

BAB III

METODE PENELITIAN

3.1 Subyek dan obyek penelitian

Dalam penelitian ini, subjek penelitian meliputi Pengembang perangkat lunak yang membutuhkan *Source Code* dan platform *E-Commerce Source Code*, sementara obyek penelitian meliputi, rancang bangun *website* platform *E-Commerce* penjualan *Source Code* dengan nama Gencode menggunakan metode *Scrum*.

3.2 Alat dan Bahan Penelitian

Alat dan bahan yang digunakan untuk penelitian ini antara lain:

3.2.1. Perangkat Keras

Perangkat keras yang digunakan dalam pembuatan yaitu laptop dengan spesifikasi di tunjukan pada tabel 3.1 berikut:

Tabel 3.1 Perangkat Keras

Perangkat	Spesifikasi
Prosesor	AMD Ryzen 5 5600H 12 CPU 3.3 GHz
Memori	8GB
VGA	AMD Radeon Graphic 4GB VRAM
Penyimpanan	SSD M2 Nvme 512GB

3.2.2. Perangkat Lunak

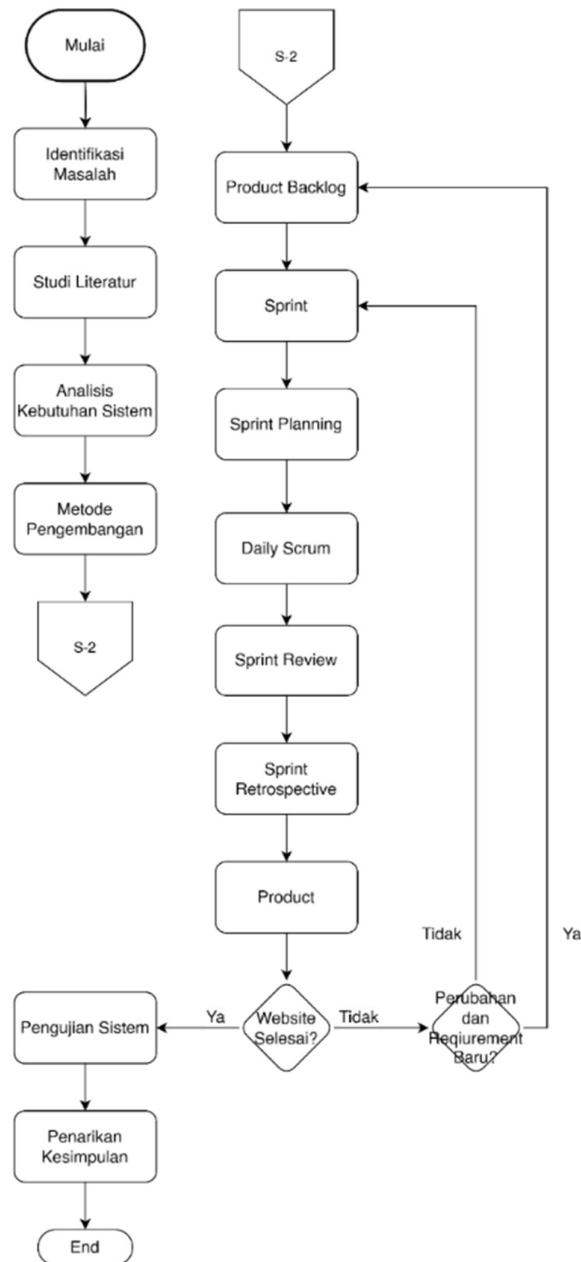
Tabel 3.2 Perangkat Lunak

Nama	Penjelasan
Sistem operasi windows 11 Home 64bit	Digunakan untuk menjalankan sistem operasi.
Visual Studio Code 1.85 64bit	<i>Code editor</i> dalam pembuatan <i>website</i>

Nama	Penjelasan
Laragon 6.0 64bit	<i>Database</i> untuk penyimpanan data.
Google Chrome 121.0 64bit	<i>Browser</i> yang digunakan untuk mencari referensi jurnal dan pembuatan <i>website</i> .
Draw.io 23.0	Digunakan untuk pembuatan diagram alir penelitian atau proses penelitian.
Microsoft Excel 2021	Digunakan untuk menghitung data responden yang sudah dikumpulkan dan membuat tabel.

3.3 Diagram Alir Penelitian/Proses Penelitian

Bagian ini menampilkan diagram alir penelitian atau proses penelitian yang telah dilakukan dalam rangka memahami dan menganalisis langkah-langkah yang diambil selama pelaksanaan penelitian. Diagram alir ini, seperti yang tergambar pada gambar 3.1, memberikan visualisasi yang jelas mengenai urutan kegiatan dan hubungannya dalam proses penelitian. Proses dimulai dengan langkah-langkah awal, yang kemudian berkembang ke langkah-langkah berikutnya seiring berjalannya waktu. Setiap simbol dan panah dalam diagram alir tersebut mewakili suatu aktivitas atau keputusan yang diambil dalam penelitian. Melalui gambaran visual ini, dapat memberikan gambaran tentang bagaimana penelitian ini dirancang, dilaksanakan, dan diselesaikan penelitian ini.



Gambar 3.1 Diagram Alir Penelitian/Proses Penelitian

3.3.1. Identifikasi Masalah

Pada identifikasi masalah bahwa sesuai dengan konteks penelitian yang telah dijelaskan, Penelitian ini untuk mengeksplorasi kebutuhan pengembang perangkat lunak akan *Source Code*, serta

rancang bangun *website E-Commerce* penjualan *Source Code* dengan nama Gencode menggunakan metode *Scrum* sebagai solusi untuk mengatasi kekurangan tersebut.

3.3.2. Studi Literatur

Berbagai tahapan penelitian ini adalah mengumpulkan data terkait dengan perancangan *website*. Peneliti melakukan literatur di situs sebagai bahan pengantar. Data yang diperoleh direferensikan dalam bentuk jurnal dan artikel. Tujuan dari penelitian literatur adalah untuk memperkuat validitas dan fitur formatif yang hadir di situs sebagai media pengantar yang akan disajikan oleh penelitian ini dan menjadi dasar desain.

3.3.3. Analisis kebutuhan Sistem

Dari hasil kuesioner untuk menentukan kebutuhan dalam *E-Commerce Source Code*, yang pertama adalah melakukan analisis persyaratan fungsional, sehingga pengembang dapat memahami fitur dan fungsi apa yang dibutuhkan setiap pengguna, pada tabel 3.3 terdapat daftar kebutuhan sistem.

