

Penggunaan *Framework* SCRUM dalam Mencapai Optimasi *Software Development*

I Gusti Ngurah Suryantara¹⁾, Johanes Fernandes Andry²⁾, Siska³⁾, Karunia Sentosa⁴⁾

^{1,3,4)} Teknik Informatika, Fakultas Teknologi dan Desain, Universitas Bunda Mulia

²⁾ Sistem Informasi, Fakultas Teknologi dan Desain, Universitas Bunda Mulia

Jl. Lodan Raya No. 2 Ancol, Jakarta Utara 14430

¹⁾ Email: gusti@bundamulia.ac.id

²⁾ Email: jandry@bundamulia.ac.id

³⁾ Email: siskael2007@gmail.com

⁴⁾ Email: hiungsin78@gmail.com

Abstract: *A company needs to be supported with applications that support the company's business processes. Sometimes the application that is made is not in accordance with business processes, or the application that is made is not timely in completing the application by the developer. Applications created by developers sometimes do not apply a methodology, so collaboration between users and application developers is not created. This certainly has an impact on application development. The choice of methodology is very important, considering that the methodology is a framework that can provide guidance during the application development process. In this study using the Scrum framework in the application development process so as to achieve optimization during application development. By achieving optimization of application development during the development process, it will have a significant impact on the progress of an application development. In this case Scrum adopts the Agile concept which provides dynamics between application developers and users to make changes during application development. By choosing the right application development framework, it will have an impact on increasing optimization during the application development process.*

Keywords: *Frameworks, Agile, Scrum, User Stories, Backlogs, Sprints*

Abstrak: *Suatu perusahaan perlu didukung dengan aplikasi yang menunjang proses bisnis perusahaan. Terkadang aplikasi yang dibuat tidak sesuai dengan proses bisnis, atau aplikasi yang dibuat tidak tepat waktu dalam menyelesaikan aplikasi oleh developer. Aplikasi yang dibuat oleh developer terkadang tidak menerapkan suatu metodologi, sehingga tidak tercipta kolaborasi antara pengguna dan pengembang aplikasi. Hal ini tentu memberikan dampak pada pengembangan aplikasi. Pemilihan metodologi sangat penting, mengingat metodologi merupakan sebuah framework yang dapat memberikan panduan selama proses pengembangan aplikasi. Pada penelitian ini menggunakan kerangka kerja Scrum dalam proses pengembangan aplikasi sehingga mencapai optimasi selama pengembangan aplikasi. Dengan tercapai optimasi pengembangan aplikasi selama proses pengembangan akan memberikan dampak yang signifikan dalam kemajuan suatu pengembangan aplikasi. Dalam hal ini Scrum mengadopsi konsep Agile yang memberikan kedinamisan antara pengembang dan pengguna aplikasi untuk melakukan perubahan selama pengembangan aplikasi. Dengan memilih kerangka kerja pengembangan aplikasi yang tepat, maka berdampak pada peningkatan optimasi selama proses pengembangan aplikasi.*

Kata kunci: *Kerangka kerja, Agile, Scrum, User Stories, Backlog, Sprint*

I. PENDAHULUAN

Dalam pengembangan aplikasi diperlukan suatu perencanaan yang matang, sehingga aplikasi yang dibuat sesuai dengan proses bisnis. Namun tidak jarang suatu perusahaan tidak memahami skop (ruang lingkup) dari aplikasi yang akan dikembangkan (dibuat), sehingga mengakibatkan saat pengembangan aplikasi berjalan dengan lambat, dan terkadang tidak sesuai dengan proses bisnis. Faktor-faktor

tersebut yang dapat mengakibatkan terjadinya tambal sulam selama proses pengembangan aplikasi. Banyak faktor yang mengakibatkan aplikasi yang dibuat gagal. Selain kendala teknis terkadang terkendala non teknis menyebabkan gagalnya pengembangan aplikasi. Kegagalan suatu pengembangan aplikasi sangat berdampak untuk proses bisnis pada suatu perusahaan. Aplikasi yang dibuat tentu harganya mahal, sehingga sangat penting dilakukan perencanaan yang matang dalam pengembangan aplikasi. Selain dengan

perencanaan yang matang terhadap aplikasi pemilihan metodologi dalam pengembangan aplikasi juga sangat penting, karena sebuah metodologi pengembangan aplikasi akan digunakan sebagai pedoman setiap tahapan yang dilalui selama pengembangan aplikasi.

Kesalahan selama pengembangan aplikasi adalah kurang tepatnya pemilihan metodologi pengembangan aplikasi, sehingga aplikasi yang dikembangkan berjalan lambat dan sering mengakibatkan gagalnya aplikasi yang dibuat. Untuk mengatasi kegagalan pengembangan aplikasi yang diakibatkan oleh kurang tepatnya pemilihan metodologi, maka kerangka kerja Scrum merupakan sebuah metodologi yang mengadopsi konsep kerangka kerja dari manifesto kerangka kerja Agile sehingga mengakomodasi setiap perubahan yang dilakukan oleh pengguna selama pengembangan aplikasi.

Dengan keterlibatan pengguna dalam pengembangan aplikasi maka aplikasi yang dibuat akan mengurangi risiko dari tidak sesuainya dengan proses bisnis, tidak tepatan waktu selama pengembangan aplikasi. Dengan adanya metodologi pengembangan aplikasi, dalam penelitian ini menggunakan kerangka kerja Scrum maka tercipta daur hidup pengembangan aplikasi. Daur hidup pengembangan aplikasi sangat penting dalam sebuah pengembangan aplikasi yang profesional, karena daur hidup pengembangan aplikasi berisi dokumentasi setiap tahapan yang dilakukan dalam pengembangan aplikasi di sebuah perusahaan. Dengan adanya sebuah dokumentasi yang berisi daur hidup pengembangan aplikasi, maka bila ada perubahan-perubahan terhadap aplikasi yang sudah dibuat maka dapat dilihat dokumentasi yang sudah ada, sehingga perubahan yang dilakukan efektif.

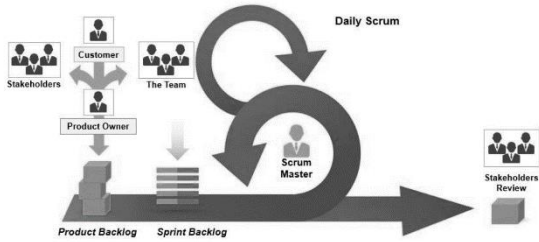
Ruang lingkup pada penelitian ini adalah: mengimplementasikan kerangka kerja Scrum dalam pengembangan aplikasi untuk mencapai optimasi selama proses pengembangan aplikasi. Yang menjadi tujuan khusus dalam penelitian ini adalah: mengembangkan aplikasi berdasarkan pendekatan *software engineering* (rekayasa perangkat lunak) dengan kerangka kerja Scrum. Sehingga antara pengembangan aplikasi dengan pengguna aplikasi saling dapat berkolaborasi selama proses pengembangan aplikasi untuk mencapai hasil yang optimal.

