

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

*) Surel korespondensi : sigit.birowo@kwikkiangie.ac.id

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 highly influential factor among business actors in order to remain competitive with other businesses. Livera is a company engaged in the health beverage industry, focusing on weight loss products. Currently, Livera does not yet have an application to support customer satisfaction after purchases are made. Therefore, the researcher intends to design a website-based application aimed at improving customer satisfaction, where customers can consume products according to a predetermined schedule, while also enabling administrators to easily input and monitor data automatically. Customer Experience is an evaluation of feelings that can be identified through the experiences perceived by customers. Customer experience can become one of the most important determining factors for customers. This research was conducted using the Scrum method. The stages of Scrum activities include product backlog, sprint backlog, daily scrum, sprint review, and sprint retrospective. The roles involved in Scrum include the product owner, scrum master, and development team. This study employs a qualitative research method and the Scrum method for system development. The data collection techniques used include non-participant observation, unstructured interviews, and literature studies from books and journals that serve as references regarding 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.v15i1.2120>

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* adalah sebuah kerangka kerja pengembangan dengan penggunaan prinsip pendekatan *agile* untuk mengatasi segala macam masalah secara kreatif dan adaptif. Peneliti menggunakan pendekatan *agile* karena pendekatan ini dapat meningkatkan produktivitas, setiap tim dapat mengerjakan setiap tugas tanpa harus menunggu tim lain untuk menyelesaikan tugasnya. Selain itu, *developer* menjadi memiliki banyak waktu untuk berkembang sesuai keinginan mereka sendiri.

Aplikasi ini mencakup beberapa hal yang bertujuan untuk meningkatkan kepuasan pelanggan, seperti fitur pengingat berupa notifikasi agar pelanggan dapat mengonsumsi produk secara tepat waktu sesuai dengan waktu yang telah ditentukan dan mendapatkan data pelanggan yang nantinya akan digunakan oleh admin pada perusahaan untuk meningkatkan kepuasan pelanggan, seperti memberikan penawaran terbaik kepada pelanggan.

Pada pembuatan aplikasi ini, peneliti sebagai *frontend developer* menggunakan bahasa pemrograman *HyperText Markup Language*, *Cascading Style Sheet*, dan kerangka kerja *react.js*.

Dengan adanya penelitian ini, diharapkan terjadinya peningkatan dalam kepuasan pelanggan dan aplikasi dapat mampu menampung data pelanggan menggunakan situs web dan basis data.

Berdasarkan latar belakang masalah yang telah disebutkan di atas, peneliti mengidentifikasi masalah sebagai berikut: (1) pelanggan belum mendapatkan *customer experience* yang cukup setelah pembelian produk; (2) belum adanya fitur penelusuran kembali untuk produk yang sudah dibeli; (3) belum adanya penawaran terbaik untuk pelanggan setelah pembelian produk; (4) pelanggan belum mendapatkan fitur notifikasi untuk mengonsumsi produk; dan (5) admin belum dapat melihat data *customer details* secara otomatis.

Dari identifikasi masalah, penulis membatasi masalah sebagai berikut: (1) belum adanya fitur penelusuran kembali untuk produk yang sudah dibeli; (2) pelanggan belum mendapatkan fitur notifikasi untuk mengonsumsi produk; dan (3) admin belum dapat melihat data *customer details* secara otomatis

Penelitian ini bertujuan untuk meningkatkan kepuasan pelanggan dan layanan perusahaan dengan mengimplementasikan fitur untuk melihat kembali produk yang sudah dibeli dan pengingat berupa notifikasi pada aplikasi, serta pengambilan dan penampilan data pelanggan yang bertujuan agar admin dapat melihat data pelanggan secara otomatis.

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 dikemudian 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 dilakukan dengan menggunakan langkah-langkah sebagai berikut:

3.3.1. Reduksi Data

Reduksi data merupakan tahapan untuk menyetik 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 pengambilan tindakan. Dengan penyajian tersebut akan dapat dipahami apa yang harus dilakukan selanjutnya.