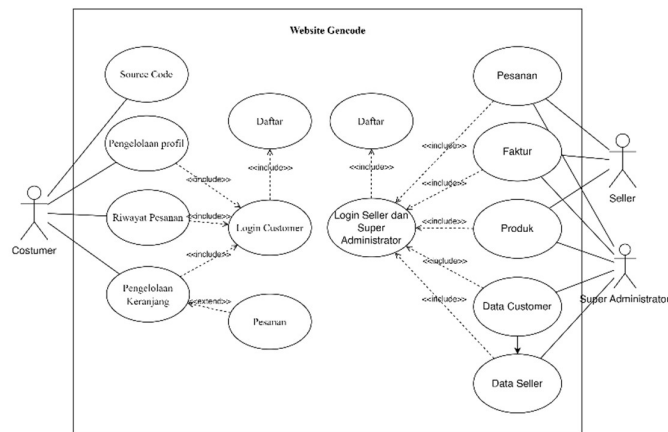
Tabel 3.3 Daftar Kebutuhan Sistem

Pengguna	Kebutuhan Sistem
<i>Administrator</i>	Dashboard
	Pengelolaan Data Pesanan
	Pengelolaan Data Faktur
	Pengelolaan Data Produk
	Pengelolaan Data <i>Customer</i>
<i>Seller</i>	Pengelolaan Data <i>Seller</i>
	Pengelolaan Data Pesanan
	Pengelolaan Data Faktur
<i>Customer</i>	Pengelolaan Data Produk
	Akses Beranda
	Akses <i>Source Code</i>
	Akses Tentang Kami
	Melakukan <i>Login</i>

Pengguna	Kebutuhan Sistem
	Melakukan Daftar
	Melakukan Keluar
	Pengelolaan Profil
	Pengelolaan Keranjang
	Akses Riwayat Pesanan

3.3.3.1. Use Case Diagram

Use case diagram digunakan untuk menggambarkan perilaku sistem dari sisi *actor*, pada *use case* diagram terdapat fungsi-fungsi yang dapat dilakukan kepada sistem sesuai dengan *actor*, Dalam *use case* diagram di ini menjelaskan interaksi antara sistem dan *customer* dengan menunjukkan fungsionalitas sistem dari sudut pandang *customer*.

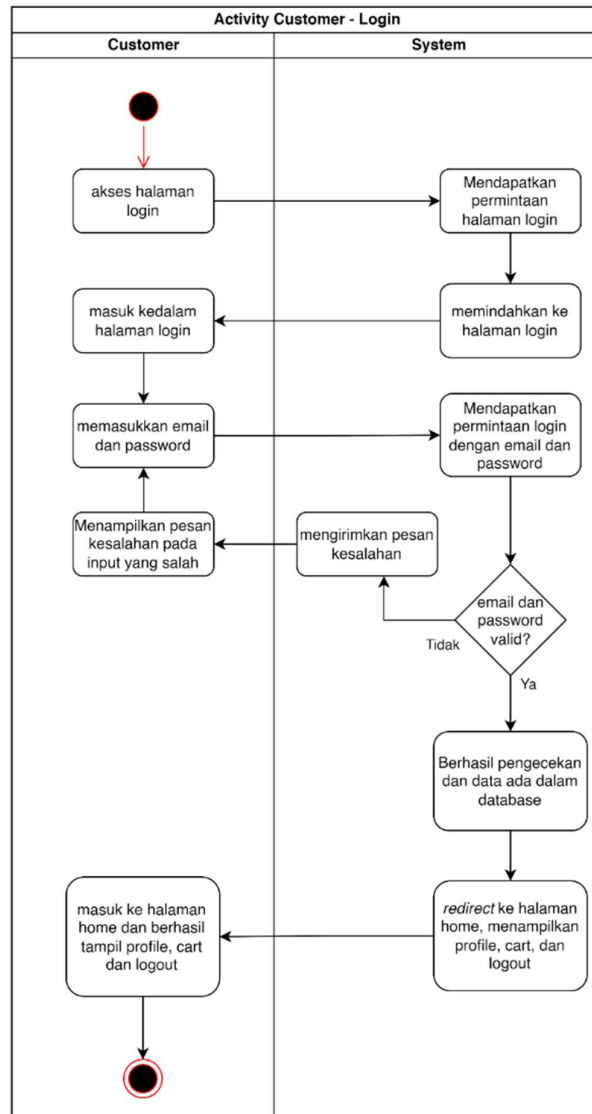


Gambar 3.2 Use Case Website Gencode

Pada Gambar 3.2, Terdapat 3 *actor* yaitu *customer*, *seller* dan *super administrator*, pada *use case* ini *customer* dapat mengakses *website* tanpa perlu melakukan *login* untuk melihat apa isi dari *website* tersebut namun ketika ingin melakukan transaksi maka *customer* harus *login* terlebih dahulu, sedangkan *seller* dan *super administrator* dapat mengakses *dashboard*.

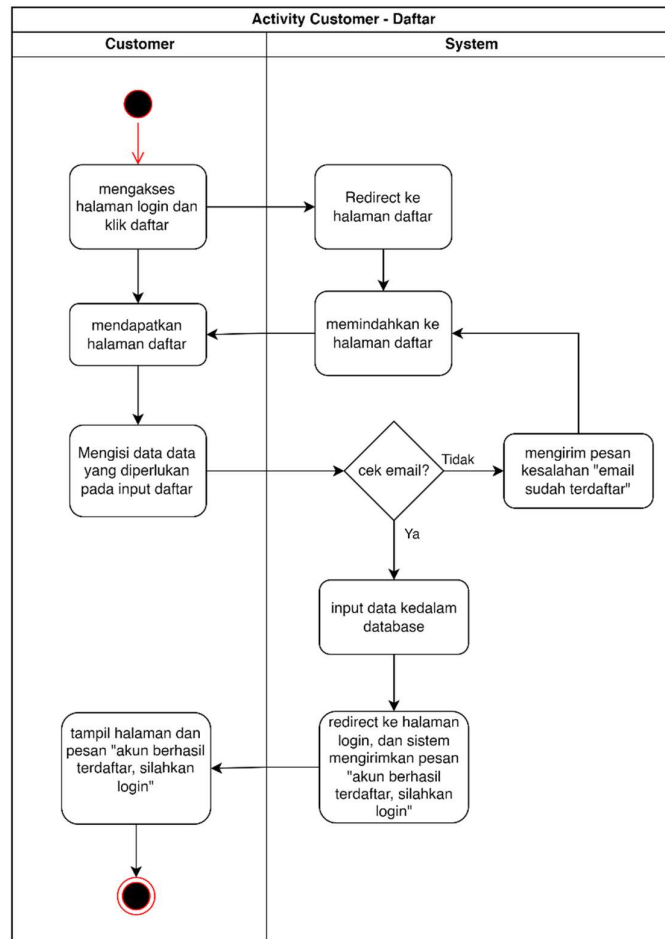
Seller bisa mengakses semua yang ada pada *dashboard* setelah melakukan *login*, kecuali mengakses data *customer* dan data *seller*, karena hal tersebut hanya bisa dilakukan oleh *super administrator*.

