

Aplikasi *Frontend* untuk Meningkatkan *Customer Experience* dan Mendapatkan *Data Customer* Menggunakan Metode Scrum pada Studi Kasus Livera

Farrel Anthony*) dan Sigit Birowo

Program Studi Teknik Informasi, Institut Bisnis dan Informatika Kwik Kian Gie, Jl. Yos Sudarso Kav 87, Sunter Jakarta 14350, Indonesia.

*) Surel korespondensi : farrelanth@gmail.com

Abstract. *The development of digital technology has had a significant impact on human life, including institutions, organizations, and business actors. Rapid technological advancement has become a crucial factor for businesses to remain competitive in an increasingly dynamic market. Livera is a company engaged in the health beverage industry, with a focus on weight loss products. Currently, Livera does not yet have an application to fulfill customer satisfaction after purchase. Therefore, this study aims to design a website-based application intended to improve customer satisfaction, where customers can consume products according to a predetermined schedule, and the website application can also facilitate administrators in inputting and automatically accessing data. Customer experience represents an evaluation of feelings that can be identified through the experiences perceived by customers. Customer experience can be one of the most important determining factors for customers. This research is conducted using the Scrum method. The Scrum activities include product backlog, sprint backlog, daily Scrum, sprint review, and sprint retrospective. The roles in Scrum consist of the product owner, Scrum master, and the development team. This study employs a qualitative research method and the Scrum framework for system development. Data collection techniques used include non-participant observation, unstructured interviews, and literature studies from books and academic journals that serve as references related to customer experience, website design, and the Scrum method.*

Keywords: *website application, scrum, customer experience*



This work is licensed under Attribution-NonCommercial-ShareAlike 4.0 International. To view a copy of this license, visit <http://creativecommons.org/licenses/by-nc-sa/4.0/>

Diterbitkan oleh LPPM Institut Bisnis dan Informatika Kwik Kian Gie, Jl. Yos Sudarso Kav 87, Sunter Jakarta 14350, Indonesia.
DOI : <https://doi.org/10.46806/jib.v14i2.1881>

1. Pendahuluan

Berhubung dengan semakin banyak dan semakin ketatnya persaingan di dunia usaha, perusahaan dituntut untuk menciptakan inovasi agar dapat memenuhi kepuasan pelanggan. Oleh karena itu, agar dapat memenuhi semua kebutuhan yang dibutuhkan oleh perusahaan, peneliti diminta untuk membuat aplikasi dengan menggunakan metode *scrum* berbasis situs web.

Metode Scrum merupakan salah satu kerangka kerja pengembangan sistem yang mengadopsi pendekatan *agile*, yang menekankan fleksibilitas, kolaborasi tim, serta kemampuan beradaptasi terhadap perubahan kebutuhan pengguna. Pendekatan *agile* dipilih karena mampu meningkatkan produktivitas pengembangan, memungkinkan setiap anggota tim untuk mengerjakan tugas secara paralel tanpa harus menunggu proses lain selesai, serta memberikan ruang bagi pengembang untuk terus melakukan

perbaikan dan pengembangan secara berkelanjutan. Oleh karena itu, penelitian ini berfokus pada perancangan dan pengembangan aplikasi berbasis situs web dengan menggunakan metode Scrum.

Aplikasi yang dikembangkan dirancang untuk meningkatkan kepuasan pelanggan melalui penyediaan fitur-fitur pendukung *customer experience*, seperti fitur penelusuran kembali produk yang telah dibeli serta fitur pengingat berupa notifikasi agar pelanggan dapat mengonsumsi produk sesuai dengan waktu yang telah ditentukan. Selain itu, aplikasi ini juga dirancang untuk mengumpulkan dan menampilkan data pelanggan secara otomatis sehingga dapat dimanfaatkan oleh pihak admin perusahaan dalam menyusun strategi layanan, termasuk pemberian penawaran terbaik kepada pelanggan. Dalam pengembangan aplikasi ini, peneliti berperan sebagai *frontend developer* dengan menggunakan bahasa pemrograman HyperText Markup Language (HTML), Cascading Style Sheet (CSS), serta kerangka kerja React.js.

Berdasarkan kondisi yang ada, permasalahan yang dihadapi antara lain belum optimalnya *customer experience* yang dirasakan pelanggan setelah melakukan pembelian, belum tersedianya fitur penelusuran kembali produk yang telah dibeli, belum adanya notifikasi pengingat konsumsi produk, serta belum tersedianya sistem yang memungkinkan admin melihat data detail pelanggan secara otomatis. Oleh karena itu, penelitian ini dibatasi pada pengembangan fitur penelusuran riwayat pembelian produk, notifikasi pengingat konsumsi produk bagi pelanggan, serta penyajian data pelanggan secara otomatis bagi admin. Adapun tujuan dari penelitian ini adalah untuk meningkatkan kepuasan pelanggan dan kualitas layanan perusahaan melalui pengembangan aplikasi berbasis web yang mampu mendukung pengelolaan data pelanggan, penyediaan fitur pengingat, serta peningkatan pengalaman pelanggan secara keseluruhan.

