



Penerapan Scrum dan Algoritma COCOMO Pada Aplikasi Manajemen Proyek Perangkat Lunak

Wildan Suharso
Universitas
Muhammadiyah Malang
wsuharso@umm.ac.id

Bayu Indra Wicaksono
PT. Kusuma Satria Dinasasri
Wisatajaya
bayuindrawicaksono69@gmail.com

Gita Indah Marthasari
Universitas Muhammadiyah
Malang
gita.voyager@gmail.com

Abstrak

Karakteristik manajemen proyek perangkat lunak adalah biaya yang mahal, waktu terbatas dan minimnya transparansi pada semua pemangku kepentingan, Perencanaan yang baik perlu diimbangi dengan pelaksanaan yang sesuai karena banyak dari proyek yang gagal karena kurangnya perencanaan. Metode Scrum dapat membantu semua pemangku kepentingan dalam memahami proyek sehingga hasil dari proyek sesuai dengan perencanaan. Pada penelitian ini dilakukan penerapan metode scrum pada aplikasi manajemen proyek perangkat lunak untuk membantu tim dalam memahami proyek. Pada penelitian ini scrum tidak hanya sebagai model untuk pengembangan perangkat lunak tetapi diimplementasikan pada aplikasi sehingga manajer proyek, master scrum dan tim pengembang dapat secara mudah mengontrol task. Algoritma CoCoMo digunakan untuk mengestimasi biaya proyek, Pengujian menggunakan pengujian fungsional dan skenario yang menjelaskan permasalahan. Hasil yang diperoleh adalah aplikasi dapat membantu manajer proyek, master scrum, dan tim pengembang dalam hal manajemen proyek perangkat lunak. Hasil pengujian estimasi biaya yang dilakukan menunjukkan nilai estimasi tidak melebihi 7% dari nilai real dan estimasi waktu real lebih efisien 40% dari waktu CoCoMo.

Kata Kunci : Scrum, CoCoMo, Manajemen Proyek Perangkat Lunak, Sistem Informasi, Codeigniter

1. Pendahuluan

Manajemen proyek perangkat lunak dikatakan berhasil jika terdapat kesuaian antara perencanaan dan pengembangan perangkat lunak, persentase keberhasilan proyek semakin meningkat jika perencanaan telah dilakukan secara maksimal dan sesuai. Perencanaan merupakan aktivitas utama yang diikuti oleh pelaksanaan, kontrol, penyelesaian, dan pemeliharaan. Sebagian pengelola proyek beranggapan bahwa hal paling penting dalam proyek adalah pembuatan aplikasi yang menyebabkan pengelola proyek mengutamakan pengerjaan program sehingga perencanaan tidak dilakukan secara maksimal (Munir, 2015).

Secara umum manajemen proyek merupakan pengaplikasian pengetahuan, keterampilan, peralatan, dan teknik untuk kegiatan proyek dalam memenuhi persyaratan proyek (Duncan, 1996). Semua faktor harus dikelola dengan baik karena hal yang dianggap tidak penting dapat menjadi faktor kesuksesan proyek. Proyek perangkat lunak meliputi ruang lingkup, waktu, kualitas, biaya proyek, sumberdaya, komunikasi, konfigurasi dan resiko. Kendala utama sering ditemukan pada ruang lingkup, waktu, kualitas dan biaya (Munir, 2015). Sehingga faktor paling penting dalam kesuksesan proyek adalah perencanaan yang baik.

Komunikasi antar anggota proyek dilakukan secara langsung sehingga menyebabkan komunikasi kurang efektif dan manajemen kerja antar anggota yang buruk. Tugas manajer proyek adalah membuat tim berkomunikasi secara efektif, mengatur sumber daya yang terbatas, pencapaian milestones yang sesuai

dengan rencana, pengerjaan proyek berdasarkan rencana sesuai dengan manajemen tugas (Thamhain, 1986). Pembagian modul secara merata kepada programmer tidak menjamin kesesuaian hasil milestones dengan detail rencana karena setiap programmer memiliki karakteristik yang berbeda dari segi tingkat penyelesaian modul ataupun kebiasaan dalam mengerjakan modul.

Penyelesaian proyek yang terlambat dapat diakibatkan oleh kesalahan dalam faktor biaya dan waktu, yang disebabkan oleh kurang efektifnya manajer proyek dalam menentukan biaya dan waktu, kurangnya perencanaan, dan kurangnya pengendalian (Syamsiah, 2014). Permasalahan-permasalahan yang dijelaskan pada manajemen proyek perangkat lunak dapat terselesaikan jika terdapat aplikasi yang dapat membantu dalam pengerjaan modul, pengelolaan proyek, penentuan tugas, dan aktifitas lain yang dapat meningkatkan resiko kegagalan proyek perangkat lunak. Pendekatan yang mungkin dilakukan salah satunya adalah mengimplementasikan Scrum, dimana pada scrum dapat diketahui detail seluruh proyek perangkat lunak yang meliputi pembagian manajer proyek, master scrum, tim pengembang, pembagian proyek, penentuan waktu pengerjaan sebuah modul, hingga penentuan rapat oleh master scrum.