3.3.3. Penarikan Kesimpulan

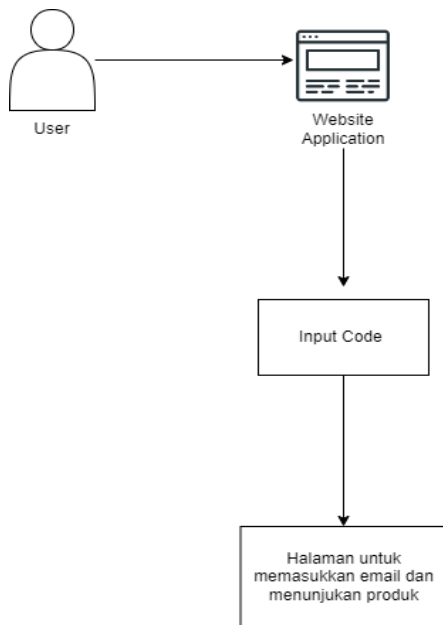
Penarikan kesimpulan hanyalah sebagian dari suatu kegiatan konfigurasi yang utuh. Kesimpulan awal yang dikemukakan masih bersifat sementara, dan akan mengalami

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.

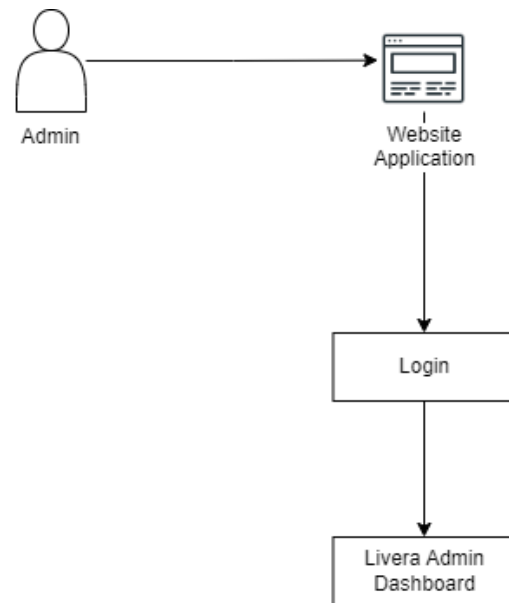
4. Hasil

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

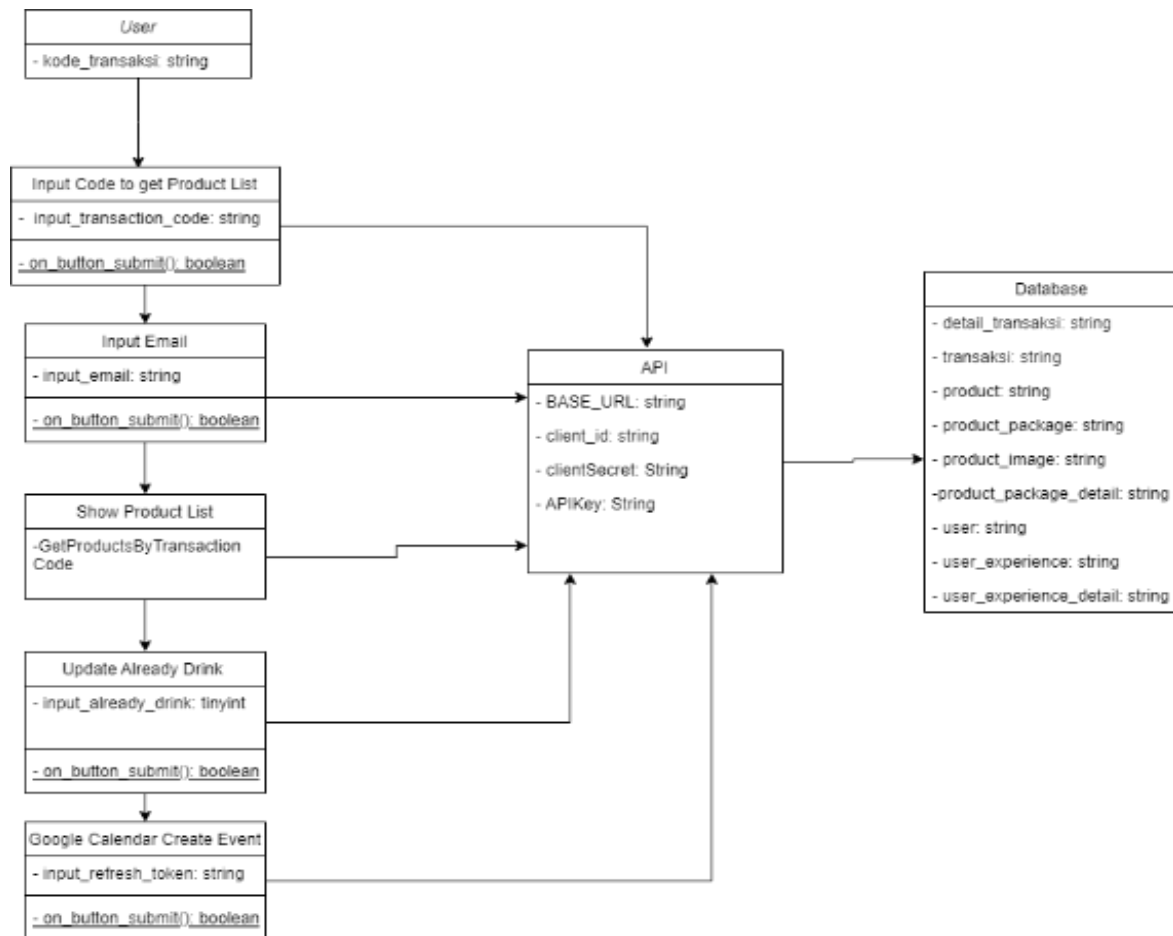
Aplikasi *website livera admin dashboard* ini dapat diakses oleh admin dengan memasukkan *Email* dan *password* lalu admin dapat mengakses website yang berisi data pengguna.

