkan dalam bentuk backlog berdasarkan modul pada formulir objektivitas, dan setiap modul pada objektivitas ditentukan waktu pengerjaannya menggunakan *ganttt chart*. Lalu pada Scrum, dilakukan rapat perencanaan sprint dalam rangka menentukan backlog yang dikerjakan, berapa minggu pengerjaannya, dan siapa saja yang akan terlibat. Setelah pengerjaan sprint selesai, maka akan dilakukan rapat lagi untuk membahas apa yang sudah dan apa yang belum, hal yang belum ini adalah permasalahan yang harus dimasukkan ke dalam backlog untuk dikerjakan pada sprint selanjutnya. Proses ini bersifat iteratif hingga semua backlog selesai dikerjakan. Setelah sistem informasi sudah jadi dan sudah teruji, maka akan

dilakukan peninjauan bersama dari pihak klien dan pengembang untuk menentukan kebenaran dari tujuan sistem informasi tersebut, dan hasil akhir diberikan kepada klien.



Gambar 4.2 Proses Pengerjaan Sprint

Pada pengerjaan sprint, backlog dibagi kategorinya menjadi 4 papan. To Do merupakan backlog yang akan dikerjakan Progress adalah backlog yang sedang dikerjakan. Done adalah backlog yang sudah selesai dikerjakan.

4.3 Identifikasi Aktor

Aktor perlu diidentifikasi untuk penjelasan secara detail terkait apa yang bisa dilakukan terhadap sistem. Pada sistem ini, ada tiga aktor diantaranya User, Leader, dan Employee. Setiap aktor tersebut dijelaskan pada tabel dibawah ini.

Tabel 4.2 Identifikasi Aktor Sistem

User	User adalah aktor yang belum login pada sistem, hanya bisa melakukan signin dan signup sebagai aktor Leader.
Leader	Leader adalah pimpinan organisasi yang mempunyai hak akses penuh pada data klien, divisi, karyawan, proyek dan manajemen proyek.
Employee	Employee adalah karyawan organisasi. Untuk aktor ini hanya bisa membaca data saja. Tetapi, aktor ini mempunyai fitur notifikasi yang tidak dimiliki aktor Leader, yang berfungsi untuk pemberitahuan undangan kolaborasi proyek.

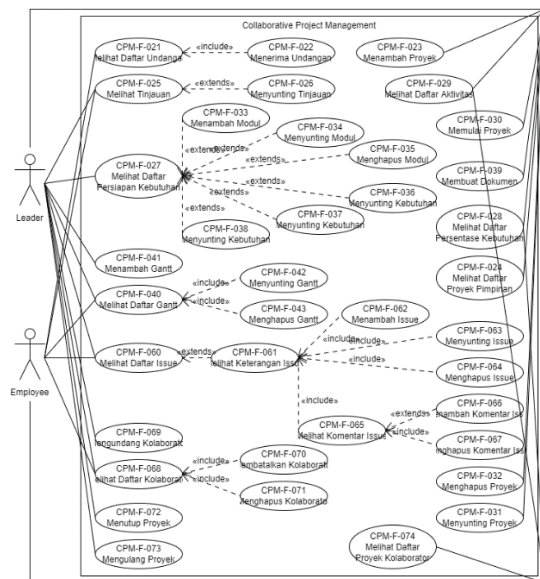
4.4 Analisis Kebutuhan

Kebutuhan yang harus ada pada sistem dan kebutuhan yang tidak terkait secara teknis harus didokumentasikan. Kebutuhan sistem yang harus ada disebut dengan kebutuhan fungsional, sedangkan kebutuhan yang tidak terkait secara teknis disebut dengan kebutuhan non fungsional. Pada sistem ini, terdapat 4 modul dengan total kebutuhan fungsional sebanyak 80, dan terdapat 3 kebutuhan non fungsional.

4.5 Pemodelan Fungsional

Perilaku aktor kepada sistem digambarkan

mengenai fungsional apa saja yang bisa diakses oleh setiap aktor berdasarkan ruang lingkup modul menggunakan diagram Use Case.



