

Pengembangan Aplikasi Sajiloka Menggunakan Metode Scrum

Rayhan Mahardhika Wijaya
Program Studi Informatika – Program Sarjana
Universitas Islam Indonesia
Yogyakarta, Indonesia
18523055@students.uii.ac.id

Andhik Budi Cahyono
Program Studi Informatika – Program Sarjana
Universitas Islam Indonesia
Yogyakarta, Indonesia
andhik.budi@uii.ac.id

Abstract— Memasak kini makin mudah dengan hadirnya *e-groceries*. Namun, masih terdapat beberapa permasalahan yang terjadi dalam aktivitas memasak. Berdasar survei awal yang dilakukan, terdapat tiga permasalahan utama yaitu: tidak adanya waktu dalam membeli bahan baku masakan, perasaan jenuh terhadap masakan yang selalu sama, dan adanya kegagalan dalam mencoba resep yang baru dikarenakan kurangnya pengetahuan dalam cara memasak yang benar. Sajiloka hadir dengan mengadopsi model bisnis *e-groceries* yang praktis dengan perpaduan produk paket bahasan makanan siap masak dan konten tutorial memasak yang lengkap dengan resep. Pengembangan Sajiloka dilakukan dengan metode *scrum* yang cepat dan adaptif. Terdapat tiga tahapan yang dilakukan dalam pengembangannya, yakni inisiasi proyek, eksekusi proyek, dan evaluasi proyek. Eksekusi proyek dilakukan dengan menggunakan *sprint*. Sedangkan, pada evaluasinya menggunakan *scrum metrics* berupa *velocity* dan *work capacity*. Selain itu terdapat pengujian perangkat lunak secara *white box* menggunakan *unit testing*. Hasil akhir dari evaluasi didapatkan bahwa beban pekerjaan (*velocity*) yang diberikan kepada pengembang melebihi kapasitas yang tersedia (*work capacity*) yakni 20 dari 15 kapasitas maksimal *sprint points*, sedangkan pada pengujian nilai keseluruhan *pass* dengan kata lain sistem bekerja dengan baik.

Keywords— *E-groceries*, *Scrum*, *Unit Testing*

I. PENDAHULUAN

Kemunculan *e-groceries* bertujuan untuk mempermudah masyarakat dalam mendapatkan bahan olahan pangan secara daring. Kemunculannya menjadi salah satu tempat untuk mendapatkan bahan olahan pangan selain pasar. Dilansir oleh [1], berdasarkan survei yang dilakukan oleh Redseer, sebuah perusahaan konsultan manajemen per Agustus 2020 menemukan adanya peningkatan penggunaan *e-groceries*, yang mana 60% dari responden yang berasal dari Indonesia mengungkapkan untuk terus menggunakan dan membeli bahan masak secara daring di masa mendatang.

Kemudahan dalam mendapatkan bahan masak merupakan tujuan dari adanya *e-groceries*. Namun, dengan kemudahan yang ditawarkan oleh adanya *e-groceries*, masih terdapat permasalahan yang dihadapi oleh masyarakat. Tim Sajiloka melakukan survei mengenai permasalahan yang masih terjadi dalam aktivitas memasak.

Didapatkan tiga permasalahan utama dari survei yang dilakukan, antara lain tidak ada waktu dalam membeli bahan baku masakan, perasaan jenuh terhadap masakan yang selalu sama, dan adanya kegagalan dalam mencoba resep yang baru dikarenakan kurangnya pengetahuan dalam cara memasak yang benar. Sajiloka bertujuan untuk memberikan solusi pada permasalahan yang ada, yang kemudian mengadopsi model

bisnis *e-groceries* yang praktis dengan perpaduan produk paket bahan masakan siap masak serta konten tutorial memasak yang lengkap.

Pengembangan Sajiloka dilakukan dengan metode *Scrum* yang cepat dan adaptif. Metode ini dipilih karena adanya temuan bahwa *Scrum* adalah metode yang ringan dan mendukung proses pengembangan yang cepat, serta dapat menerima perubahan dengan mudah, atau bersifat adaptif [2].