Yang menjadi urgensi pada penelitian ini adalah: bagaimana mencapai optimasi yang

maksimal dalam mengembangkan aplikasi dengan sebuah kerangka kerja misalkan dengan kerangka kerja Scrum, sehingga pengembangan sebuah aplikasi dapat diselesaikan dengan tepat waktu dan mendapatkan hasil yang optimal.

II. METODE PENELITIAN

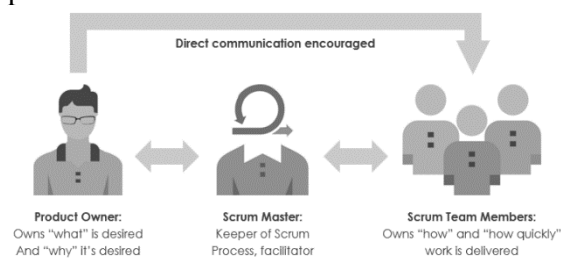
Untuk mencapai pengembangan aplikasi yang optimal berbagai metodologi berkembang dari metodologi yang tradisional sampai dengan metodologi yang modern seperti dari kerangka kerja Agile. Seiring dengan kebutuhan aplikasi yang meningkat di berbagai sektor, perusahaan, organisasi dan lainnya, maka berbagai kebutuhan metodologi juga mengalami perkembangan. Salah satu metodologi yang banyak digunakan sekarang ini adalah kerangka kerja Scrum yang merupakan kelompok dari Agile. Kerangka kerja Scrum dalam hal ini dikembangkan oleh: "Jeff Sutherland[1]", sekitar tahun 1993 dengan tujuan untuk menjadi pengembangan dan manajemen metodologi yang mengikuti prinsip-prinsip metodologi Agile (gesit tangkas). Dalam hal ini kerangka kerja Scrum merupakan bagian dari Agile[2] manifesto. Kerangka kerja Scrum adalah: salah satu metode di bidang ilmu rekayasa perangkat lunak dengan menggunakan prinsip-prinsip kerangka kerja Agile yang bertumpu pada kekuatan kolaborasi tim, *incremental product* dan proses iterasi untuk mewujudkan hasil akhir[3]. Kerangka kerja Scrum merupakan salah satu model dari *Agile Development Method* dan kerangka kerja ini sudah mulai banyak yang mengadopsi dalam pengembangan aplikasi atau dalam bidang kerekayasaan perangkat lunak, karena kerangka kerja ini memberikan kecepatan dan kedinamisan dalam pengembangan perangkat lunak. Bila menggunakan kerangka kerja Scrum kita dikenalkan dengan konsep proses kontrol untuk mengelola perubahan proyek aplikasi perangkat lunak yang "cepat[4]". Kerangka kerja Scrum menjelaskan bahwa proses pendefinisian langsung tidak dapat digunakan secara efektif dalam mengelola proyek perangkat lunak yang kompleks dan dinamis. Kerangka kerja Scrum dalam hal ini memiliki kelebihan dalam menghasilkan produk sesuai dengan pengguna[5]. Adapun proses pada kerangka kerja Scrum seperti pada tampilan Gambar 1.



Gambar 1 Kerangka kerja Scrum
 Sumber: <https://sesamedisk.com>

Kerangka kerja Scrum merupakan *framework Agile development*[6] yang banyak digunakan untuk pengembangan aplikasi, dalam menggunakan kerangka kerja Scrum untuk pengembangan aplikasi, anggota tim pengembang aplikasi dibagi menjadi tiga peran seperti berikut:

- **Product Owner:** dalam hal ini yang mewakili bisnis, pelanggan, pengguna yang mengarahkan tim pengembangan aplikasi ke arah yang tepat, yaitu sesuai dengan proses bisnis.
- **Scrum Master:** dalam hal ini berperan membantu anggota tim pengembang aplikasi dalam mengimplementasikan kerangka kerja Scrum dalam proses pengembangan aplikasi.
- **Team:** dalam hal ini merupakan grup kecil pengembang aplikasi, biasanya terdiri dari lima sampai dengan sembilan orang. Bila aplikasi yang dibuat cukup kompleks (besar), maka setiap pekerjaan dipecah pecah ke dalam bagian yang lebih kecil. Masing-masing peran digambarkan seperti pada Gambar 2.



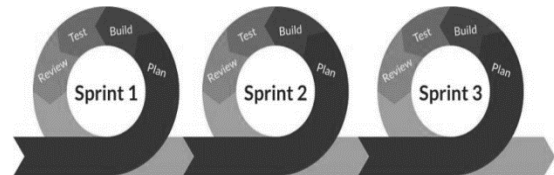
Gambar 2 Tiga peran pada Scrum
 Sumber: <https://sesamedisk.com>

Untuk mengembangkan aplikasi yang berkualitas sesuai proses bisnis dan tepat waktu pengembangannya, maka menjalankan setiap bagian (tahapan) yang ada pada kerangka kerja Scrum harus dijamin berjalan dengan baik (semua tahapan dijalankan). Sehingga setiap *sprint* yang sudah dibuat dapat diselesaikan dengan jadwal yang sudah ditentukan. Sehingga waktu pengembangan aplikasi yang sudah ditentukan targetnya dapat diselesaikan

dengan tepat waktu untuk mencapai optimasi pengembangan aplikasi. Adapun bagian-bagian penting yang menjadi proses utama dalam kerangka kerja Scrum meliputi:

- **Backlog**[7], *backlog* merupakan daftar kebutuhan atau fitur aplikasi yang dikembangkan.
- **Sprints**[8], *sprint* merupakan unit pekerjaan yang ditetapkan dalam *backlog*.
- **Scrum meeting**[9], merupakan pertemuan rutin yang membahas apa yang sudah dikerjakan, dan apa yang belum selesai dikerjakan.
- **Demo aplikasi**, merupakan presentasi aplikasi yang menandakan aplikasi sudah selesai dibuat, sehingga pengguna dapat melakukan uji coba terhadap aplikasi yang dibuat.

Sehingga digambarkan seperti tampilan pada Gambar 3.



Gambar 3 Sprint pada kerangka kerja Scrum
 Sumber: <https://dsruptr.com>

A. Desain Penelitian

Pada penelitian ini melakukan kajian implementasi kerangka kerja Scrum pada pengembangan aplikasi untuk mencapai optimasi yang optimal, sehingga aplikasi yang dibuat dapat selesai dengan tepat waktu dan sesuai dengan proses bisnis. Pemilihan kerangka kerja Scrum dikarenakan setiap tahapan pekerjaan yang dilakukan selama pengembangan aplikasi diturunkan (*breakdown*) ke dalam *product backlog*, masing-masing bagian dari *product backlog* diturunkan lagi menjadi *sprint*, setiap *sprint* di kuantisasi ke dalam satuan waktu jam dalam 5 hari kerja dalam satu minggu. Dengan menggunakan kerangka kerja Scrum setiap perjalanan *sprint* dapat dimonitor (dipantau) dengan baik, apakah selama menjalankan *sprint* ada kendala atau *sprint* dapat berjalan dengan baik. Bila ada kendala maka dilakukan *review* terhadap *sprint* yang tidak berjalan sebagaimana mestinya. Sehingga kendala penyebab suatu *sprint* tidak berjalan optimal dapat dihilangkan. Sehingga setiap *sprint* dapat berjalan sesuai dengan yang sudah dijadwalkan.