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*

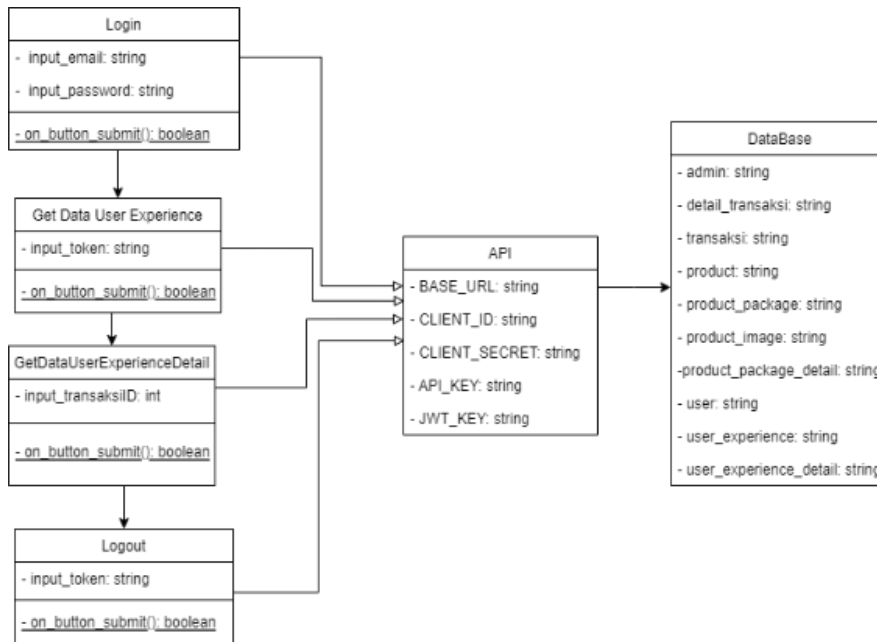
refresh_token. Semua kelas yang melakukan *input* menggunakan modul *on_submit_button* untuk disambungkan ke API. Semua *input* yang sudah berhasil dimasukan melalui API akan dilakukan validasi dengan basis data.



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*.