Permasalahan mengenai biaya dan waktu tidak bisa terselesaikan sepenuhnya namun pengimplementasian CoCoMo (Constructive Cost Model) diharapkan dapat memudahkan manajer proyek dalam estimasi biaya. CoCoMo menggunakan 2 parameter, yaitu Relative Effort yaitu parameter yang digunakan untuk mengukur biaya pembuatan perangkat lunak dan Time to Develop yaitu parameter untuk mengukur perkiraan waktu yang diperlukan untuk membangun suatu perangkat lunak. (Putra, 2011).

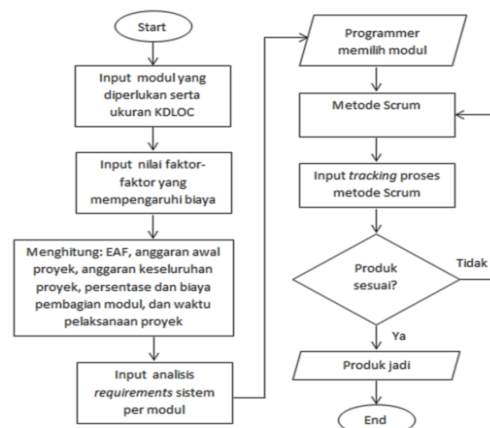
Implementasi yang dilakukan pada penelitian ini tidak hanya menggunakan scrum tetapi membuat aplikasi dimana seluruh aktifitas scrum dapat dilakukan, meliputi product backlog, sprint planning, sprint, dan sprint review. Bahkan manajer proyek hingga master scrum dapat berinteraksi langsung di dalam program. Kontribusi lain pada penelitian ini adalah digunakannya CoCoMo dalam penentuan biaya dan waktu pada aplikasi sehingga dapat memudahkan manajer proyek dalam estimasi biaya dan waktu.

Pada penelitian ini digunakan data yang dimiliki oleh salah satu *software house* yang telah mengimplementasikan scrum pada setiap proyek yang dikerjakan yang berlokasi di Kota Malang, pengujian juga dilakukan pada *software house* yang digunakan sebagai studi kasus. Pada penelitian ini juga diimplementasikan algoritma CoCoMo untuk menentukan estimasi biaya dan waktu. Pengujian estimasi biaya dan waktu dilakukan untuk menentukan sejauh mana efisiensi sistem manajemen perangkat lunak yang dibangun.

2. Metode Penelitian

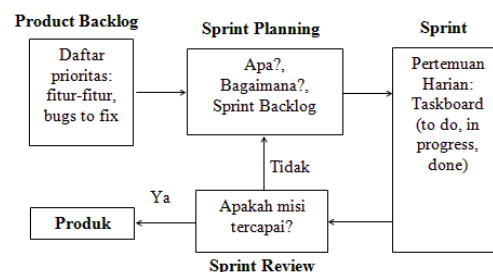
2.1. Scrum

Penerapan scrum pada aplikasi merupakan hal baru sehingga alur pada aplikasi harus sesuai dengan alur scrum secara manajemen. Secara umum metode penelitian meliputi analisis, desain, implementasi dan pengujian, tetapi secara khusus pada setiap detail tahapan akan menjelaskan penerapan metode scrum. Pada tahap analisis dilakukan observasi, pengumpulan data, dan wawancara pada salah satu *software house* yang berlokasi di Kota Malang. Pada hasil analisis diketahui bahwa perusahaan telah menggunakan kerangka scrum pada pengerjaan proyek namun tidak menggunakan aplikasi dalam menerapkan scrum sehingga setiap proses dilakukan secara langsung. Tahapan desain ditunjukkan pada Gambar 1.



Gambar 1 Tahapan Desain

Pada Gambar 1 ditunjukkan mengenai alur sistem yang menerapkan metode scrum yang dimulai dari menentukan modul dengan detail setiap instrumen hingga proyek perangkat lunak selesai. Sebelum masuk pada metode scrum diperlukan persiapan untuk semua modul yang dibutuhkan pada proyek perangkat lunak yang meliputi modul, biaya, dan waktu. Manajer proyek bertanggung jawab pada keseluruhan proses, dan programmer bertanggung jawab terhadap modul yang telah dipilih secara pribadi. Setelah programmer memilih modul yang akan dikerjakan maka dilanjutkan dengan metode scrum yang ditunjukkan pada Gambar 2.