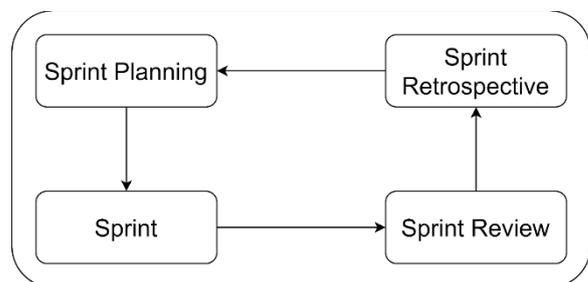
Penggunaan metode *Scrum* pada pengembangan Sajiloka dipilih karena metode tersebut merupakan metode yang familiar dan kerap digunakan pada pengembangan aplikasi, selain metode *Waterfall*, dan *Rapid Application Development*.

Selain hal tersebut, metode *Scrum* dipilih karena sifat pengembangan yang iteratif sangat membantu dalam pengembangan aplikasi Sajiloka sehingga dapat diberikan kepada calon pengguna dengan cepat.

II. LANDASAN TEORI

A. *Scrum*

Menurut Schwaber dan Sutherland [2] *Scrum* merupakan metode pengembangan yang *framework*-nya merupakan bagian dari metodologi Agile. *Scrum* memiliki sifat yang ringan serta adaptif. Sifat ringan yang dimiliki oleh *Scrum* mampu diimplementasikan ke dalam berbagai kompleksitas proyek. *Scrum* secara singkat merupakan sebuah *framework* yang membantu tim untuk mencapai tujuan dalam pengerjaan proyek yang kompleks melalui pendekatan iteratif dengan menghasilkan solusi adaptif.



Gambar 1 Alur Scrum

Gambar 1 menunjukkan adanya empat tahapan yang terjadi dalam sebuah *Scrum*, yakni *Sprint Planning*, *Sprint*, *Sprint Review*, dan *Sprint Retrospective*. Berikut penjelasan mengenai alur *Scrum*:

1. *Sprint Planning* merupakan tahap perencanaan kegiatan yang menginisiasi *Sprint*.

2. *Sprint* adalah kegiatan inti utama, di mana ide dirubah menjadi sebuah solusi yang memiliki *value*. Durasi setiap *Sprint* dilakukan setiap dua hingga empat minggu.
3. *Sprint Review* merupakan tahapan untuk melakukan inspeksi terhadap hasil dari *Sprint* yang dilakukan dan menentukan adaptasi yang akan dilakukan.
4. *Sprint Restropective* bertujuan untuk meningkatkan kualitas dan efektifitas dari kegiatan *Sprint* selanjutnya.

B. E-groceries

E-groceries adalah kemudahan yang ditawarkan dalam berbelanja dengan dapat dilakukan secara daring. *E-groceries* mencakup pengiriman makanan dan bahan makanan ke rumah, termasuk pengiriman makanan ke rumah dari restoran (*take away*), pengiriman dari warung, dan pengiriman produk pertanian [3].

C. Pengujian Perangkat Lunak

Pengujian perangkat lunak merupakan sebuah cara yang dilakukan untuk memverifikasi dan memvalidasi suatu kualitas dari sebuah perangkat lunak. Pada pengujian perangkat lunak tradisional, teknik yang digunakan dalam pengujian dibagi menjadi dua yaitu pengujian *Blackbox* dan *Whitebox*. Pengujian *Blackbox* digunakan untuk validasi sedangkan pengujian *Whitebox* digunakan untuk verifikasi [4].

D. Kajian Pustaka

Penggunaan metode serupa telah digunakan oleh [5] dalam pengembangan produk digital *startup*, yang mana menyatakan bahwa penggunaan metode *Scrum* telah membantu mereka mengatur proses pengembangan produk. Kemudian, melalui kegiatan pengembangan yang dilakukan oleh [6], penggunaan metode *Scrum* dalam proses pengembangan aplikasi memberikan fleksibilitas ketika terdapat perubahan kebutuhan pengguna secara tiba-tiba, sehingga revisi yang dilakukan bisa lebih cepat dan tidak mengganggu berjalannya *Sprint*.

Berdasarkan kegiatan pengembangan yang telah dilakukan oleh [7], metode *Scrum* mampu mendukung dokumentasi proyek yang jelas sehingga memudahkan dalam proses monitoring dari berjalannya proyek. Kemudian, paparan kegiatan pengembangan yang dilakukan oleh [8] dengan menggunakan metode *Scrum* sangat membantu dalam kondisi tim yang terbatas, dengan berbekal koordinasi yang telah dicantumkan pada panduan *Scrum*.