2. Tinjauan Pustaka

2.1. Data

Data adalah fakta yang diketahui, dapat direkam dan memiliki makna tersirat. Misalnya, nama, nomor telepon, dan alamat orang yang dikenal.

Saat ini, data juga biasanya dapat disimpan di ponsel, yang memiliki perangkat lunak basis data sederhana mereka sendiri. Data ini juga dapat direkam dalam buku alamat yang diindeks atau disimpan di perangkat keras, menggunakan komputer pribadi dan perangkat lunak seperti *Microsoft Access* atau *Excel*. Kumpulan data terkait dengan makna implisit ini adalah *database*.

2.2. Customer Experience

Menurut Phil Klaus (2017:17) Definisi dari *Customer Experience* sangat luas. Mulai dari pengalaman pembelian dan konsumsi aktual dan yang diantisipasi pelanggan, penawaran ekonomi yang khas atau hasil dari menghadapi, menjalani, atau hidup

melalui hal-hal, hingga gagasan tentang konsumen baru yang mencari pengalaman sebagai *co-creator* dari nilai dan pengalaman. Istilah "*co-creator*" menyoroti pengaruh pengalaman pelanggan pada strategi pemasaran berdasarkan pengalaman, seperti yang diinginkan dan dijalankan dalam konteks barang/jasa, pariwisata, perjalanan, dan perhotelan.

2.3. Metode Agile dan Scrum

Pendekatan dengan metode *Agile* untuk pengembangan perangkat lunak menganggap desain dan implementasi sebagai aktivitas utama dalam proses perangkat lunak. Mereka menggabungkan aktivitas lain, seperti elisitasi dan pengujian persyaratan, ke dalam desain dan implementasi.

Metode *Scrum* dikembangkan (Schwaber dan Beedle 2001; Rubin 2013) untuk menyediakan kerangka kerja untuk mengorganisir proyek tangkas dan, setidaknya sampai batas tertentu, memberikan visibilitas eksternal tentang apa yang sedang terjadi. Pengembang *Scrum* ingin menjelaskan bahwa *Scrum* bukanlah metode untuk manajemen proyek dalam pengertian konvensional, sehingga mereka dengan sengaja menciptakan terminologi baru, seperti *ScrumMaster*, yang menggantikan nama seperti manajer proyek.

2.4. Website

Situs web atau *website* adalah sebuah alat komunikasi yang Anda bangun untuk membantu mencapai tujuan yang telah Anda tetapkan. Namun, ini adalah alat komunikasi yang unik karena dapat melakukan berbagai tugas yang tidak pernah diharapkan oleh alat sebelumnya untuk melakukannya sendiri. Faktanya, sebuah situs web dapat menjadi hampir semua alat komunikasi yang pernah ada, dan beberapa lainnya. Situs web dapat berupa buku harian, brosur, buku, televisi, telepon, radio, peragaan slide, papan reklame, kartu nama, dan seterusnya.

2.5. Frontend

Aplikasi *frontend* adalah sesuatu yang terlihat oleh *end-user*. Secara umum, saat kita merujuk ke *frontend*, kita memikirkan aplikasi web yang berjalan di browser. Pekerjaan *frontend* dapat dibagi menjadi dua modul utama yaitu representasi dan logika. Representasi adalah apa yang dilihat pengguna, antarmuka, bagaimana elemen dirender, dan bagaimana berinteraksi dengannya. Logikanya adalah segala sesuatu yang membuatnya menjadi aplikasi, seperti mengambil data, mengubahnya untuk disajikan kepada pengguna, dan menangani permintaan, status, dan validasi *input* data.

3. Metode

Dalam penelitian ini, menggunakan metode kualitatif dalam pengembangan aplikasi menjadi pilihan yang ideal. Menurut peneliti, penelitian metode kualitatif sangat cocok dikarenakan dalam penelitian yang baik, perlu dilakukan pengamatan secara mendalam terhadap objek penelitian. Metode ini dapat membantu pengembang

aplikasi untuk memperoleh data dan wawasan tentang kinerja aplikasi. Hal ini memungkinkan pengembang untuk memperbaiki dan meningkatkan fungsionalitas aplikasi.