Gambar 2. Scrum

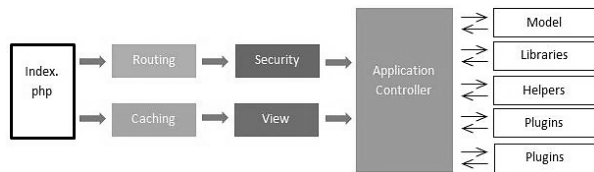
Pada Gambar 2 ditunjukkan metode scrum yang terdiri dari product backlog, sprint planning, sprint, dan

sprint review. Product backlog merupakan daftar kerja yang akan dikerjakan berdasarkan prioritas, untuk mencapai tujuan diperlukan sprint planning yaitu menentukan cara untuk menyelesaikan daftar kerja. Pada penelitian ini daftar kerja berupa modul yang harus dikerjakan oleh programmer dan modul sesuai dengan analisis *requirement* yang telah disiapkan oleh manajer proyek. Sprint digunakan untuk melakukan evaluasi semua kegiatan, sprint membahas mengenai permasalahan yang dihadapi, daftar kerja yang terselesaikan, daftar kerja yang akan dikerjakan, ataupun daftar kerja yang masih dalam pengerjaan. Sprint review dilakukan untuk menilai hasil dari sprint yang dilakukan pada rapat atau diskusi yang dilaksanakan setiap hari yang dipimpin oleh Scrum Master dan siklus sprint akan terus dilakukan hingga aplikasi sesuai. Pada penelitian semua tim scrum akan saling berkomunikasi menggunakan website, sehingga penentuan prioritas, pemilihan programmer oleh manajer proyek, pemilihan modul oleh programmer, evaluasi dan tracking dapat dilakukan melalui aplikasi.

Dan yang menentukan apakah proyek sesuai dengan *requirements* adalah *product owner*. Sistem diatas bertujuan untuk membantu dalam manajemen proyek pembuatan *website* dimana akan mengetahui estimasi waktu dan biaya, dan setiap *stakeholder* mengerti akan tugasnya sehingga menghasilkan produk yang berkualitas. Pada alur tersebut juga dijelaskan bahwa *programmer* dapat melakukan pemilihan modul dan dilakukan *tracking*.

2.2. Codeigniter

Pada tahap inplementasi dilakukan dengan menggunakan PHP & Mysql, semua aktifitas pada tahapan ini menggunakan pendekatan *prototype*, dan memanfaatkan framework *Codeigniter*.



Gambar 3.CodeigniterFramework

Pada Gambar 3 ditunjukkan alur aplikasi dari *Codeigniter Framework* yang menjelaskan mengenai penggunaan *modul,libraries, helpers, plugins* pada *index* sistem.

2.3. Pengujian

Pengujian dilakukan melalui 2 cara yaitu menggunakan pengujian secara fungsional dan pengujian berdasarkan skenario. Pengujian secara fungsional dilakukan untuk mengetahui kesesuaian

program dengan analisis yang telah dilakukan sedangkan pengujian menggunakan skenario untuk mengetahui sejauh mana kesesuaian manajemen program dengan scrum. Skenario meliputi beberapa hal sebagai berikut :

- a. Menguji program jika task yang dikerjakan oleh tim pengembang terdapat masalah , atau proyek tidak dapat diselesaikan.
- b. Menguji program jika terdapat task yang belum mulai dikerjakan pada hari atau tanggal yang ditentukan.
- c. Menguji langkah pada proses scrum jika terdapat task yang berstatus complete namun ternyata ada perubahan dapat karena kesalahan program.

Instrumen pengujian skenario secara singkat dapat dijelaskan pada Tabel 1.

Tabel 1 Instrumen Pengujian

No	Prosedur	Keluaran yang diharapkan
1	Bagaimana jika <i>task</i> dalam tahap “ <i>in progress</i> ” tidak dapat diselesaikan?	<i>Task</i> akan berganti status menjadi “ <i>need attention</i> ”.
2	Bagaimana jika <i>task</i> dalam tahap “ <i>to do</i> ” tidak bisa dikerjakan pada iterasi hari atau tanggal saat itu?	<i>Task</i> akan otomatis terpindah saat terdapat iterasi baru.
3	Bagaimana jika <i>task</i> sudah <i>complete</i> namun ternyata terdapat bugs atau harus dilakukan perubahan?	<i>Task</i> akan berganti status “ <i>in progress</i> ” kembali

Pada Tabel 1 dijelaskan mengenai instrumen dan keluaran yang diharapkan dari program, sehingga jika dapat memenuhi instrumen pengujian maka dapat dikatakan program telah sesuai dengan permasalahan pada penelitian.