Setiap kerangka kerja memiliki tahapan yang dilalui, begitu-pun kerangka kerja Scrum memiliki tahapan-tahapan yang harus dilalui dengan baik dan teliti sehingga setiap *sprint* yang ada pada *product backlog* dijalankan dan *done*. Dalam penurunan kerangka kerja Scrum dimulai dari pembuatan *user stories*, *product backlog*, *sprints*, *stakeholders review*.

- **User Stories**: *user stories* merupakan cerita pengguna (*user*) yang berisi apa saja yang terdapat pada aplikasi, sehingga dari *user stories* ini pengembang akan membuat *product backlog*.
- **Product Backlog**[10]: dari *user stories* yang dibuat, maka akan dibuatkan *product backlog* yang berisi susunan *product backlog*.
- **Sprints**[11]: setelah membuat *product backlog*, maka dari *product backlog* tersebut di bagi-bagi ke dalam *sprint*. Setiap *sprint* akan dijalankan secara iterasi selama proses pengembangan aplikasi, sampai semua *sprint* selesai dijalankan dan proses pengembangan aplikasi *done*.
- **Stakeholders**[12] *review*: bila semua *sprint* sudah selesai dijalankan maka akan di *review* oleh *stakeholders*.

B. Objek Penelitian

Yang menjadi objek pada penelitian ini adalah mengimplementasikan kerangka kerja Scrum dalam pengembangan aplikasi untuk mencapai optimasi yang optimal. Dengan mengimplementasikan kerangka kerja Scrum dalam pengembangan aplikasi, maka daur hidup pengembangan aplikasi dapat tercipta. Dengan menjalankan setiap *sprint* pada *product backlog* pada kerangka kerja Scrum, maka dapat dihitung setiap *sprint* tersebut sehingga optimasi dari setiap tahapan pengembangan aplikasi dapat diukur. Setiap akhir dari *sprint* akan dibuatkan grafik *burndown* yang menandakan suatu *sprint* sudah selesai dijalankan. Bila satu *sprint* sudah selesai dijalankan maka akan dilanjutkan dengan menjalankan *sprint* berikutnya, dan hal ini dilakukan secara berulang (*iterative*) sampai semua *sprint* pada *product backlog* selesai dijalankan (*done*). Dengan Seleksi seluruh *sprint* dijalankan maka hal tersebut menandakan seluruh proses pengerjaan aplikasi sudah selesai.

C. Prosedur Penelitian

Adapun prosedur penelitian yang dilakukan pada penelitian ini memiliki tahapan yang ada pada kerangka kerja Scrum yaitu:

- Diawali dengan pembuatan *user stories*, untuk menggali cerita pengguna mengenai fitur yang ada pada aplikasi.
- Membuat *product backlog*, berdasarkan *user stories* maka di petakan ke dalam *product backlog*.
- Membuat *sprint*, dari *product backlog* maka dipetakan ke dalam bentuk yang lebih rinci (detail) berupa *sprint*.
- *Scrum meeting*, untuk memastikan apakah setiap *sprint* yang sudah dibuat berjalan dengan baik selama proses pengembangan aplikasi maka diadakan *meeting* harian.
- Mendemokan aplikasi, untuk menguji aplikasi pada pengguna maka dilakukan dengan mendemokan aplikasi. Hal ini untuk memastikan bahwa aplikasi berjalan sesuai dengan bisnis proses yang berjalan.

Pada Gambar 4 menampilkan fase-fase prosedur penelitian yang dikaitkan dengan kerangka kerja Scrum.



Gambar 4 Prosedur penelitian

D. Metodologi Penelitian dan Pengumpulan Data

Dengan adanya prosedur penelitian maka setiap tim yang terlibat dalam penelitian ini menjalankan perannya masing-masing sesuai dengan tugas yang sudah diberikan. Setiap tahapan pada kerangka kerja Scrum dipastikan berjalan dan setiap *sprint* dipastikan berjalan dan selesai dengan bobot waktu yang sudah disusun pada saat pembuatan *product backlog*. Sehingga setiap tahapan yang ada pada kerangka kerja Scrum dilakukan secara detail, dan tidak ada yang terlewatkan untuk memastikan setiap *sprint* berjalan dengan baik. Metode pengumpulan data terdiri dari observasi, wawancara dan diskusi yang terfokus kepada pemangku kebijakan. Dalam pengumpulan data yang dilakukan dalam penelitian ini seperti berikut:

- Melakukan Kajian Pustaka, hal ini dilakukan untuk mendapatkan dokumen-dokumen yang berkaitan dengan proses pengembangan aplikasi dengan kerangka kerja Scrum.

- Melakukan Wawancara, melakukan wawancara dalam pengumpulan data sebagai bahan untuk mengetahui kebutuhan pengguna terhadap aplikasi yang akan dibuat.
- Melakukan Pengamatan, pengamatan bertujuan untuk mendapatkan gambaran proses bisnis yang berjalan, sehingga dalam pembuatan *product backlog* dan *sprint* dapat dipetakan dengan detail dan tepat untuk menjamin *sprint* dapat berjalan dengan baik.

III. HASIL DAN PEMBAHASAN

Pada tahap ini dilakukan penurunan secara rinci (*breakdown*) setiap tahapan yang ada pada kerangka kerja (*framework*) Scrum. Para pembuat aplikasi (praktisi, akademisi, konsultan) menggunakan kerangka kerja dalam pembuatan aplikasi untuk mencapai optimasi yang optimal dalam pembuatan maupun dalam pengembangan aplikasi (perangkat lunak). Pada bidang kerekayasaan perangkat lunak (*software engineering*) saat ini lebih banyak menggunakan salah satu dari kelompok metodologi Agile seperti XP (*eXtreme Programming*), Scrum, dan lainnya. Pada penelitian ini menggunakan kerangka kerja Scrum untuk mencapai optimasi dalam pembuatan atau pengembangan aplikasi. Kerangka kerja Scrum dipilih karena pada kerangka kerja Scrum memiliki tahapan yang dimodelkan dari *product backlog* menjadi *sprint*, sehingga setiap *sprint* dapat dipetakan dengan satuan hari dan jam. Dengan adanya pemetaan ke dalam hari atau jam setiap fase dari tahapan yang dilalui dalam pembuatan aplikasi data diukur sehingga dapat diketahui dari awal bagian mana yang belum optimal, dan bagian mana yang sudah mencapai optimal.