3.3.3.2. Activity Diagram



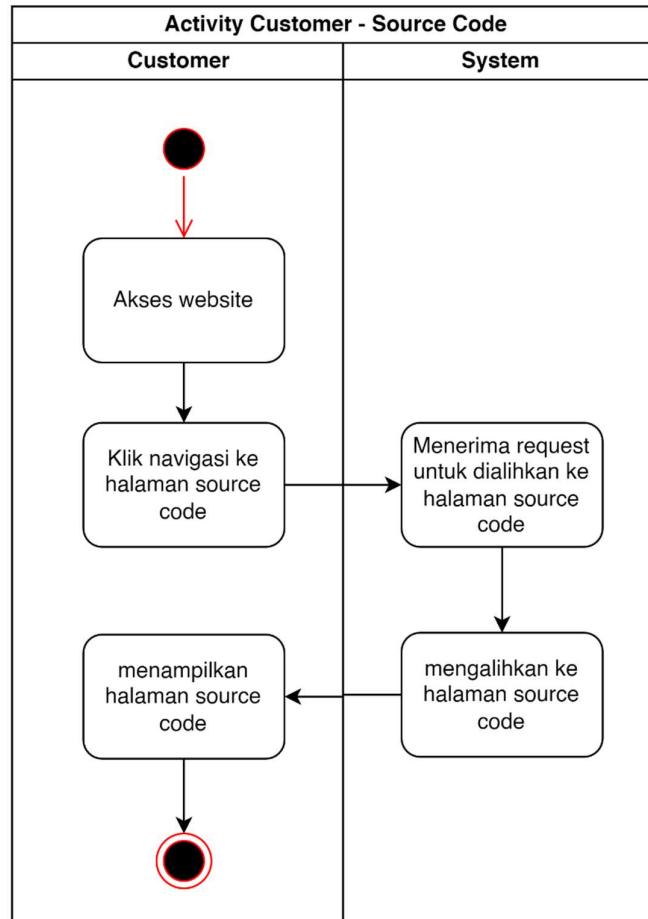
Gambar 3.3 Activity Diagram Login Customer

Gambar 3.3, activity diagram *login* pada *customer* ini ditunjukkan alur bagaimana *customer* melakukan *login* pada *website* Gencode.



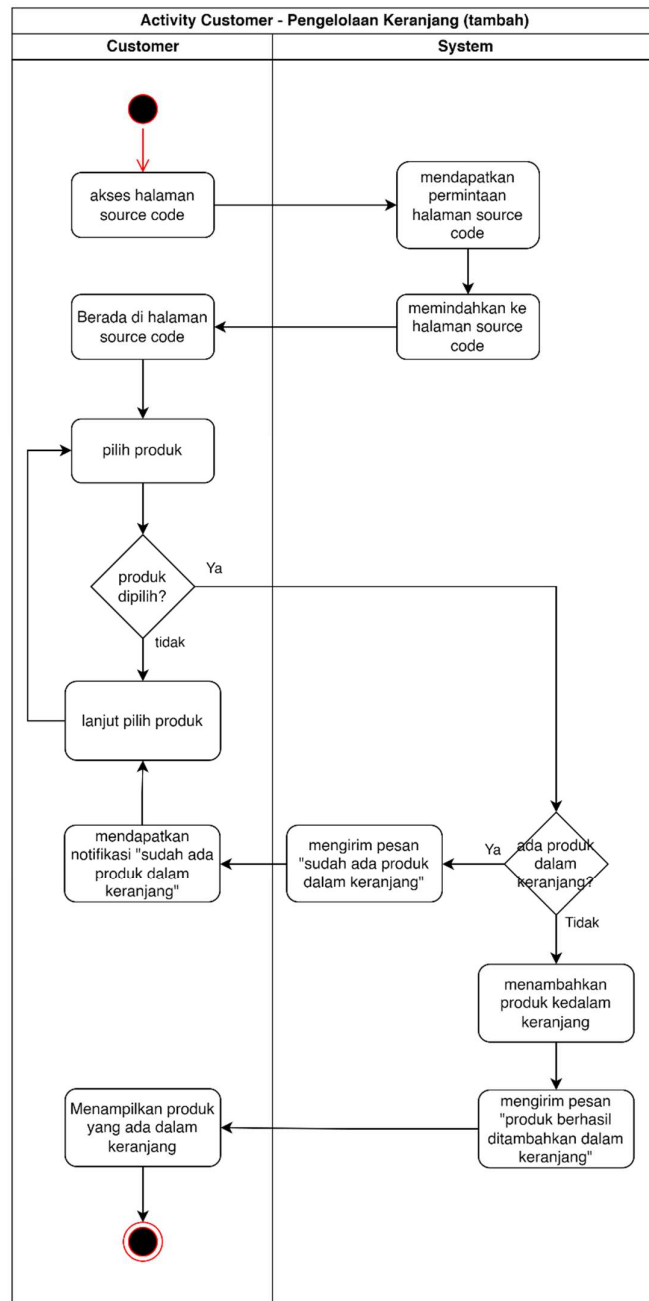
Gambar 3.4 *Activity Diagram* Daftar *Customer*

Gambar 3.4, *activity* ini menunjukkan bagaimana alur *customer* untuk melakukan daftar atau membuat akun dari *error handler* hingga akun berhasil dibuat.