2.4. CoCoMo

CoCoMo menganggarkan biaya dengan satuan personal month (PM), CoCoMo memiliki tahapan antara lain menentukan ukuran kegiatan berdasarkan KDLOC, menentukan faktor faktor yang mempengaruhi biaya, menghitung effort adjustment factor (EAF), anggaran awal dan keseluruhan proyek, waktu keseluruhan dan permodul.

KDLOC adalah Kilo Delivered Line Of Code atau ukuran satuan modul yang menilai semua kegiatan berdasarkan modul, faktor yang mempengaruhi biaya digunakan nilai *multiplying factors* dengan range nilai 0,9 hingga 1,4. Untuk menentukan nilai dari anggaran awal sebuah proyek menggunakan rumus 1.

$$E_i = a * (KDLOC)^b \dots\dots\dots (1)$$

Pada rumus 1 dijelaskan bahwa E_i merupakan anggaran awal proyek, a merupakan nilai optimis, dan

b merupakan nilai pesimis. Jenis proyek berdasarkan organic, semidetached, dan embedded. Untuk menentukan nilai keseluruhan proyek menggunakan rumus 2.

$$E = EAF * Ei \dots\dots\dots(2)$$

Dimana E adalah nilai keseluruhan proyek, EAF adalah *effort adjustment factor* yang merupakan faktor hasil perhitungan dari sub kategori, dan Ei yang merupakan anggaran awal proyek. Untuk menentukan presentase dan biaya setiap modul menggunakan rumus 3.

$$Presentase = \frac{Nilai In + \frac{(Nilai Med - Nilai In)}{(KDLOC Med - KDLOC In)} \times E}{KDLOC In} \dots\dots(3)$$

Pada rumus 3 semua terkait dengan rumus 3 dimana nilai E merupakan nilai keseluruhan proyek, nilai medium, nilai intermediate, dan KDLOC. Untuk menentukan waktu keseluruhan pelaksanaan proyek menggunakan rumus 4.

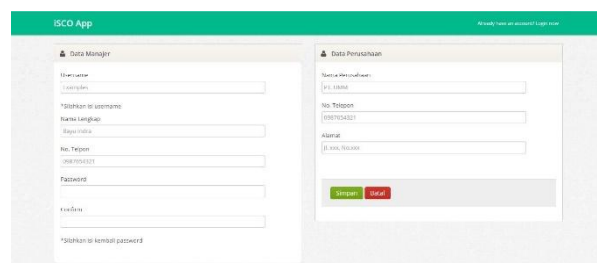
$$D = 2,5 * E^{0,38} \dots\dots\dots(4)$$

Dimana D merupakan nilai durasi, nilai 2,5 dan 0,38 diambil dari variabel proyek dan jenis proyek organic. Sedangkan untuk menentukan nilai waktu setiap modul digunakan perkalian persentase setiap modul dengan nilai durasi.

3. Hasil dan Pembahasan

3.1. Hasil

Pada bab ini dijelaskan mengenai hasil tampilan yang dihasilkan beserta pembahasan yang menjelaskan mengenai hasil penelitian. Pada Gambar 4 ditunjukkan halaman utama dari aplikasi yang dibangun, yaitu halaman daftar.



Gambar 4. Halaman Daftar

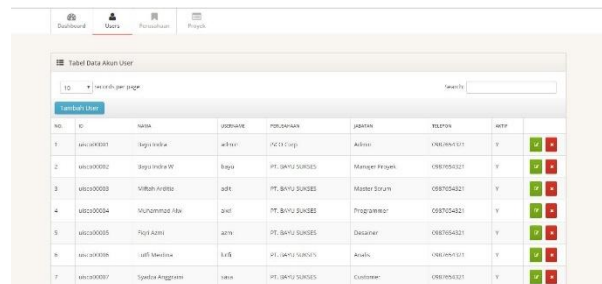
Halaman daftar yang ditunjukkan pada Gambar 4 merupakan halaman daftar bagi manajer proyek yang nantinya sebagai pemimpin proyek. Pada halaman tersebut terdapat dua form, yaitu form pengisian data manajer proyek dan form pengisian data perusahaan. Setelah halaman daftar maka yang paling penting untuk ditunjukkan adalah halaman *dashboard*, yang merupakan halaman pertama yang akan dikunjungi jika semua user *login* yang ditunjukkan pada Gambar 5.



Gambar 5. Dashboard

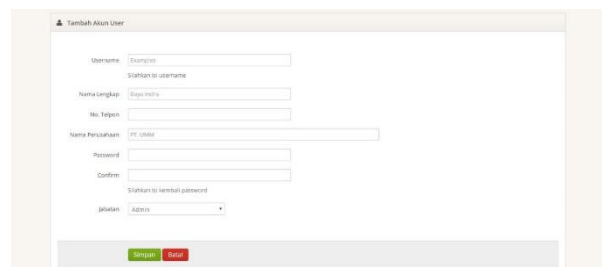
Pada Gambar 5 ditunjukkan halaman dashboard untuk setiap user, pada penelitian ini yang digunakan adalah user dari peneliti sebagai contoh. Pada gambar tersebut akan terlihat grafik yang menjelaskan pertumbuhan data user dan data proyek setiap bulan.