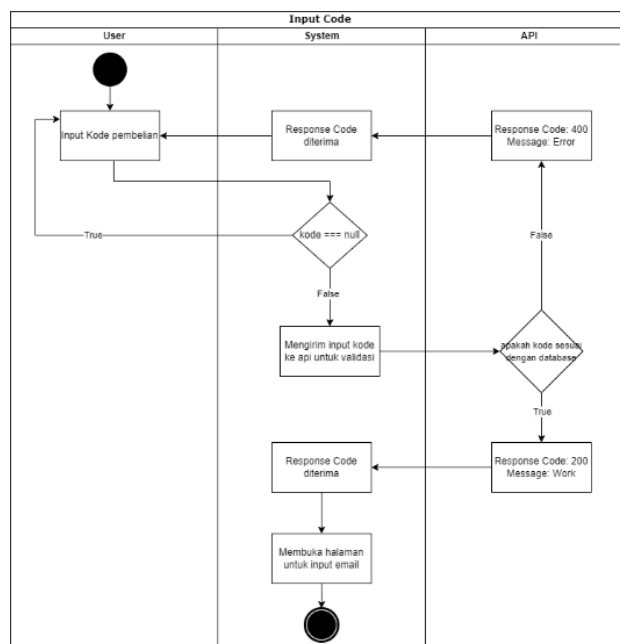


Gambar 4. Class Diagram Sistem Admin Dashboard

4.4. Activity Diagram

4.4.1. Proses Input Code

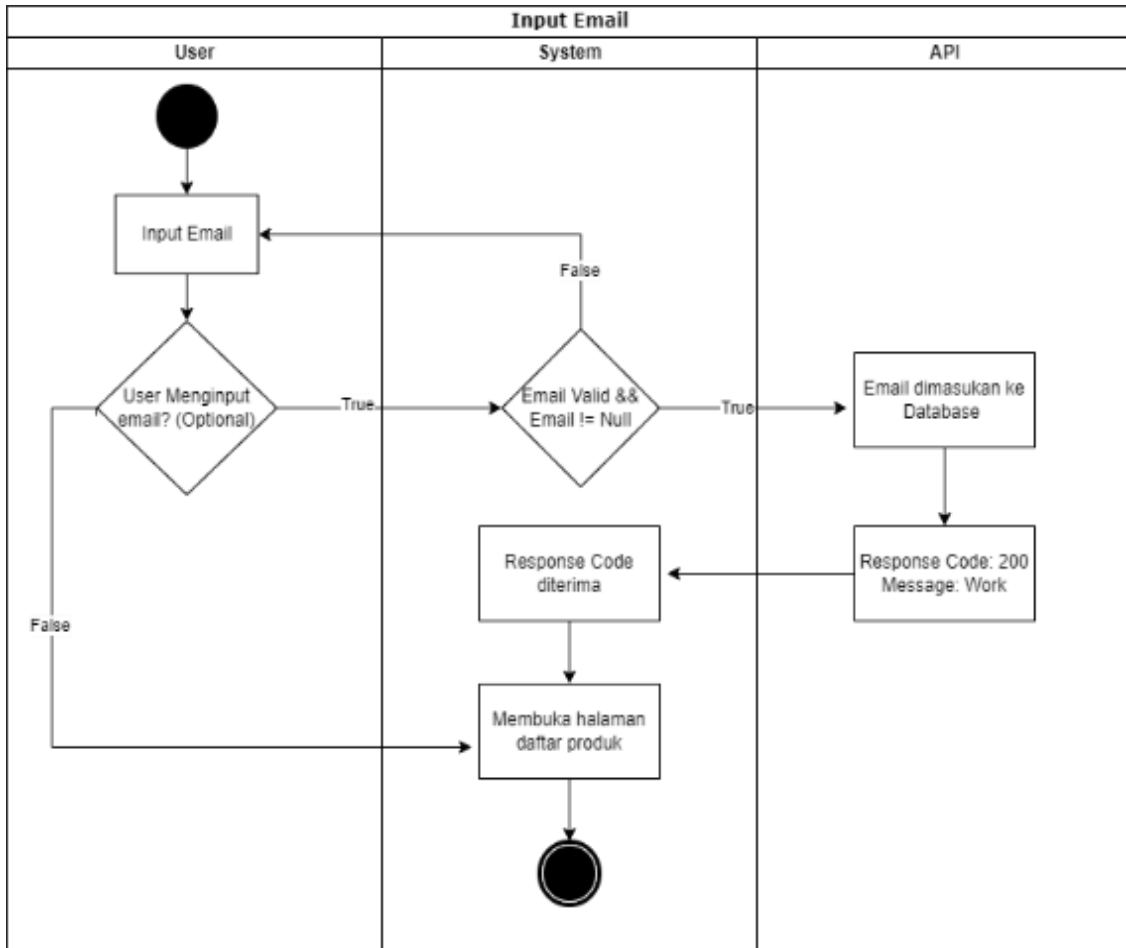
Bedasarkan gambar 5, pertama *user* akan diminta untuk memasukkan kode transaksi. Jika sistem menerima kode sebagai data kosong maka halaman akan tetap berada di halaman *input* kode, jika kode terisi maka sistem akan mengirimkan kode tersebut melalui API untuk dilakukan verifikasi. API akan mengembalikan 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