3.1. Metode Pengembangan Sistem

Dalam penelitian ini, peneliti menggunakan metode *scrum*, metode *scrum* merupakan metode pengembangan perangkat sistem responsif yang mampu memberikan nilai dan manfaat yang baik terhadap suatu perangkat lunak yang akan dikembangkan. Didalam *scrum* memiliki 3 role utama yaitu: *product owner*, *scrum master*, dan *development team*. *Scrum master* adalah orang yang bertanggung jawab untuk mengarahkan tim menuju arah dan tujuan yang benar. *Product Owner* merupakan seorang profesional yang memiliki pengetahuan yang mendalam mengenai produk yang akan dikembangkan. Sedangkan *scrum team* atau tim *development* adalah tim yang beranggotakan individu-individu yang mampu mengorganisasikan dirinya sendiri dan merupakan sebuah tim multifungsi daalam pengembangan suatu perangkat lunak. Metode pengembangan *Scrum* ini memiliki beberapa tahapan, yang peneliti jabarkan dalam tabel berikut:

Tabel 1. Tahapan Metode *Scrum* pada penelitian ini

No	Langkah	Penjelasan
1	<i>Backlog</i>	Tim <i>development</i> dan <i>scrum master</i> membuat daftar kebutuhan atau fitur produk yang harus dibuat.
2	<i>Sprint Planning</i>	Tim <i>Scrum</i> akan melakukan pertemuan untuk merencanakan <i>sprint</i> , melakukan <i>sprint goal</i> , dan memilih item-item dari <i>Product Backlog</i> yang akan dikerjakan selama <i>sprint</i>
3	<i>Sprint</i>	Tim <i>Scrum</i> akan mengerjakan item-item dari <i>Product Backlog</i> selama periode waktu tertentu yang disebut <i>sprint</i> .
4	<i>Scrum Meetings</i>	Aktivitas <i>scrum meetings</i> merupakan <i>meeting</i> rutin atau pertemuan rutin yang dilakukan setiap beberapa hari untuk mengevaluasi apa yang sedang dikerjakan, kendala yang ada, dan target penyelesaian tugas selanjutnya.
5	<i>Sprint Review/ Demo</i>	Tim <i>Scrum</i> meninjau hasil pekerjaan yang telah diselesaikan selama <i>sprint</i> . Memberikan peningkatan perangkat lunak kepada <i>scrum master</i> yang telah diimplementasikan dapat ditunjukkan dan dievaluasi oleh <i>scrum master</i> .
6	<i>Sprint Retrospective</i>	Pada setiap <i>sprint</i> yang berakhir, akan dilakukan <i>sprint retrospective</i> . Semua anggota tim diberikan kesempatan untuk menyampaikan pendapat dan evaluasi mengenai kinerja selama menerapkan <i>scrum</i> .

3.2. Teknik Pengumpulan Data

Pengumpulan data dilakukan untuk mengumpulkan sejumlah informasi yang bersumber dari informan yang diteliti agar peneliti dapat memberikan solusi yang sesuai dengan kebutuhan pengguna di kemudian hari. Dalam pengumpulan data, peneliti menggunakan beberapa metode dengan tujuan mendapatkan data yang valid dan akurat. Diantaranya adalah:

3.2.1 Observasi Non-Partisipan

Peneliti melakukan pengumpulan data melalui pengamatan secara tidak langsung pada Livera terkait dengan proses pembuatan aplikasi yang bertujuan untuk meningkatkan kepuasan pelanggan. Demi mengetahui objek penelitian saat ini, peneliti melakukan *Focus Group Discussion* dengan salah satu pendiri Livera.

3.2.2. Studi Pustaka

Peneliti juga menerapkan teknik studi pustaka. teknik ini dilakukan untuk memperoleh data, wawasan dan informasi tambahan berupa buku, jurnal dan penelitian terdahulu yang diperlukan dalam penelitian ini. Peneliti menggunakan teknik ini untuk mengumpulkan wawasan mengenai metode *Scrum*, *Customer Experience*, dan *JavaScript*.

3.2.3. Wawancara Tidak Terstruktur

Peneliti menggunakan teknik wawancara tidak terstruktur untuk mengumpulkan data dari pihak-pihak yang terkait. pada penelitian ini, peneliti melakukan wawancara kepada Bapak Victor Fernando yang menjabat sebagai *CTO* dan salah satu pendiri Livera dan sebagai *Scrum Master* pada proyek yang dirancang peneliti kali ini.

3.3. Teknik Analisis Data

Analisis data dilakukan setelah melakukan pengumpulan data, data yang diperoleh akan membantu peneliti dalam melakukan perancangan sistem yang tepat bagi para pengguna. Teknik analisis data yang sesuai dengan pendekatan penelitian ini adalah kualitatif. Data kualitatif digambarkan dengan kata-kata ataupun dengan kalimat. Pengolahan dari data kualitatif ini dilakukukan dengan menggunakan langkah-langkah sebagai berikut:

3.3.1. Reduksi Data

Reduksi data merupakan tahapan untuk mengetik kembali dalam bentuk uraian atau laporan terperinci, dirangkum, dipilih hal-hal yang pokok, diberi susunan yang lebih sistematis supaya peneliti lebih mudah untuk menarik kesimpulan.