Hasil penelitian yang perlu ditunjukkan selanjutnya adalah halaman administrasi *user* yang terdiri dari tampil data *user*, tambah data *user*, dan edit data *user* yang ditunjukkan pada Gambar 6, Gambar 7, dan Gambar 8.



Gambar 6. Tampil Data User

Pada Gambar 6 ditunjukkan halaman tampil data user yang berisi mengenai user role, termasuk didalamnya id, nama, jabatan, perusahaan, nomer tlp, dan tombol untuk edit dan hapus.



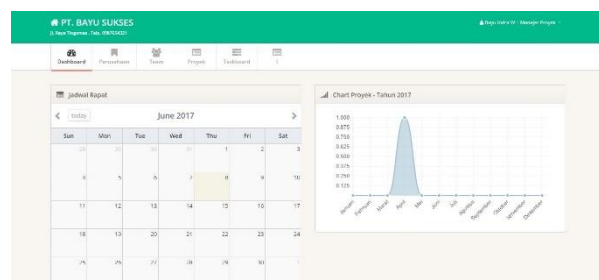
Gambar 7. Tambah User

Pada Gambar 7 ditunjukkan halaman untuk tambah user yang berisi isian mengenai nama, nomer tlp, jabatan, perusahaan, *username* dan *password*.

Gambar 8. Edit Data User

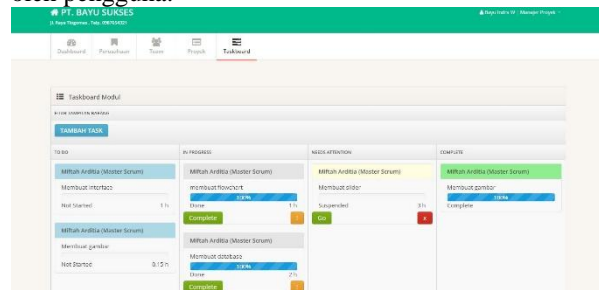
Pada Gambar 8 ditunjukkan halaman edit data user yang merupakan halaman untuk mengganti / mengubah data user yang telah terdaftar, secara keseluruhan isi dari edit data user adalah sama dengan tambah user dengan perbedaan data yang telah terisi sebelumnya.

Semua halaman yang ditunjukkan pada Gambar 6, Gambar 7, dan Gambar 8 juga dimiliki oleh Perusahaan, hanya atribut saja yang berbeda, demikian juga dengan proyek. Halaman yang tidak kalah penting adalah halaman awal yang ada pada setiap user setelah registrasi dan login akan nampak tampilan seperti yang ditunjukkan pada Gambar 9.



Gambar 9. Antarmuka Dashboard Pengguna

Pada Gambar 9 diketahui bahwa terdapat dua tampilan yang nampak saat pengguna pertama kali login pada sistem. Pada sisi kiri merupakan tabel yang menunjukkan tanggal, dan gambar pada sisi kanan merupakan grafik laporan kemajuan yang dilakukan oleh pengguna.



Gambar 10. Taskboard Pengguna

Pada Gambar 10 ditunjukkan taskboard pengguna saat pengguna memilih menu taskboard. Halaman taskboard merupakan halaman paling penting dalam aplikasi, karena pada halaman ini semua pengguna akan mengetahui aturan yang ada pada setiap jenis

pengguna. Jika pengguna menjadi manajer proyek maka akan dapat mengatur proyek secara umum, saat menjadi programmer maka pengguna dapat melihat modul apa saja yang perlu dikerjakan/telah dikerjakan/akan dikerjakan/tidak dapat dikerjakan, jika sebagai master scrum maka pengguna akan memahami bagaimana membuat jadwal rapat harian untuk mengetahui sejauh mana capain *sprint* dilakukan yang dituliskan dituliskan dalam *sprint review*.

3.2. Pengujian Fungsional

Pada tahap ini dilakukan pengujian sistem menggunakan 2 jenis pengujian, yaitu menggunakan pengujian fungsional dan pengujian secara skenario sesuai dengan task yang telah ditentukan.