Penggunaan metode *Scrum* pada penelitian ini disertai dengan *Scrum Metric* sebagai evaluasi. *Scrum Metric* yang digunakan adalah matriks *work capacity* dan matriks *velocity*. Penggunaan dua matriks tersebut bertujuan untuk mengetahui beban kerja yang didapatkan terhadap kapasitas yang tersedia, sehingga *Scrum Master* mampu mendistribusikan beban kerja secara merata.

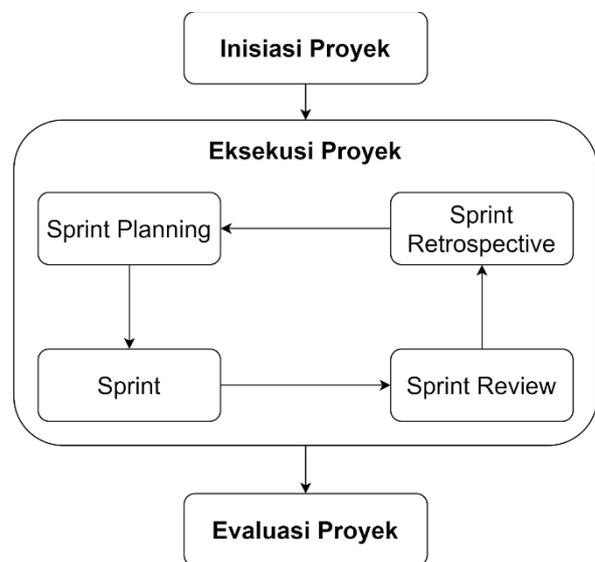
III. METODOLOGI

Alur pengembangan yang digunakan dalam pengembangan Sajiloka diilustrasikan pada Gambar 2, yang menggunakan metodologi *Scrum* pada tahapan eksekusi

proyeknya. Pengembangan mulanya dilakukan dengan inisiasi proyek Sajiloka, kemudian dilanjutkan ke dalam tahap eksekusi yang menggunakan *framework Scrum*, dan diakhiri dengan evaluasi terhadap proyek Sajiloka.

Seluruh pengembangan yang berlangsung pada Sajiloka dikerjakan oleh dua orang yang berperan sebagai *hacker*. Satu orang *hacker* bertanggung jawab dalam pengelolaan *Scrum* serta pengembangan *server backend* dan Web. Kemudian, satu orang *hacker* lainnya bertanggung jawab dalam mengembangkan tampilan aplikasi *platform mobile*.

Product backlog merupakan daftar pekerjaan yang bertujuan untuk meningkatkan kualitas produk. *Product owner* akan melakukan pembahasan terkait *product backlog* yang akan memecah *product backlog* menjadi *items* yang lebih kecil dan detail. Jika *product backlog* dianggap sudah siap, maka akan dilanjutkan ke tahap *sprint planning*.



Gambar 2 Tahap Pengembangan Sajiloka

Tahapan inisiasi proyek merupakan tahapan awal yang bertujuan untuk menyiapkan segala kebutuhan kegiatan pengembangan. Tahapan ini terdiri dari pengumpulan kebutuhan pengguna, desain kasar tampilan dari kebutuhan pengguna, serta media dan *tools* pengembangan yang akan digunakan. *Tools* yang kemudian digunakan selama tahap inisiasi proyek berlangsung adalah ClickUp, yang berguna untuk membantu dalam terlaksananya manajemen proyek.

Tahap eksekusi proyek dilaksanakan menggunakan metode *scrum* yang terdiri dari empat tahapan dan dilakukan selama tiga kali, yang kemudian menghasilkan iterasi pertama, iterasi kedua, dan iterasi ketiga. Semua tahapan *scrum* dilakukan pada setiap iterasi.

Tahap Evaluasi dilakukan setelah tahapan eksekusi proyek menggunakan *Scrum Metrics* dan pengujian aplikasi. Matriks tersebut bertujuan untuk meningkatkan efisiensi dan efektifitas. Menurut (Scott Downey, 2013) [9] terdapat beberapa matriks yang berfungsi untuk mengukur *scrum*. Matriks yang digunakan dalam tahap evaluasi proyek ini adalah *velocity*, dan *work capacity*.