Gambar 4.3 Use Case Modul Project Management

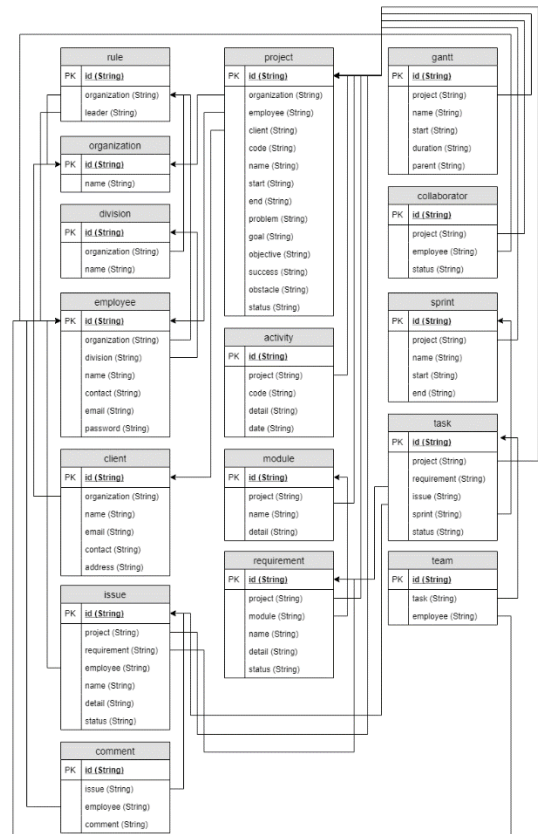
4.6 Analisis Data

Data beserta struktur basis data dianalisis berdasarkan kebutuhan fungsional. Pada sistem ini, terdapat 16 entitas untuk basis data.

5. PERANCANGAN SISTEM

5.1 Basis Data

Berdasarkan hasil analisis data pada sistem, dibuatlah diagram basis data yang digambarkan dibawah ini.

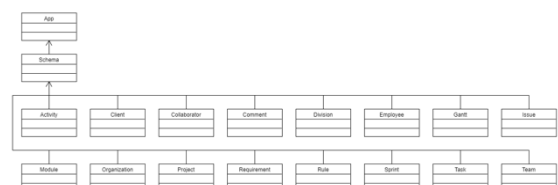


Gambar 5.1 Diagram Perancangan Basis Data

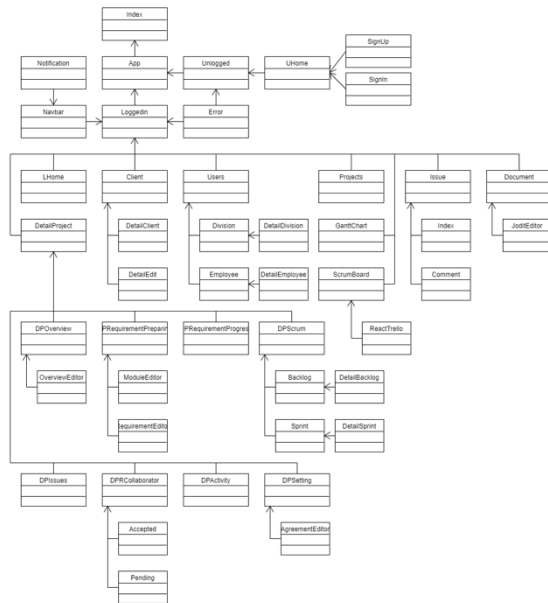
5.2 Pemodelan Statis

Pemodelan statis pada perancangan sistem disini menggunakan diagram Class. Diagram ini merupakan pemodelan kelas yang berisi atribut beserta metodenya, dan dideskripsikan dalam bentuk tabel beserta hubungan antar kelasnya.

Pemodelan statis pada penelitian ini mencakup dua ruang lingkup, yaitu sistem dan layanan web. Ruang lingkup sistem adalah rancangan komponen terhadap ReactJS sebagai antarmuka pengguna, dan ruang lingkup layanan web adalah rancangan kelas terhadap GraphQL sebagai jembatan antar klien dengan basis data.