Tabel 2. Pengujian Fungsional 1

Use Case	ID	Aktifitas	Hasil
Login	SP-11	Login sukses	Ok
	SP-12	Login gagal, 2 field kosong	Ok
	SP-13	Login gagal, username field kosong	Ok
Kelola Data User (Admin)	SP-21	Berhasiltambah data user	Ok
	SP-22	Gagal, username sudahterdaftar	Ok
	SP-23	Gagal, salah satu field kosong	Ok
	SP-24	Gagal, Confirm Password tidaksamadengan Password	Ok
	SP-25	Berhasil edit data user	Ok
	SP-26	Gagal, username sudahterdaftar	Ok
	SP-27	Gagal, salah satu field kosong	Ok
	SP-28	Gagal, Confirm Password tidaksamadengan Password	Ok
	SP-29	Berhasil menghapus user	Ok
	SP-30	Gagal, dibatalkan oleh user	Ok
Kelola Data Perusahaan (Admin)	SP-31	Berhasiltambah data perusahaan	Ok
	SP-32	Gagal, salah satu field kosong	Ok
	SP-33	Berhasil edit data perusahaan	Ok
	SP-34	Gagal, salah satu field kosong	Ok
	SP-35	Berhasilmenghapusperusahaan	Ok
SP-36	Gagal, dibatalkan oleh user	Ok	
Kelola Data Proyek	SP-41	Berhasiltambah data proyek	Ok
	SP-	Gagal, salah satu field kosong	Ok

(Admin)	42		
	SP-43	Berhasil edit data proyek	Ok
	SP-44	Gagal, salah satu field kosong	Ok
	SP-45	Berhasilmenghapusproyek	Ok
	SP-46	Gagal, dibatalkan oleh user	Ok
Kelola Data	SP-51	Berhasil edit data perusahaan	Ok
Perusahaan (Manajer Proyek)	SP-52	Gagal, salah satu field kosong	Ok

Pada Tabel 2 ditunjukkan hasil dari pengujian secara fungsional dalam hal pengelolaan data. Pengujian fungsional lain yang berkaitan dengan kelola data sesuai role scrum ditunjukkan pada Tabel 3.

Tabel 3. Pengujian Fungsional 2

Use Case	ID	Aktifitas	Hasil	
Kelola Data Tipe Proyek (Manajer Proyek)	SP-61	Berhasiltambah data typeproyek	V	
	SP-62	Gagal, salah satu field kosong	V	
	SP-63	Berhasil edit data type proyek	V	
	SP-64	Gagal, salah satu field kosong	V	
	SP-65	Berhasilmenghapus typeproyek	V	
	SP-66	Gagal, dibatalkan oleh user	V	
Kelola Data Pegawai (Manajer Proyek)	SP-71	Berhasiltambah data pegawai	V	
	SP-72	Gagal, username sudahterdaftar	V	
	SP-73	Gagal, salah satu field kosong	V	
	SP-74	Gagal, Confirm Password tidaksamadengan Password	V	
	SP-75	Berhasil edit data pegawai	V	
	SP-76	Gagal, username sudahterdaftar	V	
	SP-77	Gagal, salah satu field kosong	V	
	SP-78	Gagal, Confirm Password tidaksamadengan Password	V	
	SP-79	Berhasil menghapus pegawai	V	
	SP-80	Gagal, dibatalkan oleh user	V	
	Kelola Data Team (Manajer)	SP-81	Berhasiltambah data team	V
		SP-82	Berhasilmenghapus	V

Proyek)		team	
	SP-83	Gagal, dibatalkan oleh user	V
Kelola Data Proyek (Manajer Proyek)	SP-91	Berhasiltambah data proyek	V
	SP-92	Gagal, salah satu field kosong	V
	SP-93	Berhasil edit data proyek	V
	SP-94	Gagal, salah satu field kosong	V
	SP-95	Berhasilmenghapuspr oyek	V
	SP-96	Gagal, dibatalkan oleh user	V
	Kelola Data Taskboard (Manajer Proyek)	SP-101	Berhasiltambah data task
SP-102		Gagal, salah satu field kosong	V
SP-103		Berhasil edit status in progress	V
SP-104		Berhasil edit status complete	V
SP-105		Berhasil edit status need attention	V
SP-106		Berhasilmenghapuspr oyek	V
SP-107		Gagal, dibatalkan oleh user	V
Kelola Data Rapat (Master Scrum)	SP-111	Berhasiltambah data rapat	V
	SP-112	Gagal, salah satu field kosong	V
	SP-113	Berhasil edit data rapat	V
	SP-114	Gagal, salah satu field kosong	V
	SP-115	Berhasilmenghapusrapat	V
	SP-116	Gagal, dibatalkan oleh user	V
	Kelola Data Taskboard (Master Scrum)	SP-121	Berhasiltambah data task
SP-122		Gagal, salah satu field kosong	V
SP-123		Berhasil edit status in progress	V
SP-125		Berhasil edit status need attention	V
SP-126		Berhasilmenghapuspr oyek	V
SP-127		Gagal, dibatalkan oleh user	V
Kelola Data Upload Database (Analis)		SP-131	Berhasil upload file database .sql
	SP-132	Gagal, file selain .sql	V
	SP-133	Berhasil menghapus file database	V
	SP-134	Gagal, dibatalkan oleh user	V