Dengan adanya pemetaan setiap fase ke dalam satuan hari dan jam pada tahap pembuatan aplikasi dapat diketahui dari awal progres (kemajuan) dari setiap *sprint* yang dijalankan. Tahapan yang dilakukan meliputi:

- Membuat *user stories*.
- Menentukan *product backlog*.
- Menentukan *sprint*.
- Melakukan *Scrum meeting*.
- Demo aplikasi.

A. Membuat *User Stories*

Pada tahap ini (tahap awal) dilakukan pemetaan terhadap kebutuhan pengguna terhadap fitur aplikasi yang dibuat, hal ini dilakukan dalam bentuk pembuatan *user stories*. Pada Tabel 1 menampilkan *user stories*, yang nantinya dapat digunakan sebagai dasar (acuan) dalam membuat *product backlog*. Selama pengembangan *user stories* data-data dikumpulkan dari pengguna untuk mendapatkan apa saja yang menjadi kebutuhan pengguna terhadap aplikasi yang akan dibuat. *User stories* merupakan fase awal yang sangat penting dalam menerapkan kerangka kerja Scrum, karena untuk mengimplementasikan kerangka kerja Scrum sangat detail dalam memetakan ke dalam kuantisasi waktu masing-masing *task* yang ada dalam masa pengembangan aplikasi. Dalam penyusunan *user stories* dimulai dari penentuan item-item *user stories*, kemudian setiap item dari *user stories* di beri keterangan untuk memperjelas apa yang menjadi kebutuhan dari pengguna terhadap fitur yang ada pada aplikasi.

Tabel 1 *User Stories*

No	User Stories	Keterangan
1	Mengelola data jurnal	Sebagai pengguna dapat memasukkan data jurnal
2	Menampilkan buku besar	Sebagai pengguna dapat menampilkan data buku besar
3	Menampilkan neraca saldo	Sebagai pengguna dapat menampilkan data neraca saldo
4	Menampilkan neraca lajur	Sebagai pengguna dapat menampilkan neraca lajur
5	Menampilkan rugi/laba	Sebagai pengguna dapat menampilkan rugi/laba
6	Menampilkan perubahan ekuitas, neraca, dan neraca penutupan	Sebagai pengguna dapat menampilkan data perubahan ekuitas, neraca, dan neraca penutupan

B. Menentukan *Product Backlog*

Tahap selanjutnya adalah menentukan (membuat) *product backlog*. Yang menjadi acuan dalam membuat *product backlog* pada fase ini bisa menggunakan *user stories*. Adapun *product backlog* yang dibuat seperti pada

tampilan Tabel 2. Penentuan *product backlog* sangat penting, karena hal ini menjadi dasar acuan dalam pembuatan *sprint*. Pembuatan *product backlog* dimulai dari penyusunan deskripsi fitur, kemudian menentukan skala prioritas, selanjutnya menentukan estimasi waktu yang dibutuhkan untuk menjalankan satu task pada *product backlog* yang ditandai dengan pemetaan lebih rinci pada setiap *sprint*.

Tabel 2 *Product Backlog*

No	Deskripsi Fitur	Prioritas	Estimasi Waktu (Jam)
1	Mengelola data jurnal	Sedang	60
2	Menampilkan buku besar	Tinggi	90
3	Menampilkan neraca saldo	Tinggi	35
4	Menampilkan neraca lajur	Sangat Tinggi	50
5	Menampilkan rugi/laba	Sangat Tinggi	44
6	Menampilkan perubahan ekuitas, neraca, dan neraca penutupan	Sangat Tinggi	60

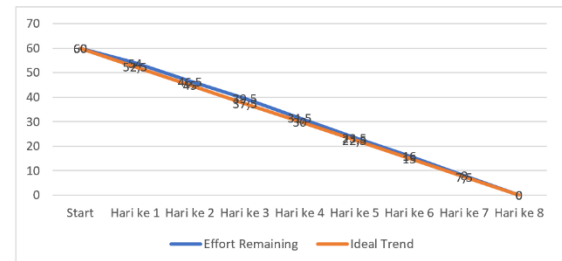
C. Menentukan *Sprint*

Setelah menentukan *product backlog*, maka fase selanjutnya melakukan *breakdown* pada setiap *product backlog* pada Tabel 2 ke dalam periode waktu. Berdasarkan *product backlog* maka dipetakan ke dalam enam *sprint*. *Sprint* 1 merupakan *task* pertama dari *product backlog*, adapun hasil dari *breakdown* (penurunan lebih rinci) pada *task* pertama di *product backlog* ke dalam bentuk *sprint* seperti pada Tabel 3. *Sprint* 1 diperkirakan dapat diselesaikan dalam durasi 60 jam. Bila dikaitkan dengan aplikasi yang akan dibuat, maka *sprint* 1 adalah bagian pembuatan *file master* dan transaksi (pada penelitian ini adalah transaksi jurnal). *File master* merupakan bagian-bagian yang nantinya akan menjadi sumber data pada saat dilakukan transaksi pada aplikasi melalui menu transaksi. Sehingga bagian *file master* dimasukkan sebagai bagian dari *sprint* 1.

Tabel 3 *Sprint 1 – Mengelola Data Jurnal*

Fitur Backlog	Task	Estimasi Waktu (Jam)
Mengelola data jurnal	Divisi keuangan dapat melakukan <i>entry</i> data periode akuntansi.	14
	Divisi keuangan dapat melakukan <i>entry</i> data jurnal.	15
	Divisi keuangan dapat melakukan <i>entry</i> data jurnal penyesuaian.	15
	Divisi keuangan dapat melakukan <i>entry</i> data saldo periode tahun lalu.	16
Total		60

Sprint 1 seperti pada Tabel 3, pada *sprint* 1 terdiri dari empat *task* dengan total perkiraan waktu 60 jam. Durasi waktu 60 jam dijalankan dalam 8 hari kerja. Tanda selesai *sprint* 1 dikerjakan maka ditandai dengan membuat *burndown*, *burndown sprint* 1 seperti pada Gambar 5.



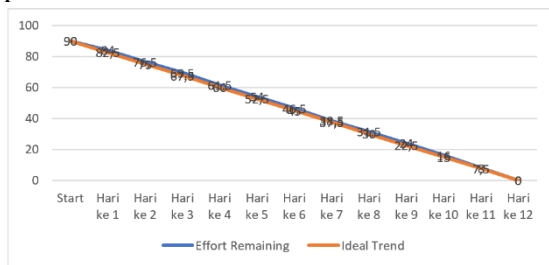
Gambar 5 *Burndown Sprint 1*

Bila *sprint* 1 selesai dijalankan maka dilanjutkan pada fase *sprint* 2. *Sprint* 2 merupakan *task* kedua dari *product backlog*, adapun hasil dari *breakdown* (penurunan lebih rinci) pada *task* kedua di *product backlog* ke dalam bentuk *sprint* seperti pada Tabel 4. *Sprint* 2 diperkirakan dapat diselesaikan dalam durasi 90 jam. Bila dikaitkan dengan aplikasi yang akan dibuat, maka *sprint* 2 adalah bagian melakukan *posting* terhadap data transaksi. *Posting* merupakan proses semua data jurnal yang dimasukkan lewat transaksi jurnal. Sehingga bagian ini dijadikan *sprint* 2. Pada *sprint* 2 juga dapat menampilkan data laporan buku besar jurnal umum dan laporan jurnal penyesuaian. Gambaran *sprint* 2 seperti pada Tabel 4.