3.3.2. Penyajian Data

Penyajian data dibatasi sebagai kumpulan informasi yang tersusun yang memberi kemungkinan adanya penarikan kesimpulan dan pengambailan tindakan. Dengan penyajian tersebut akan dapat dipahami apa yang harus dilakukan selanjutnya.

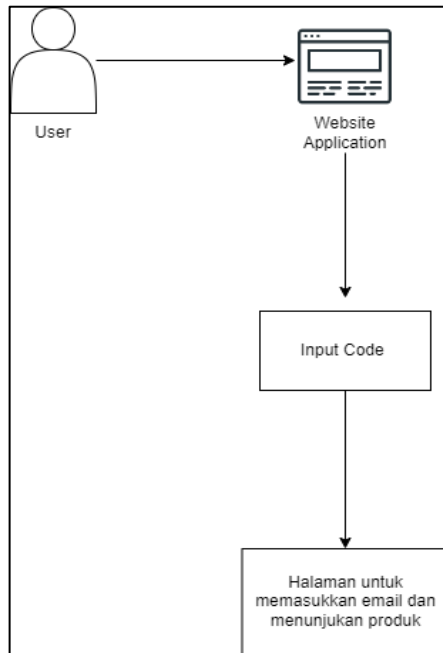
3.3.3. Penarikan Kesimpulan

Penarikan kesimpulan hanyalah sebagian dari suatu kegiatan konfigurasi yang utuh. Kesimpulan awal yang dikemukakan masih bersifat sementara, dan akan mengalami perubahan apabila adanya bukti-bukti yang kuat yang mendukung pada tahap pengumpulan data berikutnya. Dengan demikian kesimpulan dalam penelitian kualitatif mungkin dapat menjawab rumusan masalah yang dirumuskan sejak awal.

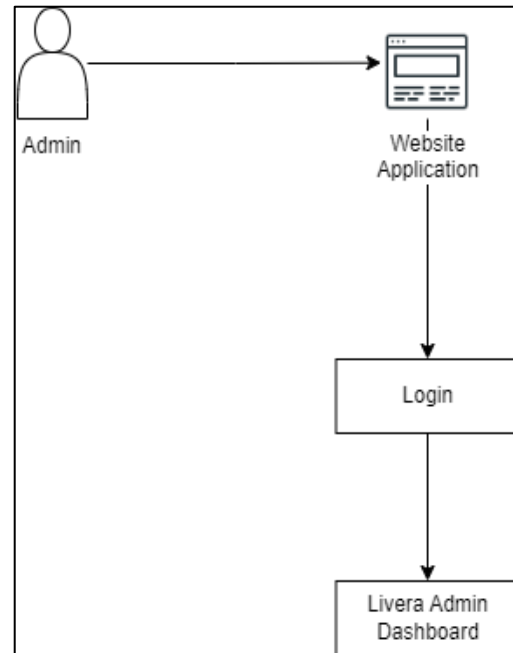
4. Hasil

4.1. Arsitektur Sistem

Dijelaskan bahwa arsitektur sistem yang dibuat pada aplikasi *Livera Experience* dapat diakses oleh siapa saja yang memiliki kode pembelian setelah pelanggan menerima produk. Aplikasi web ini dapat diakses oleh pengguna menggunakan bantuan dari *web hosting*.



Gambar 1. Rancangan Arsitektur Sistem Livera Experience



Gambar 2. Rancangan Arsitektur Sistem Livera Admin Dashboard

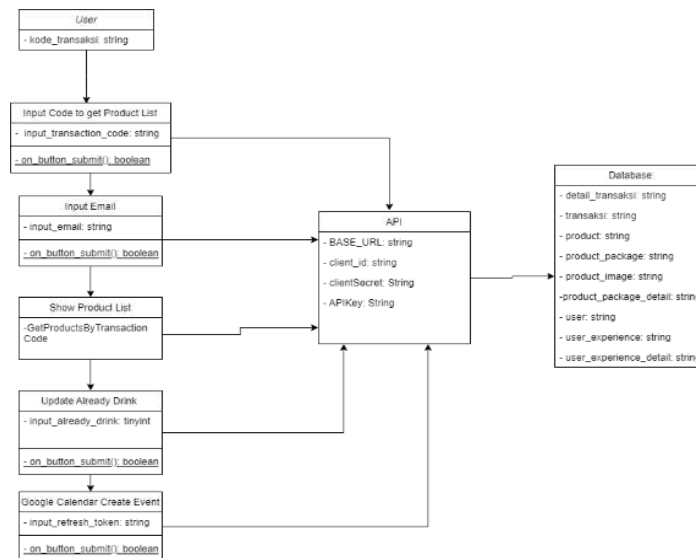
4.2. Arsitektur Sistem Livera Admin Dashboard

Aplikasi *website livera admin dashboard* ini dapat diakses oleh admin dengan memasukkan *Email* dan *password* lalu admin dapat mengakses website yang berisi data pengguna.

