

**RANCANG BANGUN *BACKEND* APLIKASI *POINT OF SALES*  
BERBASIS WEB MENGGUNAKAN GOLANG  
(STUDI KASUS: PT CAKRA RADHA MUSTIKA)**

**(Skripsi)**

**Oleh**

**RACHMAT AGUNG ANANDA**



**PROGRAM STUDI TEKNIK INFORMATIKA  
JURUSAN TEKNIK ELEKTRO  
FAKULTAS TEKNIK  
UNIVERSITAS LAMPUNG  
2023**

**RANCANG BANGUN *BACKEND* APLIKASI *POINT OF SALES*  
BERBASIS WEB MENGGUNAKAN GOLANG  
(STUDI KASUS: PT CAKRA RADHA MUSTIKA)**

**Oleh  
RACHMAT AGUNG ANANDA**

**Skripsi**

**Sebagai Salah Satu Syarat untuk Mencapai Gelar  
SARJANA TEKNIK**

**Pada  
Program Studi Teknik Informatika  
Fakultas Teknik**



**PROGRAM STUDI TEKNIK INFORMATIKA  
JURUSAN TEKNIK ELEKTRO  
FAKULTAS TEKNIK  
UNIVERSITAS LAMPUNG  
2023**

## ABSTRAK

### RANCANG BANGUN BACKEND APLIKASI POINT OF SALES BERBASIS WEB MENGGUNAKAN GOLANG (STUDI KASUS: PT CAKRA RADHA MUSTIKA)

Oleh

**RACHMAT AGUNG ANANDA**

Dalam era bisnis yang terus berkembang, teknologi informasi menjadi kunci utama dalam meningkatkan efisiensi operasional perusahaan. PT Cakra Radha Mustika membantu perusahaan lain dengan pendekatan terintegrasi. Mitra bisnis mereka yaitu Kalcare menghadapi tantangan dalam mengelola penjualan dan inventaris yang juga dikenal sebagai *Point of Sales* (POS). Penelitian ini bertujuan untuk merancang dan membangun sebuah aplikasi berbasis web yang dapat mempermudah pengelolaan penjualan dan inventaris di Kalcare. Dalam penelitian ini, metode *Agile Scrum* digunakan untuk mengembangkan aplikasi secara cepat dan terstruktur. Pada tahap *sprint product backlog*, kebutuhan sistem diidentifikasi melalui *User Stories*. Kemudian, dilakukan *sprint planning meeting* untuk merancang sistem berdasarkan analisis kebutuhan menggunakan *Unified Modeling Language* (UML) dengan bantuan *Use Case Diagram*, *Activity Diagram*, dan *Entity Relationship Diagram*. Selanjutnya, tahap *sprint backlog* digunakan untuk mengimplementasikan kode program dengan bahasa pemrograman Golang. Terakhir, dilakukan pengujian aplikasi menggunakan aplikasi Postman dengan mengirimkan parameter dan memeriksa respons sesuai yang diharapkan. Hasil dari penelitian ini adalah berhasilnya pengembangan 12 fitur pada aplikasi *Point of Sales*, yang telah diuji dengan hasil yang sesuai dengan harapan.

Kata kunci: Teknologi Informasi, Efisiensi Operasional, *Point of Sales*, *Agile Scrum*, Manajemen Inventaris

## **ABSTRACT**

### **DESIGN AND DEVELOPMENT A WEB-BASED POINT OF SALES APPLICATION BACKEND USING GOLANG (CASE STUDY: PT CAKRA RADHA MUSTIKA)**

**By**

**RACHMAT AGUNG ANANDA**

*In an era of business that continues to develop, information technology is the main key in improving company operational efficiency. PT Cakra Radha Mustika helps other companies with an integrated approach. Their business partner, Kalcare, faces challenges in managing sales and inventory, also known as Point of Sales (POST). This research aims to design and build a web-based application that can simplify sales and inventory management at Kalcare. In this research, the Agile Scrum method is used to develop applications quickly and in a structured manner. At the stage of the sprint product backlog, system requirements are identified through User Stories. Then, it's a sprint planning meeting to design a system based on needs analysis using Unified Modeling Language (UML) with help Use Case Diagram, Activity Diagram, and Entity Relationship Diagram. Next, stage sprint backlog used to implement program code with the Golang programming language. Finally, the application was tested using the Postman application by sending parameters and checking the response as expected. The result of this research is the successful development of 12 features in the application Point of Sales, which has been tested with expected results.*