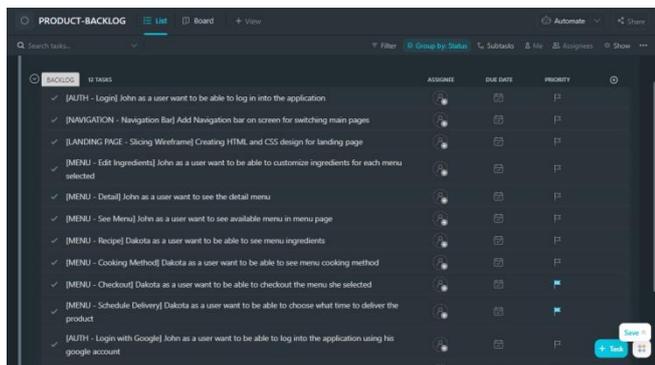
- Velocity* merupakan jumlah dari estimasi waktu semua pekerjaan yang diterima.
- Work Capacity* merupakan jumlah seluruh pekerjaan yang dilaporkan selama Sprint, menjelaskan pekerjaan yang selesai dilakukan atau tidak selesai.

IV. PEMBAHASAN DAN HASIL

Pada Bab ini akan dibahas secara jelas, hasil dari setiap tahapan yang telah diuraikan pada Gambar 2 sebelumnya, yakni inisiasi proyek, eksekusi proyek, dan evaluasi proyek.

a. Inisiasi Proyek

Pada tahapan inisiasi proyek dilakukan dengan menyusun rencana implementasi tahap awal. Rencana implementasi tahap awal dilakukan dengan menyusun seluruh rencana pengembangan yang akan dilakukan ke dalam ClickUp seperti yang tertera pada Gambar 3. Rencana-rencana tersebut disusun ke dalam *product backlog* yang kemudian menjadi bahan pengerjaan produk yang dilakukan pada setiap *sprint*.



Gambar 3 Product Backlog Sajiloka

b. Eksekusi Proyek

Tahapan eksekusi proyek akan menjelaskan proses pengembangan yang telah diinisiasi sebelumnya. Pengembangan dilakukan sebanyak tiga kali *sprint* atau tiga iterasi. Hal-hal yang terjadi setiap *sprint*-nya adalah *sprint planning*, *sprint*, *sprint review*, dan *sprint retrospective*.

1. Iterasi Sprint ke-1

• Sprint Planning

Pada tahap ini dilakukan diskusi untuk menentukan hal yang akan dikembangkan pada *sprint* iterasi ke-1. Tahap ini menghasilkan beberapa poin yaitu:

- Membuat halaman *login* dan daftar pengguna dari *mobile application* Sajiloka beserta dengan Web API.
- Membuat halaman utama dari *mobile application* Sajiloka beserta dengan Web API.
- Membuat halaman detail produk dari *mobile application* Sajiloka beserta dengan Web API.

- Membuat bagian bahan baku dan cara memasak pada halaman detail produk dari *mobile application* Sajiloka beserta dengan Web API.
- Membuat *landing page* untuk situs perkenalan produk Sajiloka.
- Melakukan *hosting* Web API menggunakan jasa Heroku.com untuk memudahkan pengembangan dan presentasi produk.

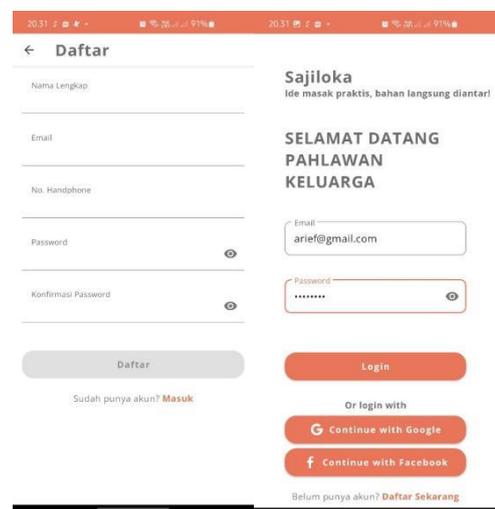
• Sprint

Melalui diskusi yang telah dilakukan pada tahap *Sprint Planning* dibentuklah beberapa dokumen yang akan digunakan dalam tahap pengembangan. Dokumen tersebut diantaranya adalah *diagram flowchart* dari setiap halaman, rancangan desain *user interface* yang telah dibuat oleh hipster sebagai acuan pengembangan aplikasi *front-end*, serta desain tabel basis data untuk Web API selaku *back-end*.