Tabel 4 Sprint 2 – Menampilkan Buku Besar

Fitur Backlog	Task	Estimasi Waktu (Jam)
Menampilkan buku besar	Divisi keuangan dapat melakukan <i>posting</i> periode akuntansi.	30
	Divisi keuangan dapat menampilkan laporan buku besar jurnal umum pada setiap periode akuntansi.	30
	Divisi keuangan dapat menampilkan laporan buku besar jurnal penyesuaian pada setiap periode akuntansi.	30
Total		90

Sprint 2 terdiri dari 3 *task* dengan total estimasi waktu 90 jam. Durasi waktu 90 jam dijalankan dalam 12 hari kerja. Tanda selesai *sprint 2* dikerjakan maka ditandai dengan membuat *burndown*, *burndown sprint 2* seperti pada Gambar 6.



Gambar 6 Burndown Sprint 2

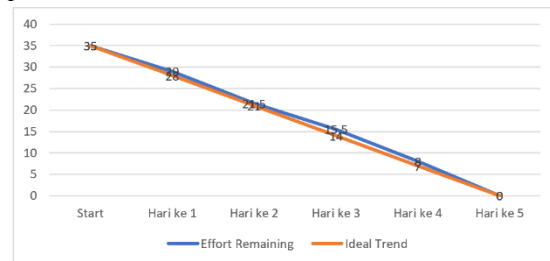
Bila *sprint 2* selesai dikerjakan maka dilanjutkan dengan mengerjakan *sprint 3*. *Sprint 3* merupakan *task* ketiga dari *product backlog*, adapun hasil dari *breakdown* (penurunan lebih rinci) pada *task* ketiga di *product backlog* ke dalam bentuk *sprint* seperti pada Tabel 5. *Sprint 3* diperkirakan dapat diselesaikan dalam durasi 60 jam. Bila dikaitkan dengan aplikasi yang akan dibuat, maka *sprint 3* adalah menampilkan neraca saldo

jurnal umum dan neraca saldo jurnal penyesuaian. Laporan neraca saldo merupakan hasil dari *posting* jurnal umum dan jurnal penyesuaian. Sehingga bagian ini dijadikan *sprint 3*. *Sprint 3* seperti pada tampilan Tabel 5.

Tabel 5 Sprint 3 – Menampilkan Neraca Saldo

Fitur Backlog	Task	Estimasi Waktu (Jam)
Menampilkan neraca saldo	Divisi keuangan dapat menampilkan neraca saldo jurnal umum.	15
	Divisi keuangan dapat menampilkan neraca saldo jurnal umum.	20
Total		35

Sprint 3 terdiri dari dua *task* dengan total estimasi waktu 35 jam. Durasi waktu 35 jam dijalankan dalam 5 hari kerja. Tanda selesai *sprint 3* dikerjakan maka ditandai dengan membuat *burndown*, *burndown sprint 3* seperti pada Gambar 7.



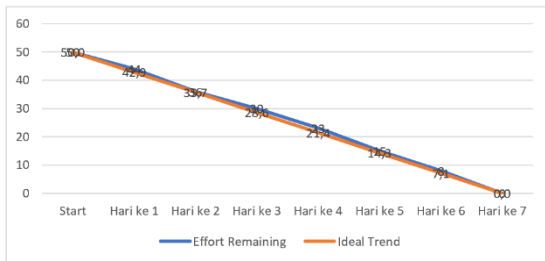
Gambar 7 Burndown Sprint 3

Bila *sprint 3* selesai dikerjakan dilanjutkan dengan mengerjakan *sprint 4*. *Sprint 4* merupakan *task* keempat dari *product backlog*, adapun hasil dari *breakdown* (penurunan lebih rinci) pada *task* keempat di *product backlog* ke dalam bentuk *sprint* seperti pada Tabel 6. *Sprint 4* diperkirakan dapat diselesaikan dalam durasi 50 jam. Bila dikaitkan dengan aplikasi yang akan dibuat, maka *sprint 4* adalah menampilkan neraca lajur. Laporan neraca lajur merupakan hasil dari proses *posting*. Sehingga bagian ini dijadikan *sprint 4*. *Sprint 4* seperti pada tampilan Tabel 6.

Tabel 6 Sprint 4 – Menampilkan Neraca Lajur

Fitur Backlog	Task	Estimasi Waktu (Jam)
Menampilkan neraca lajur	Divisi keuangan dapat melihat neraca lajur	50
Total		50

Sprint 4 terdiri dari 1 task dengan total estimasi waktu 50 jam. Durasi waktu 50 jam dijalankan dalam 7 hari kerja. Tanda selesai sprint 4 dikerjakan maka ditandai dengan membuat burndown, burndown sprint 4 seperti pada Gambar 8.



Gambar 8 Burndown Sprint 4

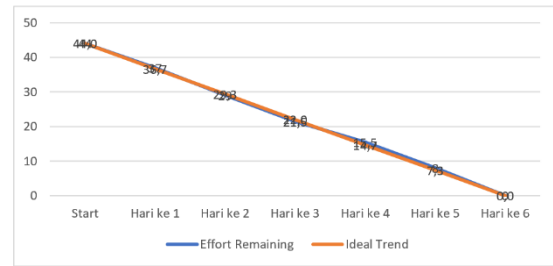
Bila sprint 4 selesai dikerjakan maka dilanjutkan dengan mengerjakan sprint 5. Sprint 5 merupakan task kelima dari product backlog, adapun hasil dari breakdown (penurunan lebih rinci) pada task kelima di product backlog ke dalam bentuk sprint seperti pada Tabel 7. Sprint 5 diperkirakan dapat diselesaikan dalam durasi 44 jam. Bila dikaitkan dengan aplikasi yang akan dibuat, maka sprint 5 adalah menampilkan rugi/laba. Laporan rugi/laba merupakan hasil dari proses posting. Sehingga bagian ini dijadikan sprint 5. Sprint 5 seperti pada tampilan Tabel 7.

Tabel 7 Sprint 5 – Menampilkan Rugi/Laba

Fitur Backlog	Task	Estimasi Waktu (Jam)
Menampilkan rugi / laba	Divisi keuangan dapat menampilkan rugi/laba per periode akuntansi	44
Total		44

Sprint 5 terdiri dari 1 task dengan total estimasi waktu 44 jam. Durasi waktu 44 jam

dijalankan dalam 6 hari kerja. Tanda selesai sprint 5 dikerjakan maka ditandai dengan membuat burndown, burndown sprint 5 seperti pada gambar 9.