Pada Tabel 3 ditunjukkan hasil dari pengujian fungsional yang lebih fokus pada kelola data secara detil meliputi kelola tipe proyek, kelola data pegawai sesuai role scrum, kelola data team, kelola data proyek, kelola data taskboard yang dilakukan oleh manajer proyek. Sedangkan aktifitas yang dilakukan oleh master scrum adalah kelola data rapat, kelola data taskboard, dan kelola data upload *database*. Secara keseluruhan pengujian fungsional memiliki nilai yang maksimal.

3.3. Pengujian Skenario

Pengujian yang kedua adalah pengujian secara skenario berdasarkan aktifitas pada scrum. Pada skenario pertama dijelaskan bahwa akan diketahui jika terdapat task yang tidak dapat dikerjakan yang ditunjukkan pada Gambar 11.



Gambar 11. Pengujian Skenario 1

Pada Gambar 11 ditunjukkan hasil dari pengujian skenario 1 yaitu jika terdapat task yang tidak selesai maka master scrum akan mengubah status task menjadi “need attention” dengan menekan tombol “!” sehingga task yang tidak selesai akan merujuk pada kolom “need attention” dan master scrum dapat mengubah kondisi suspended hingga jumlah jam yang dibutuhkan.

Skenario yang kedua dilakukan untuk menjawab pertanyaan mengenai task yang dalam tahap “to do” tidak bisa diselesaikan pada iterasi hari atau tanggal yang ditunjukkan pada Gambar 12.

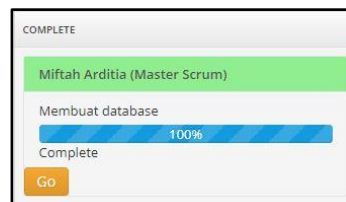


Gambar 12. Iterasi 1

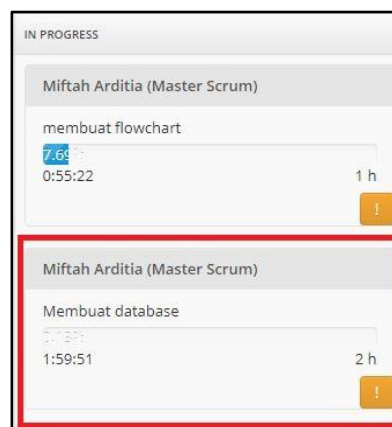
Pada Gambar 12 ditunjukkan task yang belum dikerjakan pada iterasi 1 akan otomatis ditambahkan ketika pengguna menambah iterasi berdasarkan tanggal. Sehingga pada iterasi 2 akan nampak modul

sama yang ada pada iterasi 1 dan permasalahan yang ada pada skenario dapat terselesaikan.

Pada skenario ketiga dilakukan untuk menjawab pertanyaan jika terdapat task yang sudah *complete* namun terdapat bug/kesalahan dan membutuhkan perubahan, maka task harusnya berstatus “in progress” kembali, secara detil ditunjukkan pada Gambar 13 dan Gambar 14.



Gambar 13. Kondisi Task Complete



Gambar 14. Kondisi Task In Progress

Pada Gambar 14 ditunjukkan bahwa task yang sebelumnya telah *complete* akan dalam keadaan in progress kembali jika task akan dilakukan perubahan yang ditunjukkan pada Gambar 13. Kondisi ini akan menjawab pertanyaan dari skenario ketiga terkait kondisi task yang akan diubah jika telah dalam keadaan “complete” namun terdapat bug/kesalahan atau memang perlu dilakukan perbaikan.

3.4. Pengujian CoCoMo

Pada pengujian ini dilakukan pengujian dengan menggunakan salah satu proyek yang dijalankan pada aplikasi kemudian menentukan modul berdasarkan tahapan sebelumnya. Pada pengujian ini dibandingkan antara nilai estimasi cocomo dengan nilai estimasi real. Pada aplikasi ini dapat pula diuji pada proyek yang lain tetapi pada pengujian ini digunakan proyek perangkat lunak yang telah selesai dikerjakan. Proyek yang digunakan dalam pengujian ini adalah mudik gratis online dengan nilai proyek yang tergolong kecil namun dapat digunakan untuk menguji implementasi cocomo pada aplikasi yang ditunjukkan pada tabel 4.