Gambar 5.2 Diagram Kelas Layanan Web



Gambar 5.3 Diagram Kelas Sistem

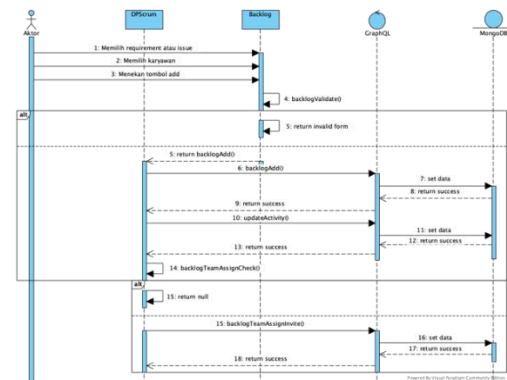
5.3 Pemodelan Dinamis

Pemodelan dinamis pada perancangan sistem disini menggunakan diagram Sequence. Diagram ini merupakan pemodelan fungsional terhadap kelas yang terlibat, dimana alur kerja fungsional tersebut digambarkan melalui kelas apa saja dari kelas yang sudah dimodelkan pada pemodelan statis, dan alur kerjanya berdasarkan skenario diagram Use Case.

Tabel 5.1 Skenario Use Case Menambah Backlog

Objektif	Sistem menambahkan data backlog pada proyek
Aktor	Leader
Pre Condition	Aktor sudah berada pada halaman DetailProject tab Scrum
Main Flow	<ol style="list-style-type: none"> 1. Aktor memilih requirement atau issue 2. Aktor memilih karyawan 3. Aktor menekan tombol add 4. Sistem memeriksa pengisian formulir aktor 5. Sistem menambahkan data backlog dan menambahkan aktivitas proyek ke basis data 6. Sistem menambahkan tim ke basis data sekaligus memeriksa status kolaborator karyawan 7. Sistem menampilkan pesan berhasil
Alternative Flow	<ol style="list-style-type: none"> 4. Jika pengisian formulir tidak sesuai, maka sistem menampilkan pemberitahuan 6. Jika karyawan belum terdaftar sebagai kolaborator, maka sistem mengundang karyawan

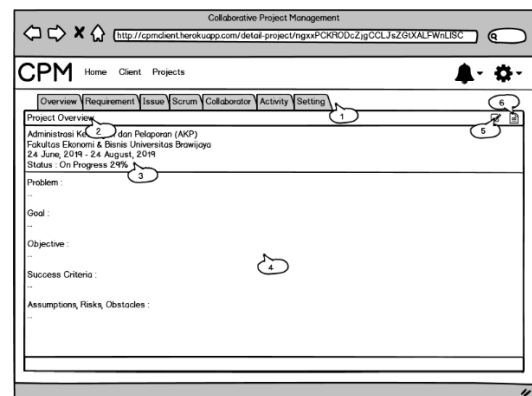
sebagai kolaborator
Post Condition Aktor bisa melihat data backlog bertambah



Gambar 5.4 Diagram Sequence Menambah Backlog

5.4 Antarmuka Pengguna

Perancangan antarmuka pengguna adalah gambaran wireframe antarmuka sistem sebagai dasar implementasi antarmuka sistem.



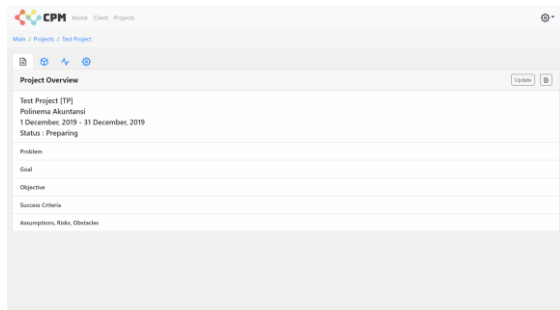
Gambar 5.5 Perancangan Halaman Detail Proyek

Tabel 5.2 Penjelasan Halaman Detail Proyek

Nama	Keterangan
Tab Menu	Tempat untuk berpindah jenis detail proyek dengan tab.
Tab Aktif	Nama tab yang sedang dibuka.
Perjanjian Proyek	Informasi data perjanjian proyek berisi kode, nama, klien, tanggal mulai, tanggal berakhir, status, dan persentase pengerjaan jika ada.
Tinjauan Proyek	Informasi data tinjauan proyek terkait Analisa kebutuhan berisi latar belakang, tujuan, objektifitas, kriteria sukses, dan hambatan.
Tombol Sunting Tinjauan Proyek	Tombol untuk membuka modal sunting tinjauan proyek.
Tombol Pembuatan Dokument	Tombol untuk membuka halaman text editor pembuatan dokumen proyek.

6. IMPLEMENTASI SISTEM

Implementasi sistem dilakukan melalui 4 tahap, yaitu basis data, layanan web, antarmuka, dan kode program.