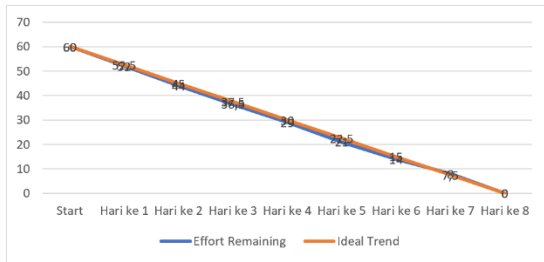
Gambar 9 Burndown Sprint 5

Bila sprint 5 selesai dikerjakan maka dilanjutkan dengan mengerjakan sprint 6. Sprint 6 diperkirakan dapat diselesaikan dalam durasi 60 jam. Bila dikaitkan dengan aplikasi yang akan dibuat, maka sprint 6 adalah menampilkan perubahan ekuitas, neraca, dan neraca penutupan. Laporan perubahan ekuitas, neraca, dan neraca penutupan merupakan hasil dari proses posting. Sehingga bagian ini dijadikan sprint 6. Sprint 6 seperti pada tampilan Tabel 7.

Tabel 8 Sprint 6 – Menampilkan Perubahan Ekuitas, Neraca, dan Neraca Penutupan

Fitur Backlog	Task	Estimasi Waktu (Jam)
Menampilkan Perubahan Ekuitas, Neraca, dan Neraca Penutupan	Divisi keuangan dapat melihat laporan perubahan ekuitas	30
	Divisi keuangan dapat melihat laporan neraca	30
	Divisi keuangan dapat melihat neraca penutupan	30
Total		60

Sprint 6 ada 3 task total estimasi waktu 60 jam. Durasi waktu 90 jam dijalankan dalam 8 hari kerja. Tanda selesai sprint 2 dikerjakan maka ditandai dengan membuat burndown, burndown sprint 6 seperti pada gambar 10.



Gambar 10 Burndown Sprint 6

D. SCRUM Meeting

Fase yang menjadi rutinitas harian adalah *meeting*. Setiap hari melakukan *meeting* harian untuk mengetahui kemajuan setiap *sprint*. Jadwal ini akan dijalankan selama pengembangan aplikasi. Dengan adanya *meeting* harian digunakan untuk memantau kemajuan *sprint*, apakah selama menjalankan *sprint* ada kendala, atau apa yang sudah dikerjakan, dan apa yang sudah diselesaikan. Evaluasi dilakukan dalam *meeting* harian, durasi *meeting* sekitar lima belas menit.

E. Demo Aplikasi

Fase selanjutnya adalah mendemokan aplikasi sekaligus melakukan *testing* (pengujian) pada aplikasi. *Testing* aplikasi yang dilakukan adalah dengan *black box testing* dan *white box testing*. Pada saat awal menjalankan aplikasi keuangan maka akan tampil kotak dialog untuk *login* ke aplikasi seperti pada Gambar 11. Masukkan nama pengguna dan *password* untuk *login* ke aplikasi.



Gambar 11 Login ke Aplikasi

Bila nama pengguna dan *password* valid maka akan tampil menu utama aplikasi seperti pada Gambar 12.



Gambar 12 Menu Aplikasi

Pada menu utama aplikasi terdiri dari sub menu, seperti: *file*, *master*, *setup*, transaksi, proses, laporan, dan *tools*. Untuk sub menu *file* pengguna dapat melakukan *login* dan *exit*. Menu *file* terdiri dari sub menu *login* dan *exit*. Pada sub menu *master* terdiri dari sub menu seperti: nama usaha, wajib pajak, daftar nama perkiraan, periode, dan *user*. Nama usaha digunakan untuk mendata identitas perusahaan, wajib pajak digunakan untuk mendata wajib pajak, daftar perkiraan digunakan untuk mendata daftar nama perkiraan, periode untuk periode akuntansi, dan mendata pengguna. Pada sub menu *setup* terdiri dari *setup* untuk saldo awal. *Setup* saldo awal hanya dilakukan sekali pada saat awal menggunakan aplikasi, selanjutnya saldo awal akan diisi secara otomatis pada saat melakukan proses *tutup* buku (tutup periode akuntansi) saat melakukan proses *posting*. Pada sub menu transaksi memiliki sub menu dengan nama jurnal umum, dan AJP (ayat jurnal penyesuaian). Sub menu transaksi ini digunakan untuk memasukkan data-data yang berkaitan dengan jurnal umum, dan ajp. Data dari jurnal umum, AJP, dan saldo awal nantinya akan sebagai data masukkan pada saat proses *posting*. Pada sub menu proses merupakan proses *posting* terhadap satu siklus akuntansi. Apabila melakukan *posting* maka hal itu dimaksudkan untuk menutup satu siklus periode akuntansi. Proses *posting* hanya dilakukan sekali. Pada sub menu laporan untuk menampilkan semua hasil proses *posting*, pada menu laporan dapat melakukan: mencetak nomor perkiraan, jurnal, buku besar, neraca saldo, neraca lajur, laba/rugi, perubahan ekuitas, neraca, neraca penutupan, dan catatan laporan keuangan. Pada sub menu *tools* digunakan untuk membersihkan data-data latihan,

Pengisian data nama usaha seperti pada Gambar 13. Pada Gambar 13 digunakan untuk mendapat identitas perusahaan.



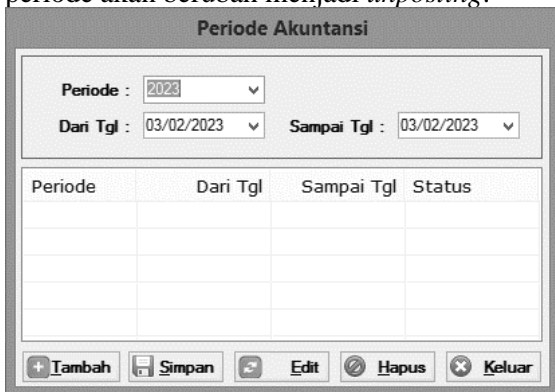
Gambar 13 Nama Usaha

Selanjutnya adalah mendata nama perkiraan (nama akun) yang akan digunakan dalam penyusunan laporan keuangan seperti membuat jurnal umum, jurnal penyesuaian, dan *setup* saldo awal, seperti pada Gambar 14.