*Keywords: Information Technology, Operational Efficiency, Point of Sales, Agile Scrum, Inventory Management*



Judul Skripsi

: **RANCANG BANGUN *BACKEND* APLIKASI  
POINT OF SALES BERBASIS WEB  
MENGUNAKAN GOLANG (STUDI KASUS:  
PT CAKRA RADHA MUSTIKA)**

Nama Mahasiswa

: **Rachmat Agung Ananda**

Nomor Pokok Mahasiswa : 1915061029

Program Studi

: Teknik Informatika

Fakultas

: Teknik




**MENYETUJUI**

1. Komisi Pembimbing

Pembimbing Utama


Pembimbing Pendamping

  
**Wahyu Eko Sulistiono, S.T., M.Sc.**  
NIP 19741201 200112 1 001

  
**Resty Annisa, S.ST., M.Kom.**  
NIP 19900830 201903 2 019

2. Mengetahui

Ketua Jurusan Teknik Elektro

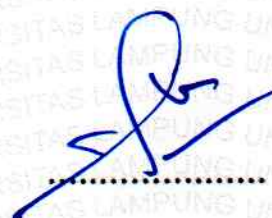
  
**Herlinawati, S.T., M.T.**  
NIP 19710314 199903 2 001



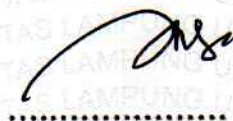
## MENGESAHKAN

### 1. Tim Penguji

**Ketua : Wahyu Eko Sulistiono, S.T., M.Sc.**



**Sekretaris : Resty Annisa, S.ST., M.Kom.**




**Penguji : Yessi Mulyani, S.T., M.T.**



### 2. Dekan Fakultas Teknik



**Dr. Eng. Ir. Helmy Fitriawan, S.T., M.Sc. j**  
NIP 19750928 200112 1 002



**Tanggal Lulus Ujian Skripsi : 20 Oktober 2023**

## SURAT PERNYATAAN

Saya yang bertandatangan di bawah ini, menyatakan bahwa skripsi saya dengan judul "Rancang Bangun *Backend* Aplikasi *Point Of Sales* Berbasis Web Menggunakan Golang (Studi Kasus: PT. Cakra Radha Mustika)" dibuat oleh saya sendiri. Semua hasil yang tertuang dalam skripsi ini telah mengikuti kaidah penulisan karya ilmiah Universitas Lampung. Apabila di kemudian hari terbukti bahwa skripsi ini merupakan salinan atau dibuat oleh oranglain, maka saya bersedia menerima sanksi sesuai dengan ketentuan hukum atau akademik yang berlaku.

Bandar Lampung, 30 Oktober 2023

Pembuat Pernyataan



Rachmat Agung Ananda

NPM. 1915061029

## RIWAYAT HIDUP



Penulis dilahirkan di Lampung, pada tanggal 19 Februari 2002. Penulis merupakan anak kedua dari pasangan Bapak Jusman Gumanti dan Ibu Juriah. Penulis menyelesaikan pendidikannya di SD Al Azhar 1 Bandar Lampung pada tahun 2013, SMP Negeri 22 Bandar Lampung pada tahun 2016, dan SMA Negeri 9 Bandar Lampung pada tahun 2019. Pada tahun 2019 penulis terdaftar sebagai mahasiswa Program Studi Teknik Informatika, Jurusan Teknik Elektro, Fakultas Teknik Universitas Lampung melalui jalur SBMPTN. Selama menjadi mahasiswa, penulis melakukan beberapa kegiatan, antara lain:

1. Menjadi anggota biasa Himpunan Mahasiswa Teknik Elektro Universitas Lampung, Departemen Pengembangan Keteknikan, Divisi Pengabdian Masyarakat pada tahun 2019.
2. Menjadi anggota biasa Himpunan Mahasiswa Teknik Elektro Universitas Lampung, Departemen Pengembangan Keteknikan, Divisi Pengabdian Masyarakat pada tahun 2020.
3. Melaksanakan Kuliah Kerja Nyata di Kelurahan Sukarame, Kecamatan Sukarame, Kota Bandar Lampung, Provinsi Lampung pada bulan Januari sampai dengan Februari 2022.
4. Mengikuti program Studi Independen Kampus Merdeka dari Kementerian Pendidikan dan Budaya dengan mengambil kelas *Full Stack Engineering* di PT Marka Kreasi Persada pada tahun 2021 dan kelas *Back End Engineering Bootcamp* di PT Ruang Raya Indonesia pada tahun 2022.
5. Melakukan kerja praktik di PT. Queen Network Nusantara sebagai *Front End Developer* pada tahun 2022.

## **MOTTO**

“Allah tidak membebani seseorang melainkan sesuai dengan kesanggupannya...”

**(Q.S. Al-Baqarah: 286)**

“Dan bersabarlah kamu, sesungguhnya janji Allah adalah benar”

**(Q.S. Ar Rum: 60)**

“Tidak mustahil bagi orang biasa untuk memutuskan menjadi luar biasa”

**(Elon Musk)**

“Jangan selalu membayangkan hasil yang besar, mulailah berusaha walaupun hasilnya kecil. Walaupun sedikit asal nyata hasilnya”

**(Penulis)**

## PERSEMBAHAN

Bismillaahirrohmaanirrahim,

Dengan mengharapkan ridho dari Allah SWT,

Kupersembahkan karyaku ini untuk orang-orang yang kusayangi

dengan setulus hati:

Orangtua tercinta,

Keluargaku,

Teman-Temanku,

Dan

Orang-orang yang telah membantu hidupku

Terimakasih untuk segalanya,

Kalian adalah hartaku yang paling berharga

## SANWACANA

Puji syukur kehadiran Allah SWT yang telah melimpahkan rahmat dan hidayat-Nya sehingga penulis dapat menyelesaikan penyusunan skripsi/tugas akhir ini dengan judul “Rancang Bangun Backend Aplikasi Point of Sales Berbasis Web Menggunakan Golang (Studi Kasus: PT Cakra Radha Mustika)”. Dalam pelaksanaan dan pembuatan skripsi/tugas akhir ini penulis menerima dukungan baik secara moril maupun materil yang sangat berharga dari berbagai pihak. Oleh karena itu, penulis ingin mengucapkan terima kasih kepada semua pihak yang telah membantu, khususnya kepada:

1. Kedua orangtuaku, Bapak Jusman Gumanti dan Ibu Juriah tercinta serta seluruh keluarga penulis yang tidak hentinya mendo'akan serta memberikan dorongan semangat dan materi;
2. Bapak Dr. Eng. Helmy Fitriawan, S.T., M.Sc., selaku Dekan Fakultas Teknik Universitas Lampung;
3. Ibu Herlinawati, S.T., M.T. selaku Ketua Jurusan Teknik Elektro Universitas Lampung;
4. Ibu Yessi Mulyani, S.T., M.T. selaku selaku Penguji yang telah membantu proses kelancaran pengerjaan penelitian dan memberikan banyak saran serta masukan terhadap penelitian ini;
5. Bapak Wahyu Eko Sulistiono, S.T., M.Sc selaku Pembimbing Utama yang selalu meluangkan waktunya untuk memberikan bimbingan dan dukunganserta memudahkan penulis dalam menyelesaikan penelitian ini;
6. Ibu Resty Annisa, S. ST., M. Kom, selaku Pembimbing Pendamping yang selalu memberikan dukungan serta bimbingan agar menjadi lebih baik;