4.3. Class Diagram

4.3.1. Class Diagram Livera Experience

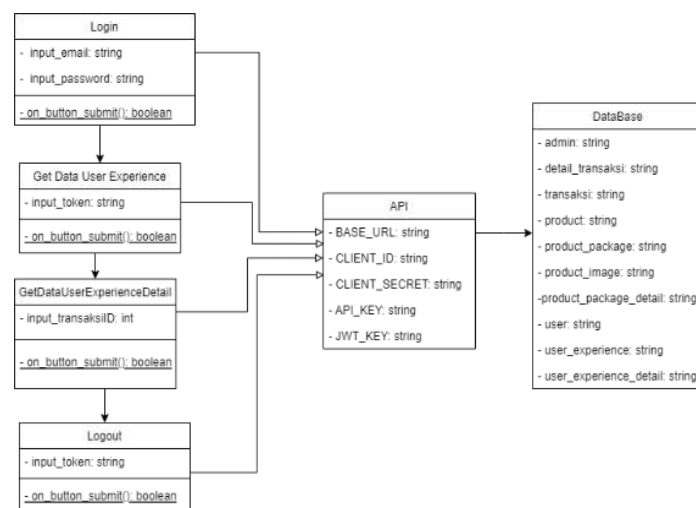
Class Diagram dari aplikasi *Livera Experience* ini digambarkan pada Gambar 3. Peneliti membagi *class diagram* ini menjadi 7 kelas yaitu: API, InputCodeToGetProductList, InputEmail, ShowProductList, UpdateAlreadyDrink, GoogleCalendarCreateEven, dan database. Setiap kelas membutuhkan masukkan yang berbeda-beda, pada InputCode ToGetProductList dibutuhkan *transaction_code*, pada InputEmail dibutuhkan *input_email*, pada UpdateAlreadyDrink dibutuhkan *input already_drink* dan pada GoogleCalendarCreateEvent dibutuhkan *input refresh_token*. Semua kelas yang melakukan *input* menggunakan modul *on_submit_button* untuk disambungkan ke API. Semua *input* yang sudah berhasil dimasukan melalui API akan dilakukan validasi dengan basis data.



Gambar 3. Class Diagram Sistem Livera Experience

4.3.2. Class Diagram Admin Dashboard

Class Diagram dari aplikasi untuk admin *dashboard* ini digambarkan pada Gambar 4. Dalam Class Diagram ini peneliti membuat 5 kelas yaitu: Database, API, Login, Logout, GetDataUserExperience dan GetDataUserExperienceDetail. Input yang dibutuhkan dari masing masing API berbeda, untuk Login dibutuhkan email dan password, untuk Logout Admin dibutuhkan token, untuk Get DataUserExperience membutuhkan sebuah *input* berupa token yang didapat setelah melakukan *login*, untuk GetDataUserExperienceDetail membutuhkan *input* transaksiID. Semua kelas memiliki modul untuk submit untuk menghubungkan *Application Programming Interface* (API) ke *database*.



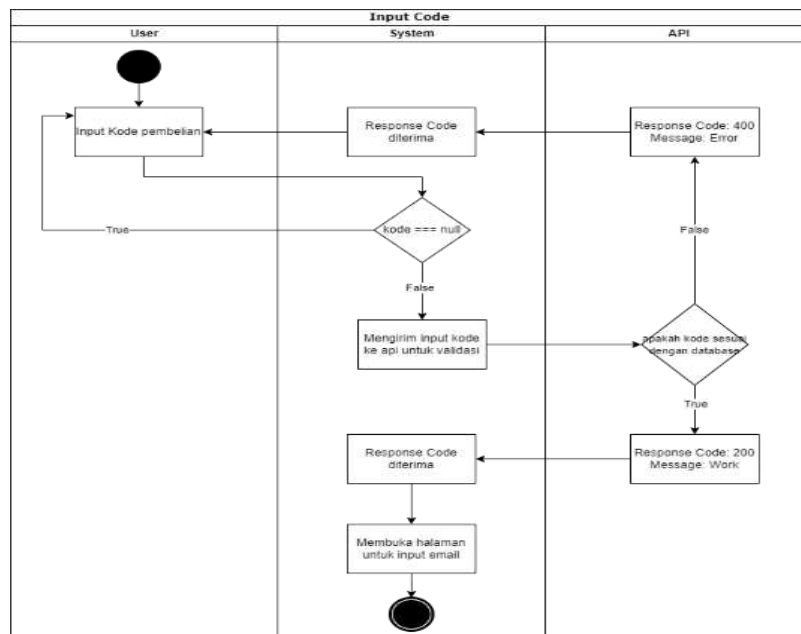
Gambar 4. Class Diagram Sistem Admin Dashboard

4.4. Activity Diagram

4.4.1. Proses Input Code

Berdasarkan Gambar 5, pertama *user* akan diminta untuk memasukkan kode transaksi. Jika sistem menerima kode sebagai data kosong maka halaman akan tetap berada di