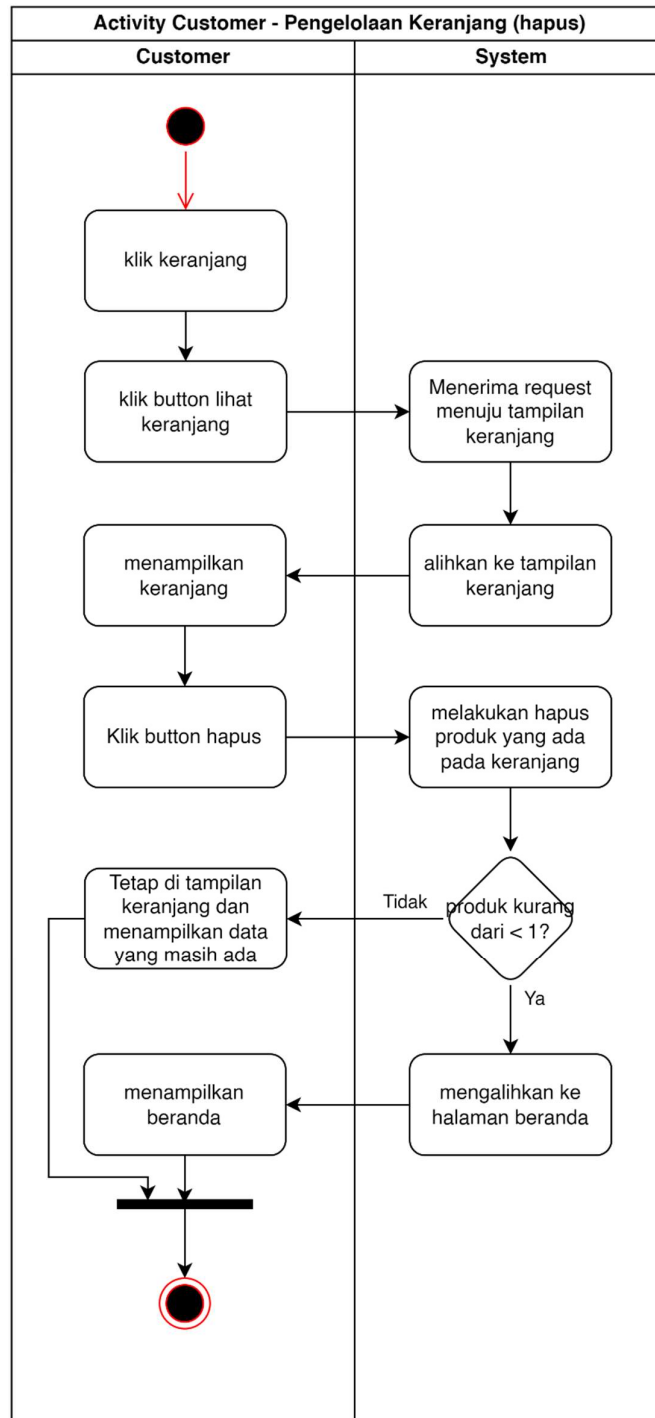
Gambar 3.5 Activity Diagram halaman *Source Code*
Customer

Gambar 3.5, *activity* ini menjelaskan bagaimana *customer* dapat mengakses halaman *Source Code*.



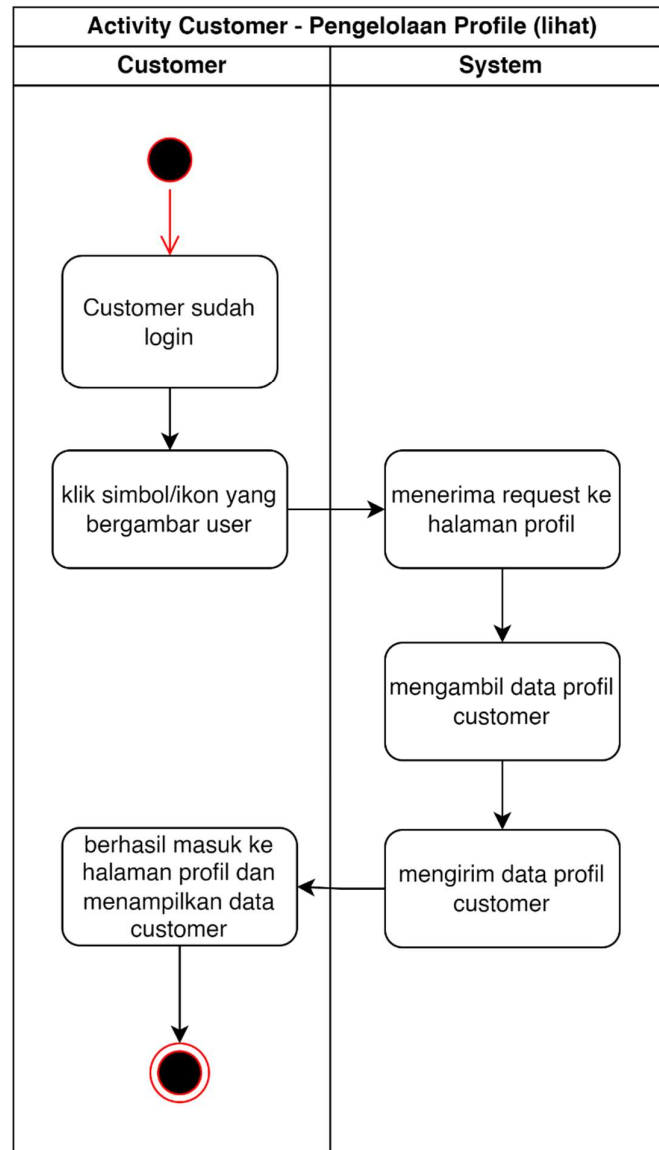
Gambar 3.6 *Activity* Diagram Pengelolaan Keranjang
(tambah)

Gambar 3.6, *Activity* ini menjelaskan bagaimana *customer* bisa melakukan tambah pada pengelolaan keranjang ketika sudah melakukan *login*.