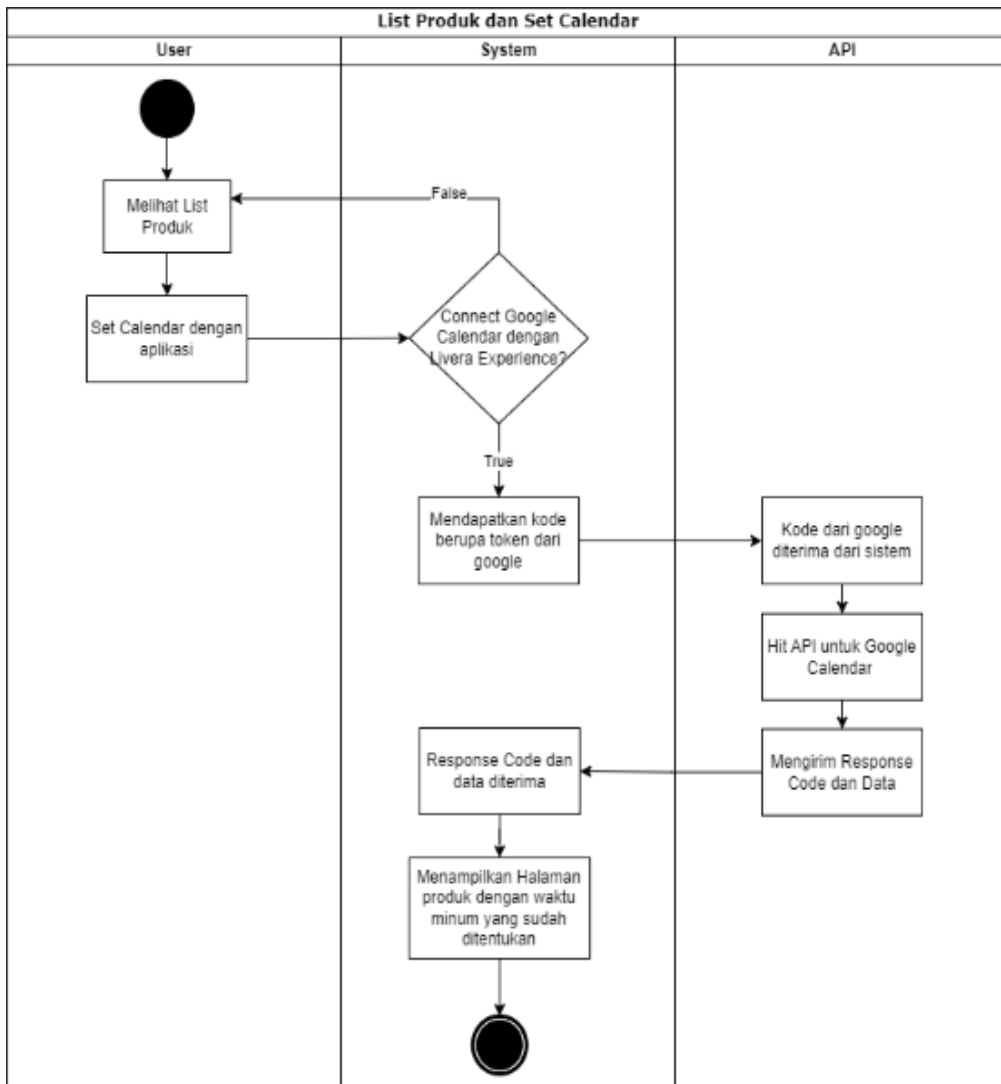
Bedasarkan gambar 6, *user* akan diminta untuk melakukan *input email* yang sifatnya opsional, jika *user* tidak memilih untuk memasukan *email*, halaman yang berisi daftar produk akan muncul. Tetapi jika *user* ingin memasukkan email, maka akan dilakukan validasi email oleh sistem lalu email tersebut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