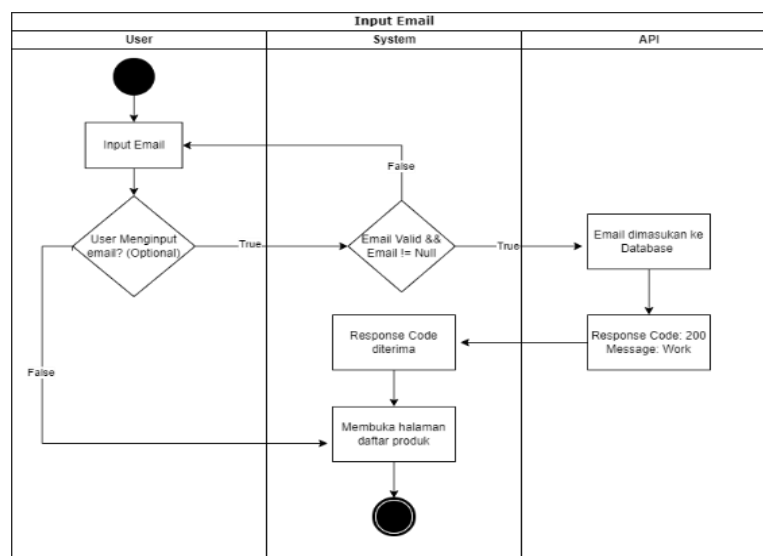
halaman *input* kode, jika kode terisi maka sistem akan mengirimkan kode tersebut melalui API untuk dilakukan verifikasi. API akan mengembalikan sebuah *response* 400 dan pesan “Error” jika kode tidak sesuai, dan jika kode yang dimasukan benar, akan dikembalikan *response* 200 dan pesan “Work” dan akan menuju ke halaman *input email*.



Gambar 5. Activity Diagram Input Code Transaksi

4.4.2. Proses Input Email

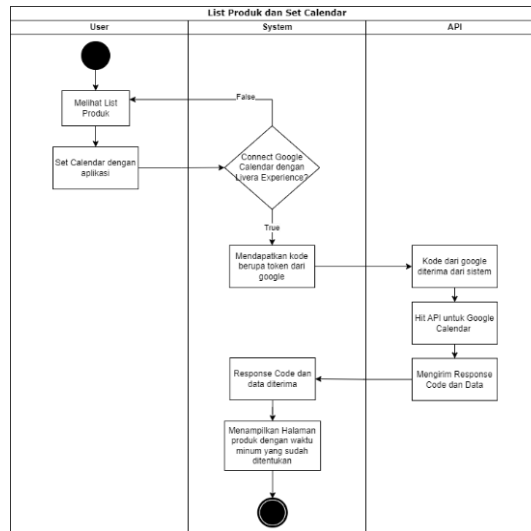
Bedasarkan Gambar 6, *user* akan diminta untuk melakukan *input email* yang sifatnya opsional, jika *user* tidak memilih untuk memasukan *email*, halaman yang berisi daftar produk akan muncul. Tetapi jika *user* ingin memasukkan email, maka akan dilakukan validasi email oleh sistem lalu email tersebut akan dikirimkan ke API yang akan mengembalikan *response* 200 untuk ke halaman daftar produk.



Gambar 6. Activity Diagram Input Email

4.4.3. Proses Penampilan Produk dan Pengaturan Waktu

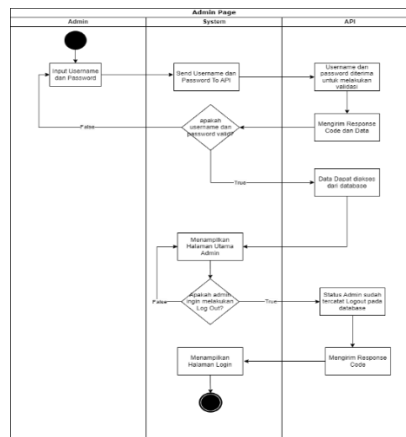
Gambar 7 menjelaskan bahwa pengguna sudah dapat melihat tampilan halaman yang menunjukkan daftar produk yang sudah dibeli. Pengguna dapat menyambungkan google kalender dengan aplikasi lalu sistem akan mengirim *code* yang didapat dari google berupa token. Lalu token akan digunakan untuk melakukan *hit* API untuk Google Calendar. API akan mengembalikan *response* berupa kode dan data yang nantinya akan digunakan sistem untuk memunculkan waktu minum yang sudah ditentukan bersamaan dengan daftar produk.