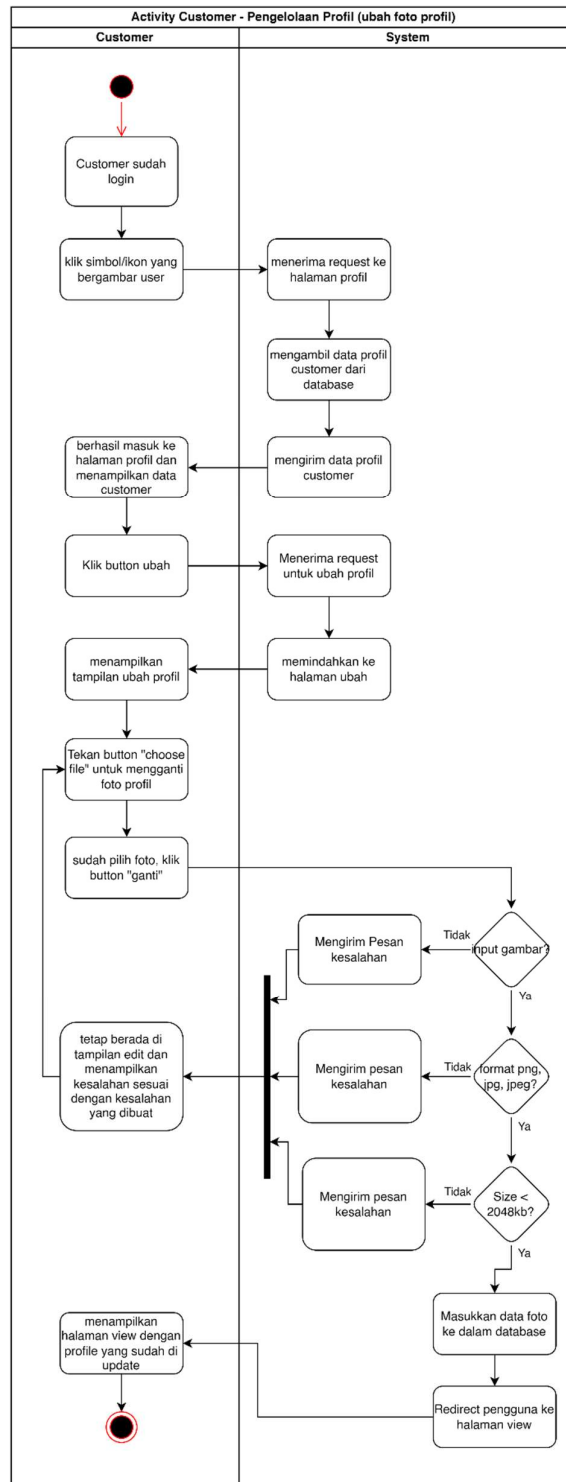
Gambar 3.7 *Activity* Diagram Pengelolaan Keranjang (Hapus)

Gambar 3.7, *activity* ini menjelaskan bagaimana alur ketika *customer* melakukan hapus produk pada keranjang sesuai dengan akun *customer* masing-masing.



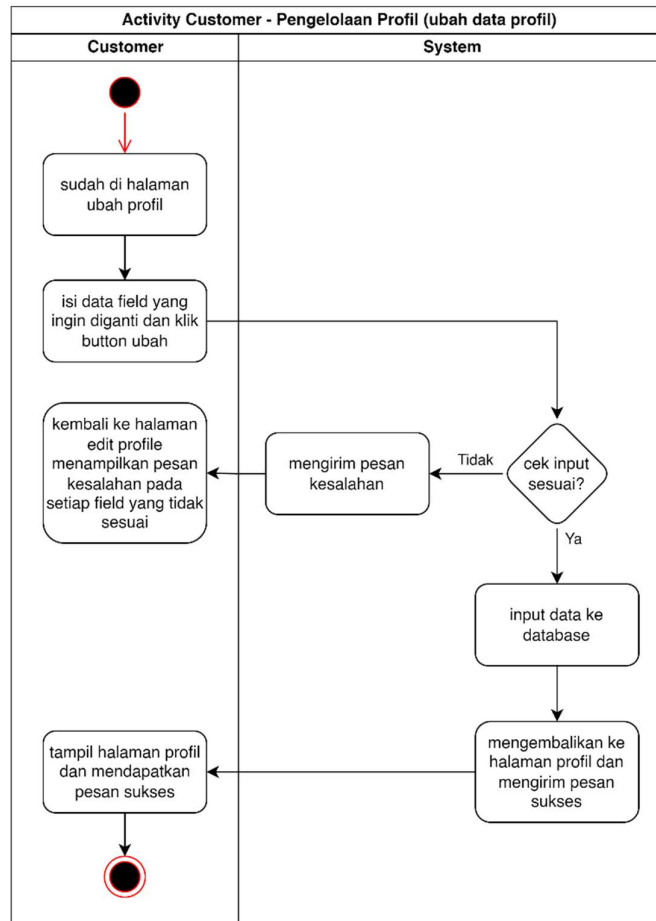
Gambar 3.8 *Activity* Diagram Pengelolaan Profil
Customer

Gambar 3.8, *activity* ini alur dimana *customer* ingin melihat data-data diri saat daftar dan ada beberapa data yang belum terisi.



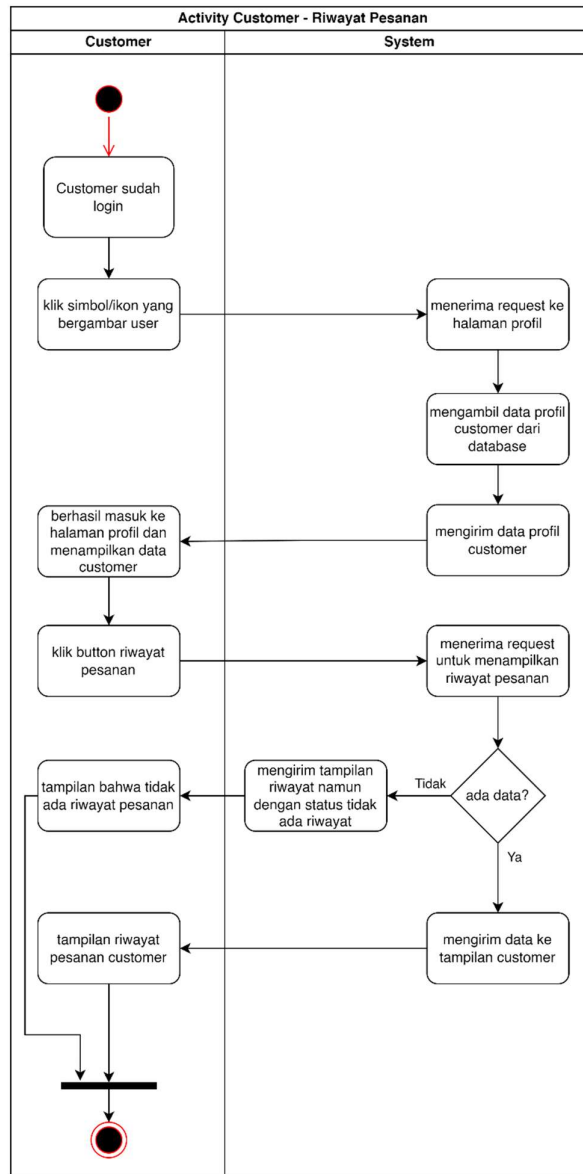
Gambar 3.9 Activity Diagram Pengelolaan Profil (ubah foto profil) Customer

Gambar 3.9, *activity* tersebut menggambarkan tentang alur dari *customer* melakukan ubah foto profil dengan melalui beberapa tahap di sistem dan *error handler* jika terjadi kesalahan data.