Tabel 4. Pengujian COCOMO

Proyek	Modul	Harga Real	Estimasi Harga CoCoMo	Waktu Real	Estimasi Waktu CoCoMo
Mudik Gratis Online	User Requirement	Rp. 3.840.000	Rp. 3.760.000	6 bulan	10 bulan
	Desain Database	Rp. 3.840.000	Rp. 3.760.000		
	Desain System	Rp. 5.120.000	Rp. 5.014.572		
	Desain Visual	Rp. 2.384.000	Rp. 2.507.286		
	Routing App #1	Rp. 2.500.000	Rp. 2.507.286		
	Admin Landing Page	Rp. 4.704.000	Rp. 5.014.572		
	Admin CRUD Operator	Rp. 3.696.000	Rp. 3.760.929		
	Admin CRUD Rute	Rp. 3.920.000	Rp. 3.760.929		
	Admin CRUD Kendaraan	Rp. 2.772.000	Rp. 2.507.286		
	Admin Approval Tiket	Rp. 3.696.000	Rp. 3.760.929		
	Admin Edit Data Pemudik	Rp. 3.696.000	Rp. 3.760.929		
	Admin Printing	Rp. 2.680.000	Rp. 2.507.286		
	Routing App #2	Rp. 2.500.000	Rp. 2.507.286		
	Visitor Landing Page	Rp. 5.280.000	Rp. 5.014.572		
	Visitor Rute	Rp. 2.556.000	Rp. 2.507.286		
	Visitor Pendaftaran	Rp. 5.280.000	Rp. 5.014.572		
	Visitor Revisi Data	Rp. 1.000.000	Rp. 1.253.643		
	Styling Admin	Rp. 3.920.000	Rp. 3.760.929		
	Styling Visitor	Rp. 3.920.000	Rp. 3.760.929		

Secara umum nilai dari pengujian ini adalah baik karena nilai waktu real lebih cepat 40% dari nilai estimasi waktu dan nilai real harga tidak ada yang lebih dari 7% dari nilai estimasi harga.

4. Simpulan

Berdasarkan penelitian yang telah dilakukan maka dapat disimpulkan beberapa hal antara lain :

- Penerapan scrum pada aplikasi dapat membantu tim pengembang dalam mengelola proyek perangkat lunak.
- Aplikasi telah berjalan sesuai dengan fungsionalitas yang diharapkan dan memberikan alternatif solusi untuk permasalahan

c. Implementasi cocomo pada aplikasi dapat membantu manajer proyek dalam estimasi biaya dan waktu.

Berdasarkan hasil dari penelitian maka saran yang diberikan untuk penelitian selanjutnya adalah aplikasi dapat diuji pada jenis usaha yang lain seperti Instansi Pemerintah namun tetap dalam kajian manajemen proyek perangkat lunak.

5. Referensi

Ali, E. Susandri. Rahmadden (2015). Sistem Informasi Akademik (SIKAD) untuk Solusi Kompleksitas Manajemen Data dan Informasi di Perguruan Tinggi. SATIN Sains dan Teknologi Informasi, 1(1), 63-68.

Azdy, R.A. Azhari, S. N. (2015). Implementasi Scrum Pada Pengembangan Software Terdistribusi. In *Seminar Nasional Informatika (SEMNASIF)* (Vol. 1, No. 2).

Duncan. W. R. (1996). *A guide to the Project Management body of knowledge - PMBoK. 1st ed. PMI Standards Committee, Upper Daryby, 1996.*

Krisnanda, M. (2015). 11. Implementasi Metodologi SCRUM dalam Pembangunan Situs Harga Komoditas. *Jurnal Sistem Informasi*, 9(2).

Mann, C., & Maurer, F. (2005). A case study on the impact of scrum on overtime and customer satisfaction. In *Agile Conference, 2005. Proceedings* (pp. 70-79). IEEE.

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

Partogi, J. (2015). *Manajemen Modern dengan Scrum*. Yogyakarta: Penerbit Andi.

Permana, P. A. G. (2015). Scrum method implementation in a software development project management. *International Journal of Advanced Computer Science and Applications*, 6(9), 198-204.

Suharso, W. Widyanto, A.N.S. Sari, Z. (2018). Sistem Informasi Pelaporan Pada UPTD Pendidikan. *Jurnal Instek*, 3(1), 101-110.

Syamsiah Ali, E., & Pudjo, D. (2014). Analisis Penerapan Network Planning Dalam Upaya Efisiensi Biaya dan Waktu Pada Penyelesaian Proyek Pengembangan Gedung RSD dr. Soebandi Jember.

Thamhain, H. J., & Wilemon, D. L. (1986). *Criteria for controlling projects according to plan. Project Management Institute. Softw. Eng. Inst.*, no. October, pp. 75-81.

Putra, F.F.A.H., Kusumo, D. S. (2011). Analisis Perbandingan dan Implementasi Metode Scrum Terhadap Metode Waterfall, 1-8.