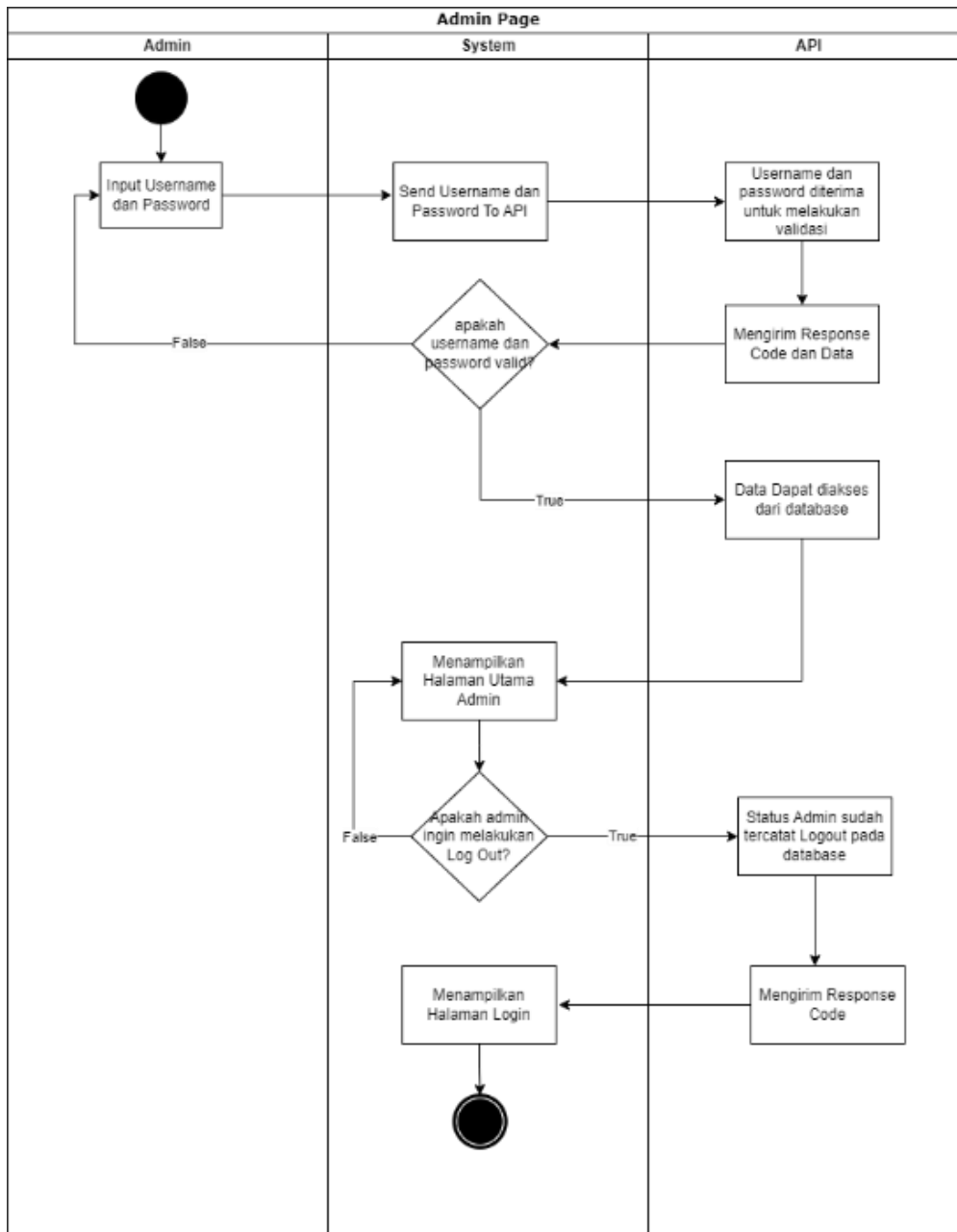
Gambar 7 menjelaskan bahwa pengguna sudah dapat melihat tampilan halaman yang menunjukkan daftar produk yang sudah dibeli. Pengguna dapat menyambungkan google kalender dengan aplikasi lalu sistem akan mengirim *code* yang didapat dari google berupa token. Lalu token akan digunakan untuk melakukan *hit* 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

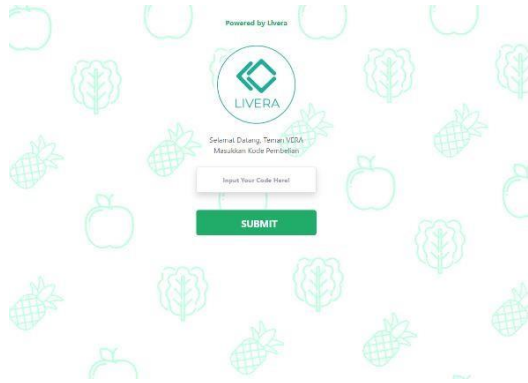
Gambar 8 menjelaskan bahwa admin harus memasukkan *email* dan *password* terlebih dahulu untuk mengakses halaman admin. *Email* dan *password* yang dimasukan oleh admin akan dikirim ke API untuk dilakukan validasi dan akan dikembalikan dalam bentuk token atau *response code*. Jika *email* dan *password* 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.



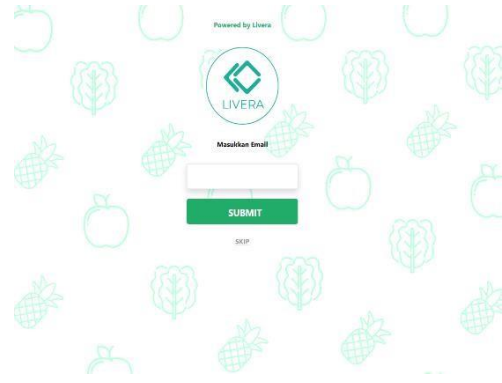
Gambar 8. Activity Diagram Halaman Admin