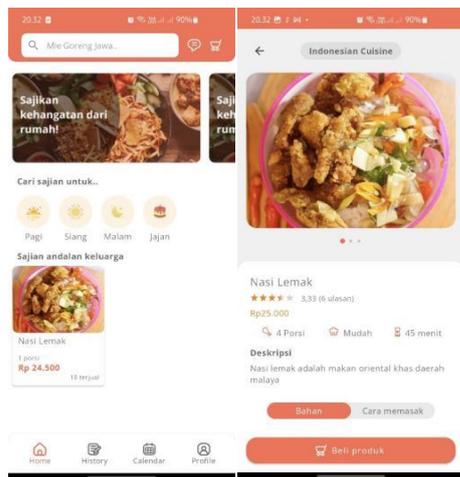
Pada tahap *sprint* iterasi ke-1 ini fitur-fitur yang mendapat prioritas utama untuk dikembangkan antara lain adalah autentikasi pengguna (*login*, *register*, dan *auth check*), halaman utama (beranda), halaman detail produk, dan halaman arahan (*landing page*).

• Sprint Review

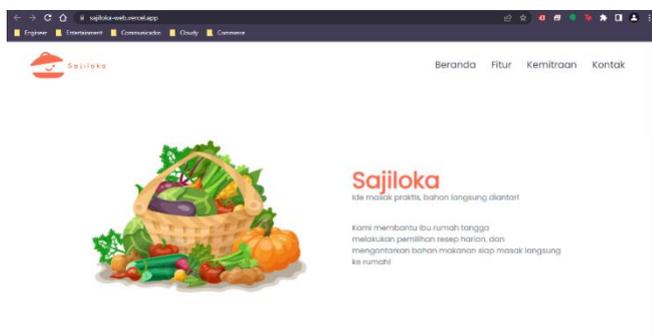
Fitur-fitur yang telah selesai dikembangkan pada iterasi *sprint* pertama ini antara lain adalah autentikasi pengguna yaitu *login* dan *register* pada Gambar 4, halaman utama (beranda) serta halaman detail produk pada Gambar 5, dan halaman arahan (*landing page*) Gambar 6.



Gambar 4 Halaman Pendaftaran dan Login Pengguna



Gambar 5 Halaman Utama dan detail produk



Gambar 6 Halaman Landing Page

- *Sprint Retrospective*

Setelah semua kegiatan *sprint* selesai dan hasil sudah dinyatakan selesai pada iterasi pertama. Kemudian, tim melakukan sebuah diskusi refleksi dari kegiatan *sprint* yang telah berlangsung. Hasil dari diskusi ditampilkan sebagai berikut:

1. Performa yang diberikan pada saat pengerjaan *task* sangat baik karena koordinasi dari sisi pengembang dan desainer cukup konsisten sehingga hasil dari desain sesuai dengan yang diharapkan.
2. Hasil dari fitur aplikasi yang dikembangkan cukup sesuai dengan desain yang diberikan oleh desainer sehingga aplikasi sudah dapat berfungsi.

2. *Iterasi Sprint ke-2*

- *Sprint Planning*

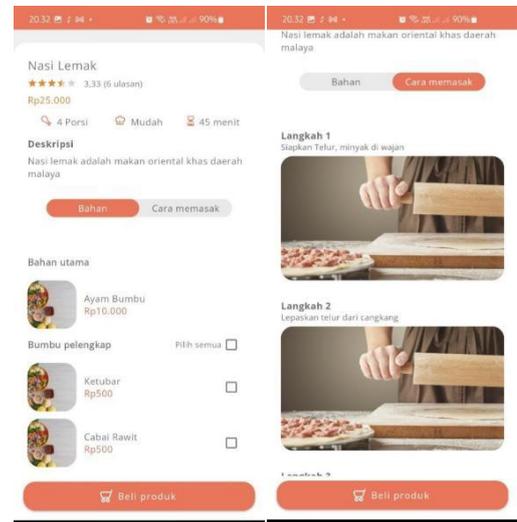
Pada tahap ini dilakukan diskusi untuk menentukan hal yang akan dikembangkan pada *sprint* iterasi kedua. Tahap ini menghasilkan beberapa poin yaitu pengembangan dalam bagian bahan baku memasak dan pada bagian cara memasak.