Gambar 6.1 Antarmuka Halaman Detail Proyek

7. PENGUJIAN SISTEM

Setelah sistem dikembangkan maka harus dilakukan pengujian dan analisis terkait verifikasi dan validasi sistem, untuk menemukan kesalahan sebelum digunakan kepada pengguna.

7.1 Pengujian Fungsional

7.1.1 Verifikasi Fungsional

Diagram *control flow graph* dibuat terhadap algoritma fungsional, kemudian dihitung jumlah *cyclomatic complexity*, dan dibuat *independent path* pengujian sejumlah hasil perhitungan, dan dibuat tabel hasil pengujian.

a. Pseudocode

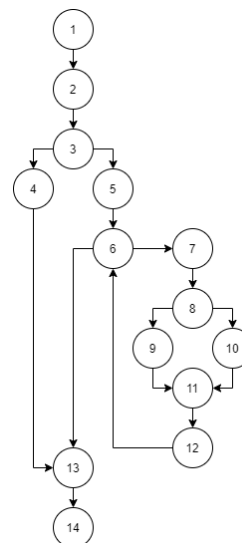
Tabel 7.1 Pseudocode Menambah Backlog

```

Start (1)
Initialize array collaborator (2)
Initialize array backlog (2)
Initialize array team (2)
Click add button (2)
If backlog value == null (3)
Return invalid form (4)
Else (5)
Set data to task (5)
Set data to activity (5)
For each team as i (6)
Set team[i] value to team (7)
If team[i] value isn't collaborator (8)
Set team[i] value to collaborator (9)
Else do nothing (10)
End if (11)
End for (12)
End if (13)
End (14)

```

b. Control Flow Graph



Gambar 7.1 CFG Menambah Backlog

c. Cyclomatic Complexity

$$V(G) = 16E - 14N + 2 = 4$$

$$V(G) = 4R = 4$$

$$V(G) = 3P + 1 = 4$$

d. Independent Path

- 1-2-3-4-13-14
- 1-2-3-5-6-13-14
- 1-2-3-5-6-7-8-9-11-12-6-13-14
- 1-2-3-5-6-7-8-10-11-12-6-13-14

e. Test Result

Tabel 7.2 Hasil Uji Verifikasi Jalur 4

Data Uji	collaborator:[inputBacklog:[{id:001}] inputTeam:[{id:C01},{id:C02}]
Ekspetasi	Sistem berhasil mendaftarkan backlog dan mengundang karyawan pada tim yang belum menjadi kolaborator
Hasil	Sistem berhasil mendaftarkan backlog dan mengundang karyawan pada tim yang belum menjadi kolaborator
Status	Pass

7.1.2 Validasi Fungsional

Sistem diuji melalui antarmuka pengguna tanpa memperhatikan bagaimana algoritma kode program berjalan untuk melihat kebenaran dari tujuan fungsional tersebut menggunakan jenis pengujian *equivalence partitioning*. Jenis ini menggunakan skenario pengujian yang dipartisi

menjadi kelas valid dan tidak valid, kemudian diuji salah satu skenario pada setiap kelas untuk mewakili semua skenario pengujian yang ada pada kelas tersebut.

a. Scenario

1. Menambahkan backlog dengan formulir kosong
2. Menambahkan backlog dengan formulir terisi tanpa tim
3. Menambahkan backlog dengan formulir terisi dan tim yang bukan kolaborator
4. Menambahkan backlog dengan formulir terisi dan tim yang kolaborator

b. Test Result

Tabel 7.3 Hasil Uji Validasi Jalur 4

Data Uji	Backlog : Reqs Team : Faesal
Ekspetasi	Data backlog bertambah dan aktivitas proyek bertambah
Hasil	Data backlog bertambah dan aktivitas proyek bertambah
Status	Pass

7.2 Pengujian Non-Fungsional

Kebutuhan yang tidak berhubungan dengan kebutuhan sistem diuji juga agar menghasilkan produk berkualitas sebelum diberikan kepada pengguna.