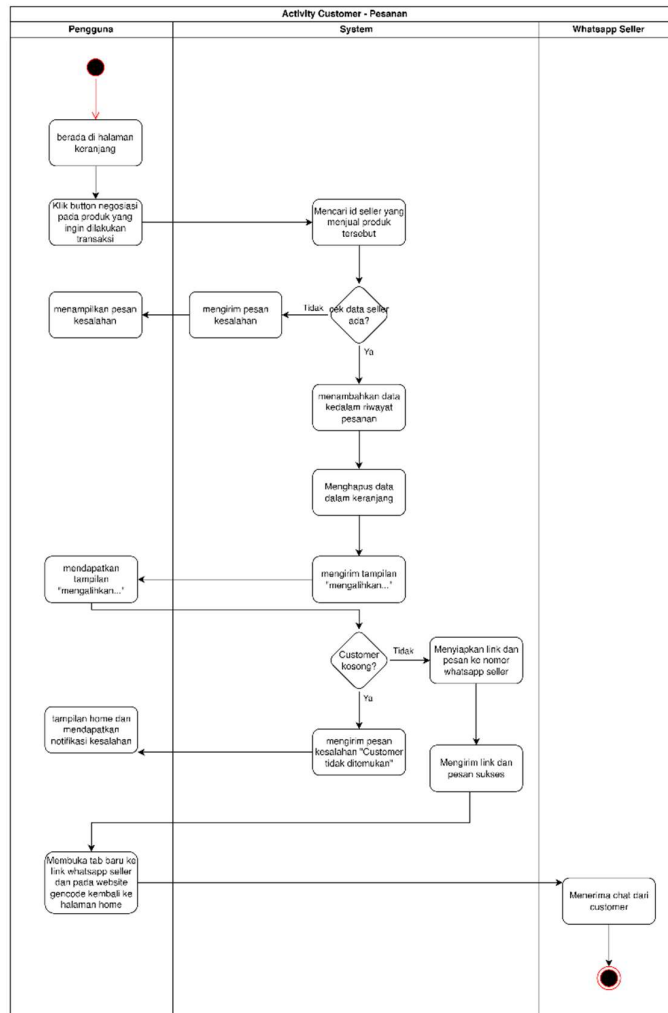
Gambar 3.10 *Activity* Diagram Pengelolaan Profil (ubah data profil)

Gambar 3.10, *activity* ini menggambarkan proses alur dari *customer* pada saat melakukan ubah data yang diperlukan.



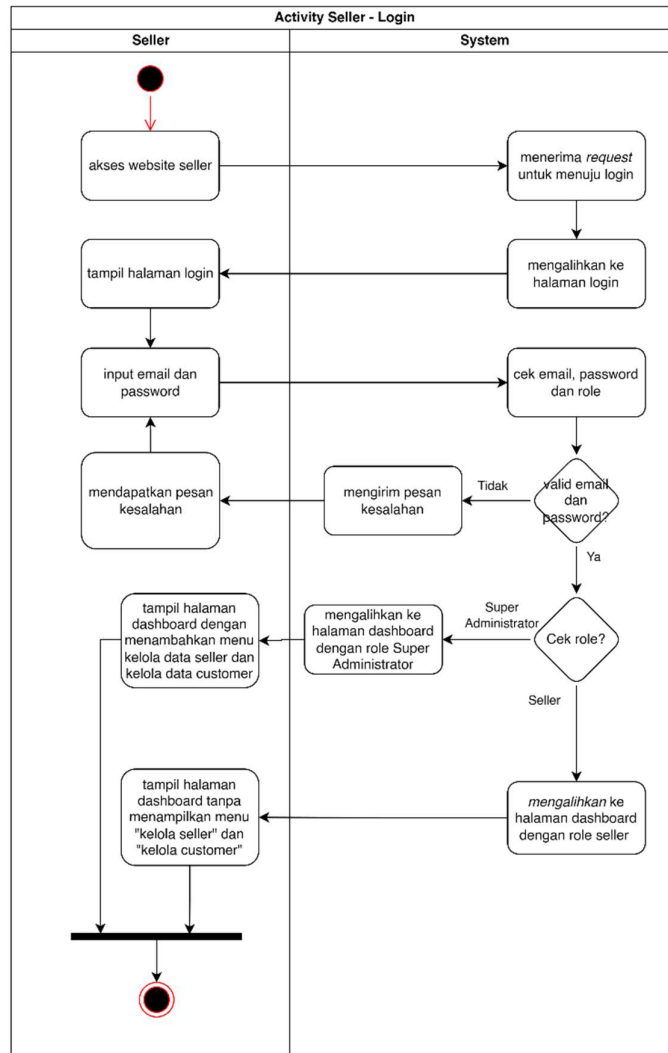
Gambar 3.11 *Activity Diagram Riwayat Order Customer*

Gambar 3.11, *activity* ini adalah alur *customer* saat ingin melihat riwayat pesanan, dan ketika riwayat pesanan nya kosong maka hanya menampilkan pesan “tidak ada data riwayat pesanan yang dapat ditampilkan” dan ketika ada riwayat order maka akan menampilkan data order yang *customer* sudah pernah lakukan.