- *Sprint*

Pada tahap *sprint* iterasi kedua ini fitur-fitur yang mendapat prioritas utama untuk dikembangkan antara lain adalah kedua bagian bahan baku dan cara memasak.

- *Sprint Review*

Fitur-fitur yang telah selesai dikembangkan pada iterasi *sprint* kedua ini antara lain adalah bagian bahan baku dan cara memasak pada Gambar 7.



Gambar 7 Halaman Bahan Baku dan cara pengolahannya

- *Sprint Retrospective*

Pada akhir dari iterasi Scrum ke-2 ini tim melakukan diskusi *sprint retrospective* yang membahas terkait hasil kinerja dari kegiatan *sprint* kedua ini. Hasil dari diskusi ditampilkan sebagai berikut:

1. Performa pengerjaan dinilai stabil karena terdapat perkembangan fitur dari *sprint* sebelumnya
2. Hasil dari tampilan aplikasi Sajiloka cukup sesuai ekspektasi karena sisi pengembang dan desainer cukup siap dalam melaksanakan *task* dari *sprint*.

3. *Iterasi Sprint ke-3*

- *Sprint Planning*

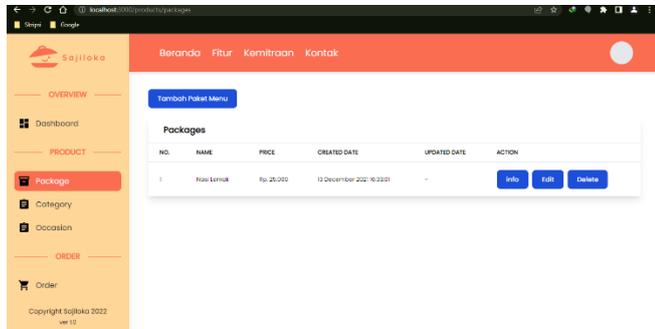
Pada tahap ini dilakukan diskusi untuk menentukan hal yang akan dikembangkan pada *sprint* iterasi ketiga. Tahap ini menghasilkan beberapa poin yaitu halaman lihat daftar produk, halaman lihat detail produk, halaman tambah produk, dan halaman ubah produk.

- *Sprint*

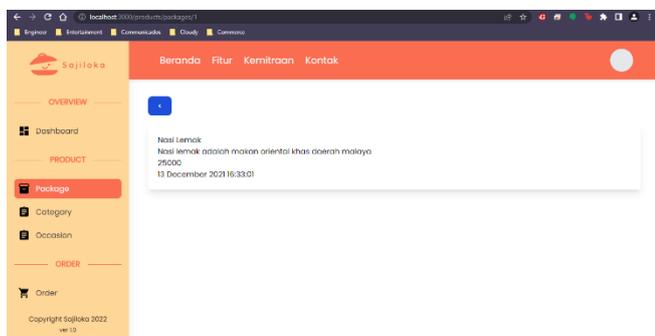
Pada tahap *sprint* iterasi ketiga ini fitur-fitur yang mendapat prioritas utama untuk dikembangkan antara lain halaman lihat daftar produk, halaman lihat detail produk, halaman tambah produk, dan halaman ubah produk.

- *Sprint Review*

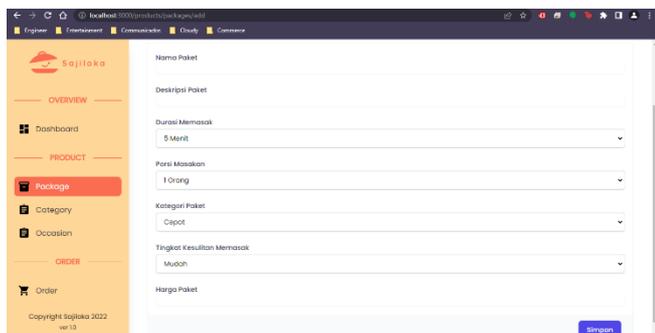
Fitur-fitur yang telah selesai dikembangkan pada iterasi *sprint* ketiga ini antara lain adalah halaman lihat daftar produk Gambar 8, halaman lihat detail produk Gambar 9, dan halaman tambah produk Gambar 10. Sedangkan terdapat *task* yang tidak dapat diselesaikan yaitu halaman ubah produk dikarenakan keterbatasan waktu dan tenaga di mana hanya terdapat 1 (satu) pengembang yang mengerjakan semua *tasks*.