4.5. Hasil Program

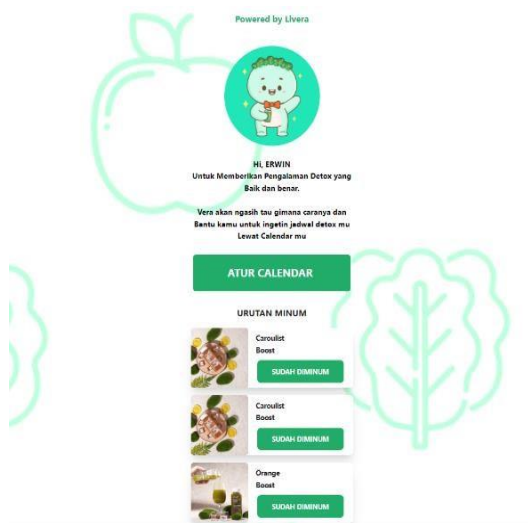
Berikut merupakan hasil program pada penelitian ini yang disajikan berupa gambar (Gambar 9 - 15):



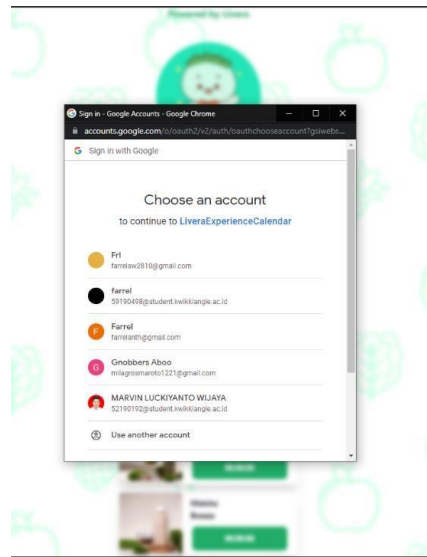
Gambar 9 Hasil Halaman Input Code untuk use



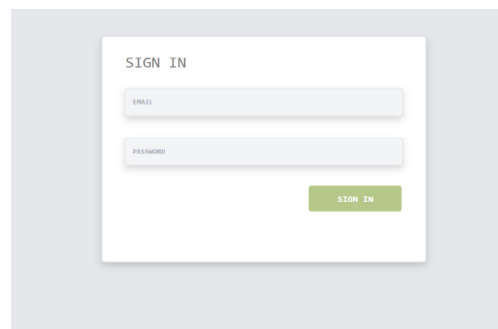
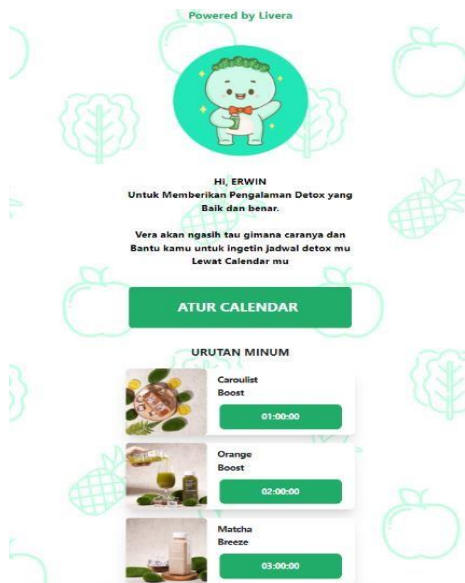
Gambar 10 Hasil Halaman input email untuk pengguna