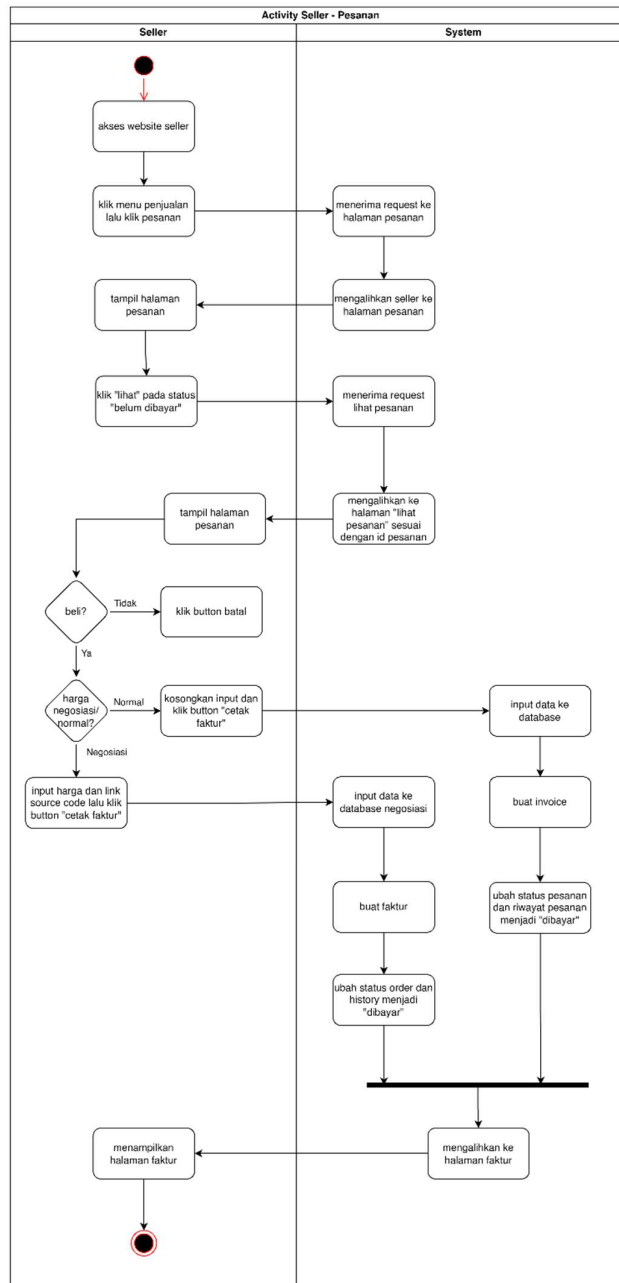
Gambar 3.12 *Activity Diagram Pesanan Customer*

Gambar 3.12, pada *activity* diagram ini pesanan untuk *customer*, dimana ketika pengguna sudah ada produk di dalam keranjang yang ingin di beli dan ingin melakukan negosiasi maka klik *button* negosiasi pada produk yang ingin dilakukan negosiasi, setelah itu akan berpindah ke *link* baru yang mengarah ke WhatsApp *Seller*.



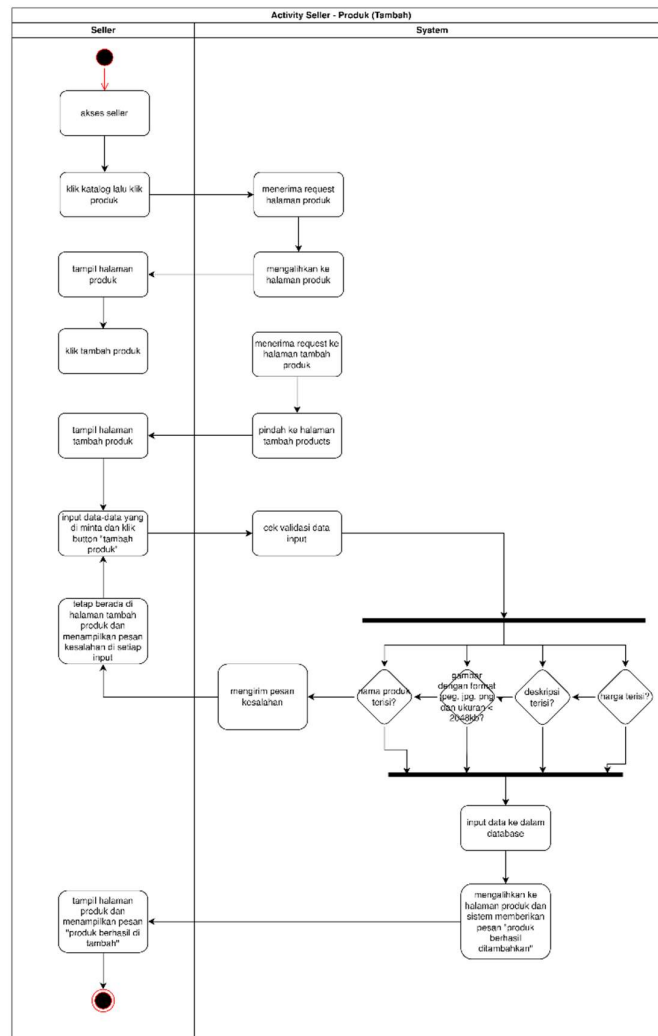
Gambar 3.13 *Activity Login Seller*

Pada gambar 3.13 adalah alur *activity* yang akan dijalankan ketika *seller* melakukan *login*.



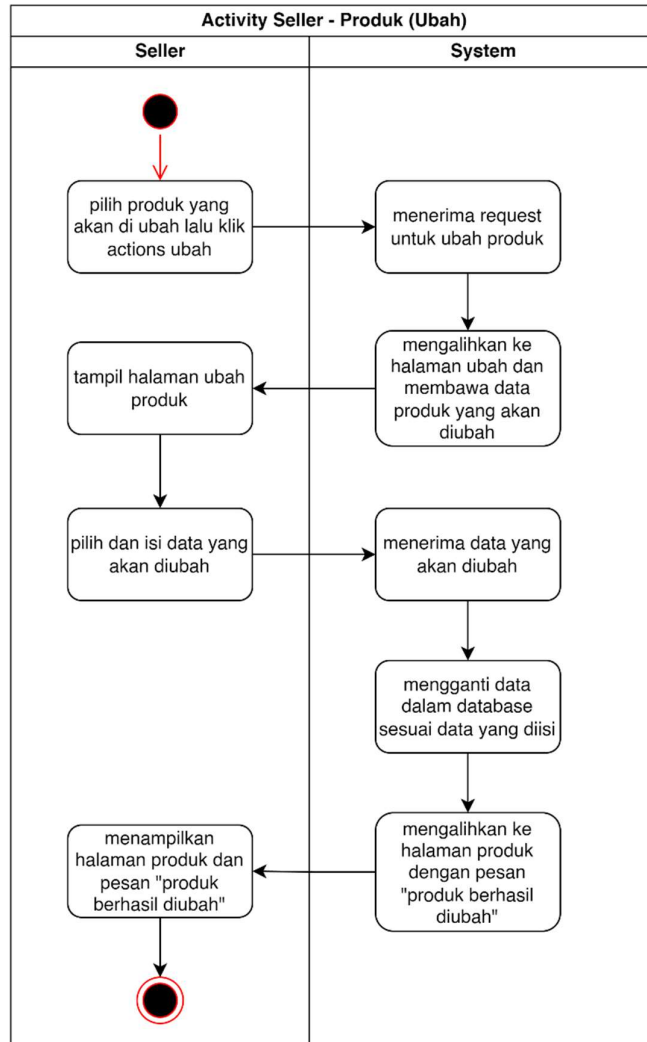
Gambar 3.14 *Activity Pesanan Seller*

Pada gambar 3.14 merupakan alur *activity* pada saat melakukan konfirmasi pesanan dan membuat faktur jika *customer* tersebut setuju melakukan pembelian. Namun, jika *customer* tidak setuju dalam pembelian produk tersebut akan dibatalkan.