Gambar 8 Halaman Daftar Produk



Gambar 9 Halaman Detail Produk



Gambar 10 Halaman Tambah Produk

- *Sprint Retrospective*

Pada akhir dari iterasi Scrum ketiga ini tim melakukan diskusi *sprint retrospective* yang membahas terkait hasil kinerja dari kegiatan *sprint* ketiga ini. Hasil dari diskusi ditampilkan sebagai berikut:

1. Performa pengerjaan cukup menurun dikarenakan hanya terdapat 1 (satu) pengembang yang mengerjakan aplikasi admin baik dari sisi *front-end* maupun *back-end*.

2. Hasil dari tampilan aplikasi admin dikerjakan dengan seadanya karena berfokus kepada fungsionalitas sistem serta dikarenakan pada awal inisiasi proyek desainer tidak membuat desain tampilan untuk aplikasi sisi admin.

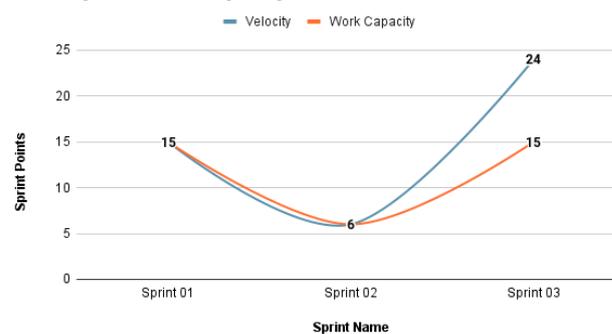
c. *Evaluasi Proyek*

Terdapat beberapa evaluasi yang dilakukan setelah tahapan eksekusi usai, yakni hasil dari *scrum metric*, dan hasil dari pengujian aplikasi.

1. *Scrum Metric*

Penggunaan matiks *velocity* dan *work capacity* dapat dilihat pada Gambar 11. *Work capacity* turun di bawah *velocity*, yang sebelumnya berada pada angka yang sama. Hal ini disebabkan karena *velocity* dihitung berdasarkan estimasi *points* yang dikerjakan secara asli, sedangkan *work capacity* dihitung berdasarkan laporan jumlah *points* yang selesai dikerjakan. Hal ini membuktikan, bahwa Tim Sajiloka telah memberikan beban pekerjaan kepada pengembang lebih dari kapasitas maksimal yang diberikan.

Velocity vs. Work Capacity Trend



Gambar 11 Scrum Metric Trend

2. *Pengujian Aplikasi*

Pengujian aplikasi dilakukan secara *whitebox* terhadap setiap unit sistem pada sisi *backend* dengan menggunakan *unit testing* Go melalui *library testify*. Hasil *unit testing* dilakukan pada setiap fungsi yang ada pada Web API Sajiloka. Pengujian dilakukan menggunakan menggunakan *consumer API* yang dimiliki oleh sisi pengguna dan *admin API* yang dimiliki oleh sisi pengelola.

- *Consumer API*

Pada pengujian ini, sisi pengguna dibagi menjadi beberapa kelas *service* berdasarkan domainnya, yaitu *auth*, *menu*, *ingredient*, *category*, *occasion*, dan *difficulty*. Pada pengujian yang dilakukan berdasarkan Tabel 1, semua *service* memberikan nilai *pass*. Nilai *pass* pada Tabel 1, dimaksudkan bahwa unit-unit yang dilakukan *testing* bekerja dengan baik.

Tabel 1 Hasil Unit Testing Consumer API

Service	Unit	Result	Waktu (s)
Auth	<i>generateToken</i>	Pass	0.4s
	<i>validateToken</i>	Pass	0.4s
Menu	<i>getAllMenu</i>	Pass	1.3s
	<i>getMenuById</i>	Pass	1.3s
	<i>getProcessingStepsById</i>	Pass	1.3s
Ingredients	<i>getMenuIngredients</i>	Pass	1.3s
Category	<i>getAllCategories</i>	Pass	1.4s
	<i>getAllByMenuId</i>	Pass	1.2s
Occasion	<i>getAllOccasions</i>	Pass	1.3s
	<i>getAllByMenuId</i>	Pass	1.2s
Difficulty	<i>getAllDifficulties</i>	Pass	1.3s
	<i>getById</i>	Pass	1.3s

- Admin API

Pada pengujian unit API sisi pengelola dilakukan pada *service Menu*. Pada pengujian yang dilakukan berdasarkan Tabel 2, *service* pengujian memberikan nilai *pass*, yang membuktikan bahwa *service* tersebut bekerja dengan baik.