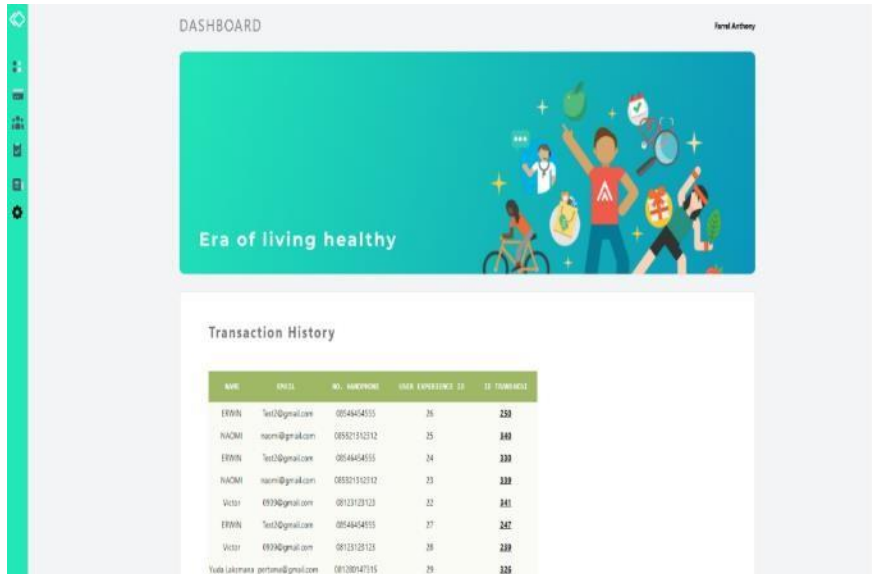
Gambar 11 Hasil Halaman daftar produk



Gambar 12 Hasil Halaman atur kalender



Gambar 13 Hasil Halaman Daftar Produk Dengan Waktu Yang Ditentukan Gambar 14 Halaman Login Admin



Gambar 15 Halaman 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.

Bedasarkan 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 pada hasil penelitian ini, maka peneliti dapat menarik beberapakesimpulan sebagai berikut: (1) dengan aplikasi *Livera Experience*, pelanggan dapat melihat daftar produk yang sudah dibeli; (2) aplikasi *Livera Experience* yang terintegrasi dengan *google calendar* memungkinkan aplikasi untuk melakukan sinkronisasi secara otomatis antara jadwal dengan kalender pada perangkat pengguna. Aplikasi dapat memberikan pemberitahuan dan pengingat yang muncul di dekstop atau perangkat mobile, sehingga jadwal yang sudah ditentukan tidak akan terlewat. (3) dengan adanya aplikasi *website* *Livera Admin Dashboard* yang otomatis, admin dapat menghemat waktu dan tenaga dalam menjalankan tugas-tugas yang dilakukan secara rutin seperti memasukan *email* atau produk-produk yang sudah dibeli pelanggan.

Berdasarkan hasil penelitian dan kesimpulan, peneliti memiliki saran untuk perusahaan dan para peneliti di masa mendatang yang ingin mengembangkan ataupun memilih topik sejenis yaitu: (1) menambahkan QR kode pada produk yang memungkinkan pelanggan untuk langsung disambungkan ke aplikasi *livera experience* tanpa harus memasukan kode transaksi; (2) melakukan pengembangan aplikasi yang lebih luas seperti integrasi ke aplikasi *mobile* (Android, IOS); (3) dalam menu admin, untuk pengembangan selanjutnya dapat ditambahkan fitur untuk *export* data pelanggan kedalam bentuk *Microsoft Excel* atau *Google Workspace*; serta (4) melakukan pemeliharaan dan pembaruan terhadap sistem secara berkala agar dapat meningkatkan dan mempertahankan kinerja sistem.

Daftar Pustaka

- Blumenthal, Stephen. (2017), *JavaScript: Javascript tFor Beginners - Learn JavaScript with ease in Half the Time*, CreateSpace Independent Publishing Platform.
- Cohen, Louis, Lawrence Manion, and Keith Morrison. (2019), *Research Methods In Education* 8Th Edition, T&F INDIA.
- Elmasri, Ramez & Shamkant B. Navathe. (2016), *Fundamentals of Database Systems* (7th ed), New York: Pearson Higher Education.
- Filipova, Olga & Rui Vilao. (2018), *Software Development From A to Z*. Berlin, Germany: Apress.
- Hume, Dean Alan. (2018), *Progressive Web Apps*, New York: Manning Publications Co.
- Klaus, Phil. (2015), *Measuring Customer Experience: How to Develop and Execute the Most Profitable Customer Experience Strategies*, New York: Palgrave Macmillan.

- Laudon, C. Kenneth & Jane P. (2012), *Essentials of Management Information Systems* (10th Edition), New York: Pearson.
- Plumley, George. (2012), *Website Design and Development: 100 Questions to Ask Before Building a Website*, Indianapolis: Wiley Publishing, Inc.
- Rice, Chris. (1997), *Understanding s*, Second Edition, Great Britain: Butterworth-Heinemann.
- Santoso, Agung Budi. (2021), *Aplikasi Komputer*, Manado: IAIN MANADO PRESS.
- Sheppard, Dennis. (2017), *Beginning Progressive Web App Development*, New York: Springer Science+Business Media New York.
- Simanungkalit, Janry Haposan U. P. (2012), *Konsep Dasar Sistem Informasi*. Jakarta.
- Sommerville, Ian. (2016), *Software Engineering* (10th ed), United Kingdom: Pearson Education.
- Vicky (2019), *Skripsi: Aplikasi Sistem Parkir Menggunakan Face Recognition Berbasis Android Dengan Framework React Native*, IBII, Jakarta (tidak dipublikasikan).