Gambar 7. Activity Diagram Penampilan Produk dan Set Calendar

4.4.4. Proses Penampilan Halaman Admin

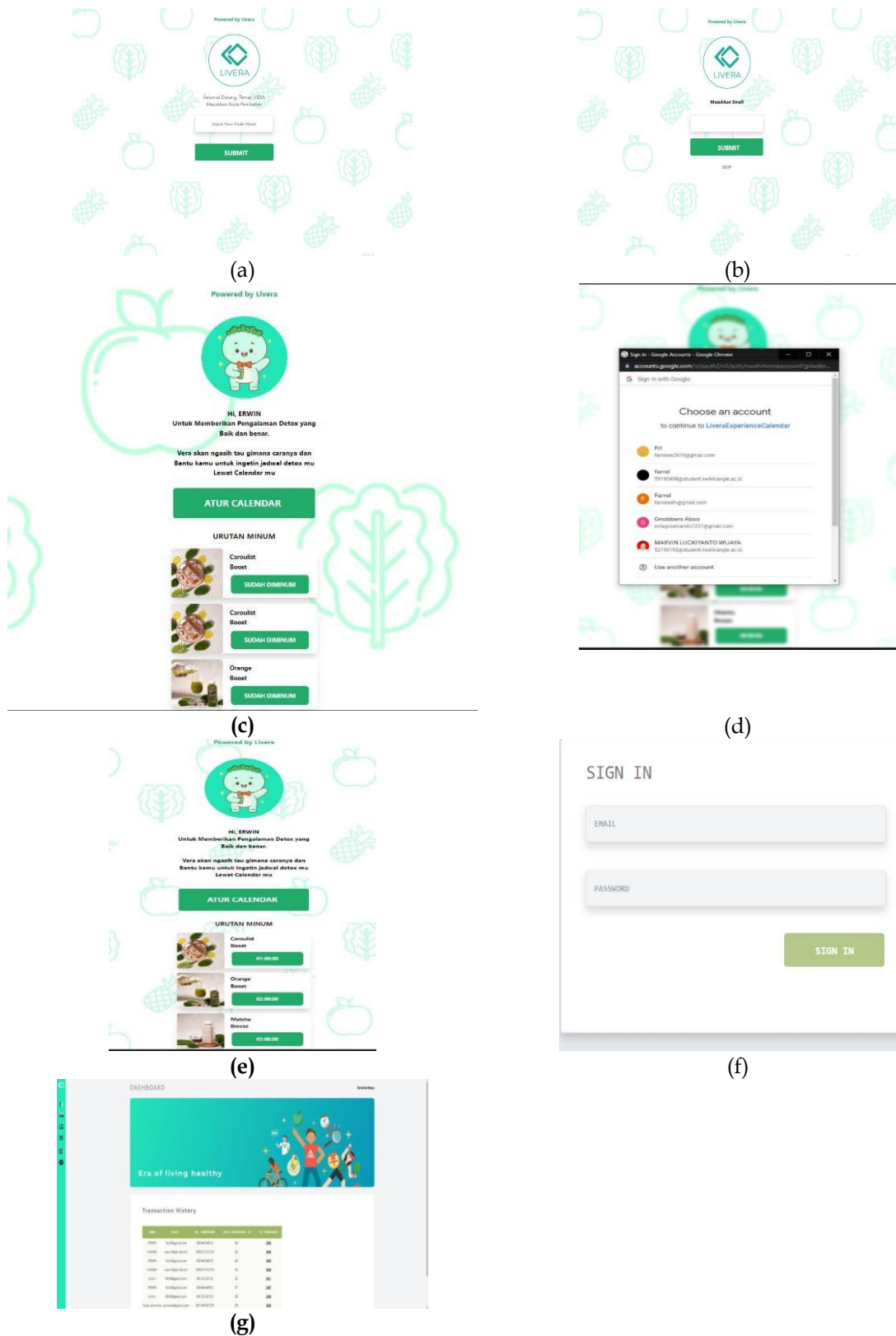
Gambar 8 menjelaskan bahwa admin harus memasukkan *email* dan *password* terlebih dahulu untuk mengakses halaman admin. *Email* dan *password* yang dimasukan oleh admin akan dikirim ke API untuk dilakukan validasi dan akan dikembalikan dalam bentuk token atau *response code*. Jika *email* dan *password* yang dimasukan salah maka dikembalikan ke halaman *login* sedangkan jika *email* dan *password* benar, halaman utama admin akan muncul. Admin dapat melakukan *Logout* untuk merubah status admin dari *online* menjadi *offline* pada basis data.



Gambar 8. Activity Diagram Halaman Admin

4.5. Hasil Program

Berikut merupakan hasil program pada penelitian ini yang disajikan berupa gambar:



Gambar 9 Hasil halaman (a) Input Code untuk user, (b) input email untuk pengguna, (c) daftar produk, (d) atur kalender, (e) Daftar Produk Dengan Waktu Yang Ditentukan, (f) Login Admin, (g) Dashboard Admin

5. Pembahasan

Peneliti mendefinisikan beberapa pembahasan tentang bagaimana aplikasi ini digunakan untuk meningkatkan kepuasan pelanggan. Sebelum aplikasi ini dibuat, perusahaan tidak memiliki sistem atau aplikasi yang dapat mendukung dan meningkatkan kepuasan pelanggan. Aplikasi ini menjadi solusi bagi perusahaan dalam meningkatkan *customer service* mereka dan meringankan beban perusahaan karena aplikasi ini termasuk dalam aplikasi dengan sistem yang otomatis.