Tabel 2 Hasil Unit Testing Admin API

Service	Unit	Result	Waktu (s)
Menu	<i>CreateMenu</i>	Pass	1.3s

V. KESIMPULAN

Berdasarkan hasil dari evaluasi yang dilakukan menggunakan *Scrum Metric* dan pengujian aplikasi secara *whitebox* menggunakan *Unit Testing*. Dapat disimpulkan bahwa penggunaan metode *Scrum* mendukung proses pengembangan yang cepat, yakni dengan adanya tiga iterasi *sprint* yang digunakan.

Namun, hasil yang didapat dari evaluasi proyek yang telah dilakukan terhadap matriks *velocity* dan *work capacity* pada Gambar 11 menunjukkan data dimana *work capacity* nilainya lebih rendah dibandingkan dengan *velocity* hal ini disebabkan karena beban pekerjaan melebihi *work capacity* maksimal yang tersedia. Hal tersebut merupakan akibat dari tidak dilakukannya kegiatan *Daily Scrum* atau *Daily Standup* yang memiliki tujuan utama sebagai kegiatan *monitoring* terhadap proses berjalannya *Scrum*. Kegiatan *monitoring* tersebut dapat memberikan penyesuaian pada kegiatan *Sprint* yang berlangsung sehingga *velocity* dan *work capacity* dapat seimbang.

Tidak dilakukannya kegiatan *Daily Scrum* merupakan akibat dari kurangnya personel yang terlibat dalam *Scrum*, yang mana seorang *hacker* memiliki peran ganda. Hal ini kemudian menyebabkan *hacker* tidak dapat fokus karena banyaknya tanggung jawab yang dipikul yaitu sebagai *Scrum Master*, *Backend Developer*, dan *Web Developer*.

Metode *Scrum* yang digunakan pada Sajiloka cukup memberikan hasil yang sesuai dengan ekspektasi yaitu mampu membawa tim menghasilkan sebuah *deliverables* yang cepat serta efisien.

REFERENSI

- [1] Jakarta Post, "Online grocery shopping to drive e-commerce growth as PSBB reimposed: Experts - Business - The Jakarta Post." 2020.
- [2] K. Schwaber and J. Sutherland, "The Scrum Guide," no. November, 2020.
- [3] A. Bjørgen, K. Y. Bjerkan, and O. A. Hjelkrem, "E-groceries: Sustainable last mile distribution in city planning," *Res. Transp. Econ.*, vol. 87, no. June, p. 100805, 2021, doi: 10.1016/j.retrec.2019.100805.
- [4] S. Nidhra and J. Dondeti, "BLACK BOX AND WHITE BOX TESTING TECHNIQUES –A LITERATURE REVIEW," *Proj. Manag. J.*, vol. 2, no. 2, pp. 29–50, 2012.
- [5] N. T. Rachman and D. Sushandoyo, "Analysis of scrum implementation in digital startup product," *COSTINGJournal Econ. Bus. Account.*, vol. 5, no. 1, p. 7, 2021.
- [6] 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.
- [7] B. Kharisma and N. Santoso, "Pengembangan Aplikasi Manajemen Proyek Perangkat Lunak Kolaboratif Menggunakan Scrum," *J. Pengemb. Teknol. Inf. dan Ilmu Komput. Univ. Brawijaya*, vol. 4, no. 3, pp. 723–732, 2020.
- [8] A. Andipradana and K. Dwi Hartomo, "Rancang Bangun Aplikasi Penjualan Online Berbasis Web Menggunakan Metode Scrum," *J. Algoritma*, vol. 18, no. 1, pp. 161–172, 2021, doi: 10.33364/algoritma/v.18-1.869.
- [9] S. Downey and J. Sutherland, "Scrum metrics for Hyperproductive Teams: How they fly like fighter aircraft," *Proc. Annu. Hawaii Int. Conf. Syst. Sci.*, pp. 4870–4878, 2013, doi: 10.1109/HICSS.2013.471.