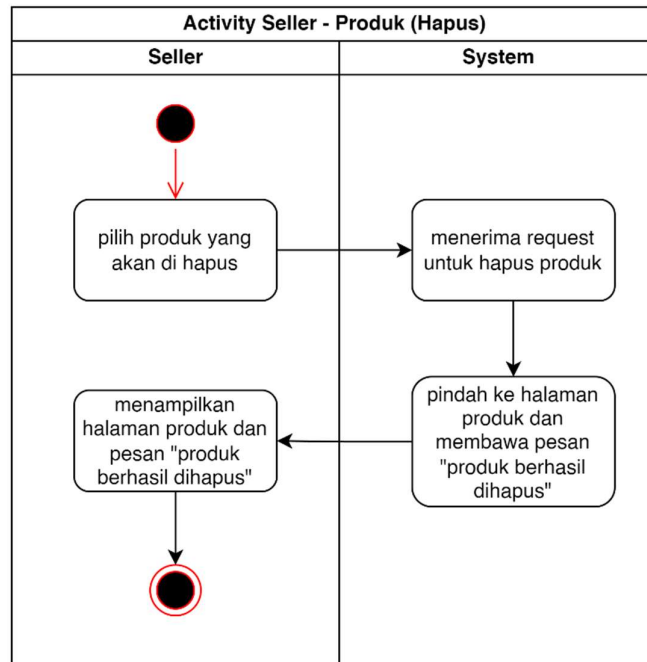
Gambar 3.15 Activity Produk (Tambah) Seller

Pada gambar 3.15 merupakan *activity* ketika melakukan tambah produk yang ingin ditambahkan dengan beberapa validasi di sistem agar tidak terjadi kesalahan *input* atau data yang di *input* kurang.



Gambar 3.16 *Activity* Produk (Ubah) *Seller*

Pada gambar 3.16 merupakan *activity* untuk melakukan ubah data ketika ingin mengganti foto, teks, gambar atau harga.



Gambar 3.17 *Activity* Produk (Hapus) *Seller*

Pada gambar 3.17 merupakan *activity* untuk melakukan hapus data yang tidak sesuai atau data salah.

3.3.4. Metode Pengembangan

Metode *Scrum* adalah metode pengembangan perangkat lunak yang menggunakan prinsip *Agile* untuk menjalankan perulangan. *Scrum* dipercaya dapat membantu memecahkan masalah yang kompleks dan menghasilkan produk berkualitas tinggi. Selanjutnya, metode *Scrum* juga didefinisikan sebagai kerangka kerja untuk mengelola pengembangan produk aplikasi dari proyek perangkat lunak[29].

3.3.4.1. *Product Backlog*

Pada *Product Backlog* merupakan *Product Backlog* yang tercipta dari kebutuhan pengguna selama fase pengumpulan data. Persyaratan pengguna untuk *website* penjualan produk meliputi umpan balik pengguna dan pengembang perangkat lunak dari kuesioner.

3.3.4.2. *Sprint*

Setelah menentukan *Product Backlog*, waktu pengerjaan pengembangan sistem dibagi menjadi beberapa *Sprint*. Selama *Sprint*, *Product Backlog* dievaluasi. Tujuan dari setiap fitur sesuai dengan keinginan *Product Owner*. Dengan demikian di setiap perbedaan, tampilan dan sistem diperbaiki berdasarkan hasil review dengan *Product Owner*.

3.3.4.3. *Sprint Planning*

Sprint Planning adalah strategi yang terdapat dalam *Sprint Backlog* untuk setiap *Sprint*. Perencanaan *Sprint* akan menetapkan batas waktu untuk setiap *Sprint*. Fase ini juga akan mengidentifikasi *Sprint Goals* yang dapat digunakan sebagai panduan bagi pengembang untuk mengembangkan *Increments*

3.3.4.4. *Daily Scrum*

Fase ini untuk memantau laporan kemajuan dari setiap pekerjaan yang telah dilakukan pengembang. *Scrum* harian akan dilakukan setiap 24 jam, yang akan menentukan progres dari setiap *Product Backlog* yang diselesaikan.

3.3.4.5. *Sprint Review*

Setelah fase *Sprint* selesai, aplikasi ditinjau untuk kesesuaian dengan *Increment* dan perubahan dilakukan pada *Product Backlog* jika diperlukan. Berdasarkan hasil *review*, akan dilakukan penyesuaian terhadap *Product Backlog*, apakah diperlukan perubahan terhadap *Product Backlog* pada *Sprint* yang telah selesai.

3.3.4.6. *Sprint Retrospective*

Tujuan fase ini adalah untuk mendapatkan umpan balik dari persyaratan fungsional yang ditinjau. Hasil *Sprint Retrospective* berpengaruh dalam mempertimbangkan

apakah perubahan perlu dilakukan pada *Product Backlog* yang ditujukan pada *Sprint* mendatang.

3.3.4.7. *Product*

Pada tahapan ini adalah produk yang dihasilkan yang sudah dirancang sebelumnya, lalu produk tersebut adalah menghasilkan *website E-Commerce* penjualan *Source Code* dengan nama Gencode berbasis *website*.

3.3.5. Pengujian Sistem

Black Box Testing akan dilakukan menguji coba fungsi-fungsi fitur yang ada pada *website* Gencode. Memastikan setiap fitur sesuai dengan harapan.

3.3.6. Penarikan Kesimpulan

Proses terakhir adalah laporan akhir, yang mengumpulkan kesimpulan dan rekomendasi setelah beberapa tahapan pengembangan aplikasi. Kesimpulan ini didasarkan pada hasil pengujian *website* yang berhasil dan analisis fungsionalitas dari *website* penjualan *Source Code* yang dirancang. Selain itu, memperbanyak masukan terkait penelitian yang dapat digunakan untuk penelitian selanjutnya di kemudian hari.