Penggunaan metode *scrum* membantu peneliti dalam mempercepat proses pengembangan aplikasi, karena dilakukan perencanaan secara matang dan disesuaikan dengan kebutuhan *project* sehingga keterlambatan pengerjaan *project* dapat diminimalisir.

Berdasarkan temuan dalam penelitian ini, peneliti dapat memperoleh informasi dan pemahaman yang baru sehingga mampu menghasilkan hasil penelitian yang sesuai. Data yang digunakan dalam penelitian ini juga diperoleh oleh peneliti dengan observasi langsung wawancara terstruktur dan studi pustaka.

Beberapa hasil evaluasi pada penelitian ini secara umum adalah: Pertama, aplikasi ini dibuat dan dirancang untuk dapat dikembangkan dan ditingkatkan di masa depan. Dalam mengantisipasi fitur teknologi yang akan datang, peneliti telah menyiapkan ruang pengembangan aplikasi sehingga akan memudahkan pengembangan aplikasi untuk dapat menyesuaikan dengan kebutuhan perusahaan. Kedua, pembuatan aplikasi ini dilakukan menggunakan metode *scrum*, secara umum metode *scrum* dirancang untuk digunakan oleh tim yang bekerja sama kolaboratif, namun beberapa elemen dari metodologi *scrum* dapat dimodifikasi dan diterapkan secara individual sehingga pengembangan aplikasi pada perusahaan dapat dilakukan baik dengan tim maupun secara individual.

6. Kesimpulan

Berdasarkan hasil penelitian, dapat disimpulkan bahwa aplikasi *Livera Experience* memberikan kemudahan bagi pelanggan dalam melihat daftar produk yang telah dibeli sehingga riwayat transaksi dapat diakses dengan lebih praktis dan terorganisir. Integrasi aplikasi dengan Google Calendar memungkinkan terjadinya sinkronisasi otomatis antara jadwal pada aplikasi dan kalender di perangkat pengguna. Fitur ini mendukung pemberian notifikasi dan pengingat yang muncul pada desktop maupun perangkat mobile, sehingga jadwal yang telah ditentukan tidak terlewat. Selain itu, keberadaan aplikasi website *Livera Admin Dashboard* yang bersifat otomatis membantu admin dalam menghemat waktu dan tenaga pada pekerjaan rutin, seperti memasukkan email pelanggan dan data produk yang telah dibeli.

Berdasarkan temuan tersebut, peneliti memberikan beberapa saran bagi perusahaan maupun peneliti selanjutnya yang ingin mengembangkan topik serupa. Disarankan untuk menambahkan fitur QR code pada produk agar pelanggan dapat langsung

terhubung ke aplikasi Livera Experience tanpa perlu memasukkan kode transaksi secara manual. Selain itu, pengembangan aplikasi ke platform mobile seperti Android dan IOS juga direkomendasikan agar aksesibilitas dan jangkauan penggunaan aplikasi menjadi lebih luas.

Referensi

- Blumenthal, S. (2017). *JavaScript: JavaScript for beginners – Learn JavaScript with ease in half the time*. CreateSpace Independent Publishing Platform.
- Cohen, L., Manion, L., & Morrison, K. (2019). *Research methods in education* (8th ed.). T&F India.
- Elmasri, R., & Navathe, S. B. (2016). *Fundamentals of database systems* (7th ed.). Pearson Higher Education.
- Filipova, O., & Vilao, R. (2018). *Software development from A to Z*. Apress.
- Hume, D. A. (2018). *Progressive web apps*. Manning Publications.
- Klaus, P. (2015). *Measuring customer experience: How to develop and execute the most profitable customer experience strategies*. Palgrave Macmillan.
- Laudon, K. C., & Laudon, J. P. (2012). *Essentials of management information systems* (10th ed.). Pearson.
- Plumley, G. (2012). *Website design and development: 100 questions to ask before building a website*. Wiley Publishing.
- Rice, C. (1997). *Understanding computers* (2nd ed.). Butterworth-Heinemann.
- Santoso, A. B. (2021). *Aplikasi komputer*. IAIN Manado Press.
- Sheppard, D. (2017). *Beginning progressive web app development*. Springer.
- Simanungkalit, J. H. U. P. (2012). *Konsep dasar sistem informasi*. Jakarta.
- Sommerville, I. (2016). *Software engineering* (10th ed.). Pearson Education.
- Vicky. (2019). *Aplikasi sistem parkir menggunakan face recognition berbasis Android dengan framework React Native* (Unpublished undergraduate thesis). IBII, Jakarta.