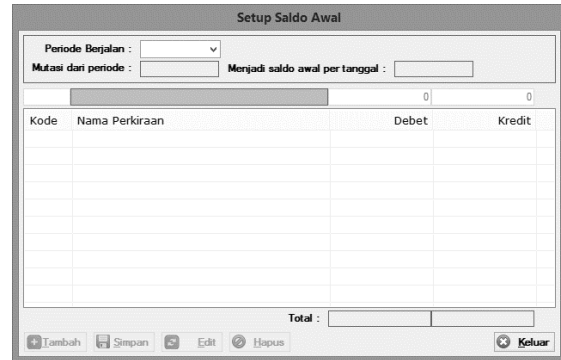
Gambar 14 Nama Perkiraan

Membuat periode akuntansi, periode akuntansi merupakan satu siklus akuntansi. Pada aplikasi ini memungkinkan untuk membuat periode akuntansi sesuai dengan kebutuhan, apakah per bulan, atau per empat bulan, per enam bulan, atau satu tahun. Adapun format untuk membuat seperti tampilan pada Gambar 15. Status periode akuntansi adalah *unposting* dan *posting*, bila *unposting* maka periode yang dibuat merupakan periode berjalan dalam satu siklus akuntansi, dan bila sudah menjalankan proses *posting* maka status periode akan berubah menjadi *unposting*.



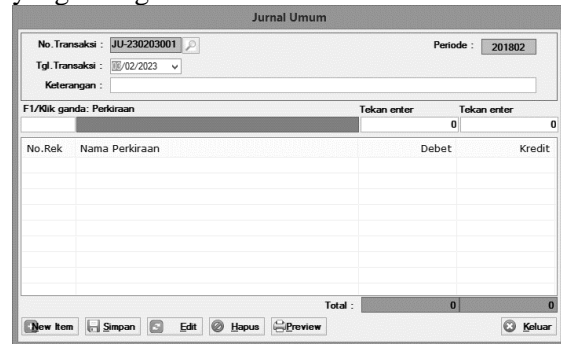
Gambar 15 Periode Akuntansi

Mendata pengguna (*user*) yang berhak menggunakan aplikasi, pendataan pengguna seperti tampilan pada Gambar 16. Setup saldo awal digunakan untuk pertama kali menggunakan aplikasi, dan akan digunakan untuk mengisi saldo awal periode akuntansi. Setup ini dilakukan hanya sekali pada saat pertama aplikasi digunakan. Tampilan setup aplikasi seperti tampilan pada Gambar 17.



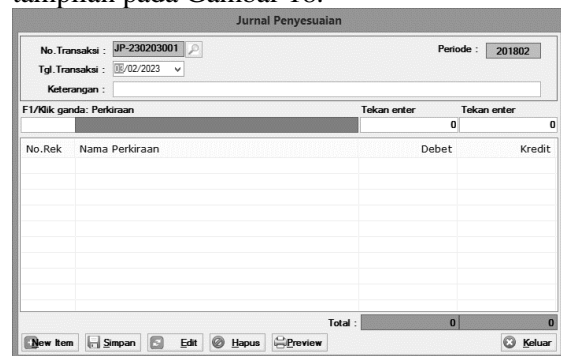
Gambar 16 Setup Saldo Awal

Setelah semua *file master* dan setup saldo awal dilakukan maka aplikasi sudah bisa digunakan untuk memasukkan data-data yang berkaitan dengan jurnal umum. Untuk melakukan *entry* jurnal umum seperti pada Gambar 17. Pada jurnal umum mendata jurnal yang di bagian debet dan kredit.



Gambar 17 Jurnal Umum

Bila ada yang perlu dilakukan penyesuaian pada jurnal, maka dapat dilakukan pada bagian pendataan jurnal penyesuaian. Untuk melakukan *entry* jurnal penyesuaian seperti tampilan pada Gambar 18.



Gambar 18 Jurnal Penyesuaian

Bila semua data jurnal umum dan jurnal penyesuaian sudah di masukkan, selanjutnya melakukan proses *posting*. Proses *posting* seperti tampilan pada Gambar 19.



Gambar 19 Proses Posting

Prototype laporan yang akan dihasilkan seperti pada Gambar 20. Gambar 20 menampilkan prototype laporan jurnal umum. Dalam jurnal umum total debet dan kredit sama.

Jurnal Umum				
Periode :				
Dari Tanggal :		s/d Tanggal :		
Kode	Nama Perkiraan	D/K	Debet	Kredit
No. Transaksi :				
Tanggal :				
Keterangan :				

Gambar 20 Prototype Laporan Jurnal Umum

Prototype laporan jurnal penyesuaian yang akan dihasilkan seperti pada Gambar 21. Gambar 21 menampilkan prototype laporan jurnal penyesuaian, jumlah debet dan kredit pada jurnal penyesuaian harus sama (balance).

JURNAL PENYESUAIAN				
Periode :				
Dari Tanggal :		s/d Tanggal :		
Kode	Nama Perkiraan	DK	Debet	Kredit
No. Transaksi :				
Tanggal :				
Keterangan :				

Gambar 21 Prototype Laporan Jurnal Penyesuaian

Prototype laporan buku besar jurnal umum seperti tampilan pada Gambar 22.

BUKU BESAR JURNAL UMUM					
Periode :					
Dari Tanggal :		s/d Tanggal :			
No. Transaksi	Tanggal	Keterangan	D/K	Debet	Kredit

Gambar 22 Prototype Laporan Buku Besar Jurnal Umum

Prototype laporan buku besar jurnal penyesuaian seperti tampilan pada Gambar 23.

Buku Besar AJP					
Periode :					
Dari Tanggal :		s/d Tanggal :			
No. Transaksi	Tanggal	Keterangan	D/K	Debet	Kredit

Gambar 23 Prototype Laporan Buku Besar Jurnal Umum

Prototype laporan neraca saldo jurnal umum seperti tampilan pada Gambar 24.

Neraca Saldo Jurnal Umum			
Periode :			
Dari Tanggal :		s/d Tanggal :	
No. Perkiraan	Nama Perkiraan	Debet	Kredit

Gambar 24 Laporan Neraca Saldo Jurnal Umum

Prototype laporan neraca saldo jurnal penyesuaian seperti tampilan pada Gambar 25.

Neraca Saldo AJP			
Periode :			
Dari Tanggal :		s/d Tanggal :	
No. Perkiraan	Nama Perkiraan	Debet	Kredit

Gambar 25 Laporan Neraca Saldo Jurnal Penyesuaian

Prototype laporan neraca lajur seperti tampilan pada Gambar 26.

Neraca Lajur											
Periode :											
Dari Tanggal :		s/d Tanggal :									
Kode	Nama Perkiraan	Neraca Saldo		AJP		N. Saldo Diuraikan		Rugi / Laba		Neraca	
		Debet	Kredit	Debet	Kredit	Debet	Kredit	Debet	Kredit	Debet	Kredit

Gambar 26 Prototype Laporan Neraca Lajur

Prototype laporan rugi / laba seperti pada Gambar 27. Dengan melihat laporan rugi / laba maka dapat diketahui mengalami keuntungan atau kerugian dalam satu periode akuntansi.

Rugi / Laba	
Periode : 202302	
Dari Tanggal : 01/02/2023 s/d Tanggal : 28/02/2023	
Pendapatan	
Pendapatan Usaha	0
Total Pendapatan	0
Biaya-Biaya	
Biaya Gaji	0
Total Biaya-Biaya	0
Rugi / Laba Bersih :	0

Gambar 27 Prototype Laporan Rugi/Laba

Prototype laporan perubahan ekuitas seperti pada Gambar 28. Laporan ini bertujuan untuk mengetahui perubahan ekuitas selama satu periode akuntansi.