Tabel 7.4 Hasil Uji Non-Fungsional Compatibility

Skenario	Sistem dibuka pada peramban Chrome, Firefox, dan Opera dengan versi terakhir tahun 2019, lalu menjalankan fungsional login, melihat dashboard, dan logout.
Ekspetasi	Sistem bisa menjalankan fungsional login, melihat dashboard, dan logout pada ketiga peramban tersebut.
Hasil	Sistem bisa menjalankan fungsional login, melihat dashboard, dan logout pada ketiga peramban tersebut.
Status	Pass

8. KESIMPULAN DAN SARAN

8.1 Kesimpulan

Pada hasil analisis pada sistem didapatkan pengertian metode Agile dengan kerangka kerja Scrum, dan bagaimana metode beserta kerangka kerja tersebut berjalan. Jumlah aktor yang menggunakan sistem ada 3. Jumlah kebutuhan fungsional ada 80, kebutuhan non-fungsional ada 3, dan entitas basis data ada 16. Kebutuhan

fungsional dimodelkan dalam bentuk diagram Use Case untuk mempermudah secara visual terkait siapa yang bisa melakukan fungsional apa.

Pada hasil perancangan sistem didapatkan model basis data dalam bentuk kelas beserta garis relasinya. Kerangka sistem dimodelkan dalam bentuk pemodelan statis menggunakan diagram Class. Pemodelan statis pada web service menyesuaikan hasil analisis entitas basis data. Alur berjalannya fungsional dimodelkan dalam bentuk pemodelan dinamis menggunakan diagram Sequence berdasarkan skenario diagram Use Case. Algoritma kode fungsional dirancang menggunakan pseudocode sebagai dasar implementasi kode dan dasar pengujian verifikasi fungsional. Antarmuka pengguna dirancang dengan *mock up low fidelity* sebagai dasar implementasi antarmuka pengguna.

Pada hasil implementasi sistem didapatkan antarmuka dengan dasar ReactJS. Setiap fungsional yang membutuhkan koneksi ke basis data dijembatani oleh layanan web GraphQL. Basis data yang digunakan berjenis cloud dari penyedia jasa MongoDB. Baik dari *client* maupun *web service*, semuanya dijalankan menggunakan NodeJS berbasis JavaScript.

Pada hasil pengujian sistem didapatkan 9 verifikasi fungsional metode *white box* dengan hasil benar semua, dan 110 validasi pengujian metode *black box* jenis *equivalence partitioning* dengan hasil benar semua. Pada pengujian non-fungsional, 3 kebutuhan diluar aspek fungsional ini berjalan dengan baik tanpa ada kesalahan.

8.2 Saran

1. Halaman pada sistem disederhanakan hanya berdasarkan empat modul saja, diantaranya organisasi, manajemen klien, manajemen proyek, dan *scrumming* proyek.
2. Data pada *gant chart* dijadikan satu terhadap data kebutuhan dan permasalahan kebutuhan proyek.
3. Penetapan tanggal mulai *sprint* pada sistem dibuat fleksibel tanpa harus menunggu tanggal mulai yang sebenarnya.
4. Proses verifikasi *rule* pengguna terkait hak akses fungsional masih bersifat *client-side*, jika dilakukan secara *server-side* dari sisi keamanan akan lebih bagus.

9. DAFTAR PUSTAKA

Balaiji, S., & Murugaiyan, D. M. (2012).

- Waterfall vs V-Model vs Agile: A Comparative Study on SDLC. *International Journal of Information Technology and Business Management*, 26-29.
- Sommerville, I. (2011). *Software Engineering*. Boston: Pearson Education.
- Zheng, J., Feng, Y., & Zhao, Y. (130). *A Unified Modeling Language-Based Design and Application*. Cybernetics and Information Technologies: 2014.
- Nugraha, H. S. (2016). *Analisis Performansi Web Service API GraphQL Pada Domain M-Commerce Menggunakan NodeJS Dan MongoDB*. Bandung: Universitas Komputer Indonesia.
- Kumar, A., & Singh, R. K. (2015). Comparative Analysis of AngularJS and ReactJS. *International Journal of Latest Trends in Engineering and Technology*, 225-227.
- Nidhra, S., & Dondeti, J. (2012). Black Box and White Box Testing Techniques – A Literature Review. *International Journal of Embedded Systems and Applications (IJESA)*, 29-38.
- Manish, & Kaur, I. (2016). An Overview of Non-Functional Testing. *Biz and Bytes*, 72-76.
- Azanha, A., Junior, J. B., & Argoud, A. R. (2017). Agile project management with Scrum: A case study of a Brazilian pharmaceutical company IT project . *International Journal of Managing Projects in Business*.