Perubahan Modal	
Periode : 202302	
Dari Tanggal : 01/02/2023 s/d Tanggal : 28/02/2023	
Modal	0
Lababersih	0
Prive	0
Penambahan Modal	0 +
Modal akhir per periode 202302	0

Gambar 28 Prototype Laporan Perubahan Modal

Prototype laporan neraca seperti pada Gambar 29. Laporan neraca digunakan untuk mengetahui posisi pasiva dan aktiva perusahaan selama satu periode akuntansi.

Neraca	
Periode: 202302	
Dari Tanggal : 01/02/2023	s/d Tanggal : 28/02/2023
Pasiva	
Modal	
Rugi/Laba bersih	0
Laba / Rugi Bersih	0

Gambar 29 Prototype Laporan Neraca

Prototype laporan neraca saldo penutupan seperti pada Gambar 30. Laporan neraca saldo penutupan digunakan untuk memindahkan saldo-saldo satu periode akuntansi yang sudah di *posting* (ditutup) ke dalam saldo awal pada periode berikutnya.

NERACA SALDO PENUTUPAN			
Untuk Periode Yang Berakhir Pada Tanggal:			
Menjadi Saldo Awal Pada Periode :			
Kode	Nama Perkiraan	Debet	Kredit

Gambar 30 Prototype Saldo Penutupan

IV. SIMPULAN

Dengan mem-*breakdown* kerangka kerja (*framework*) Scrum pada penelitian ini maka tercipta daur hidup pengembangan aplikasi (*system development life cycle*) berupa aplikasi keuangan. Adapun daur hidup pengembangan aplikasi dimulai dari pembuatan *user stories*, kemudian dilanjutkan dengan pembuatan *product backlog*, dari *product backlog* dijadikan *sprint*. Adapun *sprint* yang dibuat terdiri dari enam *sprint*, setiap *sprint* dijalankan dari *sprint* satu sampai dengan *sprint* enam. Dari *sprint* satu sampai dengan *sprint* enam membutuhkan waktu 339 Jam, dalam 46 hari kerja, dalam 46 hari kerja sama dengan 6,6 minggu. Untuk melakukan pengujian aplikasi dilakukan dengan pengujian *black box testing* dan *unit testing*. Hasil *testing* yang dilakukan pada aplikasi, dapat disimpulkan berjalan dengan baik, hal ini dikarenakan setiap fase menjalankan *sprint* sudah dilakukan pengujian secara *unit testing*, sehingga bila ada tidak sesuai pada fase pengembangan sudah dapat diketahui dari awal. Dapat dikatakan bahwa dengan melakukan tahapan yang ada pada *sprint* optimasi pengembangan aplikasi dapat diketahui secara rinci dan terukur dengan adanya *burndown* pada setiap *sprint*.

DAFTAR RUJUKAN

[1] D. Fernando, A. Anharudin, and F. Fadli, "Rancang Bangun Aplikasi E-Portofolio Hasil Karya Mahasiswa Unsera Menggunakan Metode Scrum,"

JSiI (Jurnal Sist. Informasi), vol. 5, no. 1, pp. 7–12, 2018, doi: 10.30656/jsii.v5i1.579.

[2] U. Ependi, "Implementasi Model Scrum pada Sistem Informasi Seleksi Masuk Mahasiswa Politeknik Pariwisata Palembang," J. Inform. J. Pengemb. IT, vol. 3, no. 1, pp. 49–55, 2018, [Online]. Available: <http://ejournal.poltektegal.ac.id/index.php/informatika/article/view/640>.

[3] A. Andipradana and K. Dwi Hartomo, "Rancang Bangun Aplikasi Penjualan Online Berbasis Web Menggunakan Metode Scrum," J. Algorithm., vol. 18, no. 1, pp. 161–172, 2021, doi: 10.33364/algorithm/v.18-1.869.

[4] N. Rafianto, Dimas, and Saifulloh, "Penerapan Metode Scrum Pada Pembuatan User Experience Landing Page Sistem Informasi Lentera," J. Sist. Inf. dan Sains Teknol., vol. 3, no. 2, pp. 1–14, 2021.

[5] Afriyan Firdaus, "Penerapan Scrum Agile Development Dalam Pengembangan Sistem Informasi Monitoring Mahasiswa Bidikmisi Berbasis Web (Studi Kasus Di Universitas Sriwijaya)," Kntia, vol. 4, pp. 31–36, 2017.

[6] D. Setiawan, N. Rafianto, T. D. S, I. B. K, and A. Setianto, "Implementasi Scrum & Agile Pada Pengerjaan Sistem Informasi Lentera," Semin. Nas. Teknol. Inf. dan Komun., pp. 61–75, 2021.

[7] I. Kurniawan and R. R. Sani, "Pemodelan SCRUM dalam Pengembangan Sistem Informasi Kesehatan pada Klinik Ar-Rokhim Sragen Kabupaten Sragen SCRUM Model on Development of Health Information System at Ar-Rokhim Clinic in Sragen Regency," J. Inf. Syst., vol. 4, no. 1, pp. 76–86, 2019.

[8] W. Adi Prabowo and C. Wiguna, "Sistem Informasi UMKM Bengkel Berbasis Web Menggunakan Metode SCRUM," J. Media Inform. Budidarma, vol. 5, no. 1, pp. 149–156, 2021, doi: 10.30865/mib.v5i1.2604.

[9] Lia Farokhah, Fadhli Almu'iini Ahda, and Lukman Hakim, "Implementasi SCRUM dalam Perancangan Aplikasi Emergency Button PMI Kota Malang," Digit. Zo. J. Teknol. Inf. dan Komun., vol. 11, no. 1, pp. 59–70, 2020, doi: 10.31849/digitalzone.v11i1.3869.

[10] S. Hadji, M. Taufik, and S. Mulyono, "Implementasi Metode Scrum Pada Pengembangan Aplikasi Delivery Order Berbasis Website (Studi Kasus Pada Rumah Makan Lombok Idjo Semarang)," Konf. Ilm. Mhs. Unissula 2, pp. 32–43, 2019.

[11] H. R. Suharno, N. Gunantara, and M. Sudarma, "Analisis Penerapan Metode Scrum Pada Sistem Informasi Manajemen Proyek Dalam Industri & Organisasi Digital," Maj. Ilm. Teknol. Elektro, vol. 19, no. 2, p. 203, 2020, doi: 10.24843/mite.2020.v19i02.p12.

[12] I. G. N. Suryantara, "Pengembangan Aplikasi Penjualan Mobil dengan Framework Scrum pada PT XYZ," Go-Integratif J. Tek. Sist. dan Ind., vol. 2, no. 2, pp. 70–85, 2021, doi: 10.35261/gijtsi.v2i2.5326.