7. Bapak Ir. Meizano Ardhi Muhammad, S.T., M.T. selaku Pembimbing Akademik yang selalu memberikan bimbingan dan dukungan kepada penulis selama menyelesaikan perkuliahan.
8. Mbak Rika selaku Admin Program Studi Teknik Informatika yang telah banyak membantu penulis dalam segala urusan administrasi selama perkuliahan;
9. Seluruh dosen dan staf Jurusan Teknik Informatika Unila yang memberi masukan dan mempermudah proses pembuatan skripsi / tugas akhir ini.
10. Squad CRM, mas dendi, mas bob, mba inas, bia, hafif, wisnu, toni, verel dan hafid, selaku tim pengembangan yang selalu memberi dukungan dan membantu proses pengembangan aplikasi hingga tidak ada bug yang tersisa
11. Winona Maharani yang selalu memberi dukungan dengan tulus sehingga penulis dapat menyelesaikan skripsi ini hingga tuntas.
12. Sahabat Laut, Sanjaya Ilham Kurnia, M. Iqbal Syahbana, Surya Saputra, M. Irfan Fadhila, Royyan Fajrul Falah, Ramadhani Asrofa Kemal, M. Naufal Ashiddiq, Nabila Tusyifa, Presillia Br Sianturi, Rachel Adella Ferninda, Ridha Aisyiyah Ryandani, dan Tania Argadini yang selalu memberikan saran, dukungan, dan bantuan kepada penulis selama menjalani perkuliahan.
13. Keluarga besar Teknik Elektro Angkatan 2019 yang telah menjadi teman seperjuangan sejak mahasiswa baru. Terimakasih telah mewarnai masa perkuliahan penulis.

Penulis berharap agar laporan ini dapat menjadi referensi bagi pengembangan keilmuan di bidang teknik informatika. Oleh karena itu, semoga penelitian ini bermanfaat bagi yang membacanya.

Bandar Lampung, Oktober 2023

Penulis,

Rachmat Agung Ananda



## DAFTAR ISI

	Halaman
<b>PERSEMBAHAN .....</b>	<b>i</b>
<b>SANWACANA.....</b>	<b>ii</b>
<b>DAFTAR ISI.....</b>	<b>iv</b>
<b>DAFTAR GAMBAR.....</b>	<b>vii</b>
<b>DAFTAR TABEL.....</b>	<b>xii</b>
<b>I. PENDAHULUAN.....</b>	<b>1</b>
1.1 Latar Belakang .....	1
1.2 Tujuan Penelitian.....	3
1.3 Rumusan Masalah .....	3
1.4 Batasan Masalah.....	3
1.5 Sistematika Penulisan .....	4
<b>II. TINJAUAN PUSTAKA .....</b>	<b>6</b>
2.1 <i>Point of Sales</i> (POS) .....	6
2.2 PT Cakra Radha Mustika .....	6
2.3 Sistem Informasi .....	7
2.4 <i>Website</i> .....	8
2.5 <i>Backend</i> .....	8
2.6 <i>Application Programming Interface</i> (API) .....	8
2.7 <i>Database</i> .....	9
2.8 PostgreSQL .....	9
2.9 Golang .....	9

2.10	<i>Visual Studio Code</i> .....	10
2.11	<i>Postman</i> .....	10
2.12	<i>Agile Software Development Methods</i> .....	11
2.13	<i>Unified Modelling Language</i> .....	11
2.14	<i>Framework Fiber</i> .....	12
2.15	Google Cloud Platform .....	12
2.16	<i>Clean Architecture</i> .....	13
2.17	Penelitian Terdahulu.....	14
<b>III. METODOLOGI PENELITIAN .....</b>		<b>19</b>
3.1	Waktu dan Tempat.....	19
3.2	Alat dan Bahan Penelitian.....	19
3.2.1	Alat Penelitian.....	19
3.2.2	Bahan Penelitian.....	20
3.3	Tahapan Penelitian .....	20
3.3.1	<i>Product Backlog</i> .....	21
3.3.2	Sprint Planning Meeting .....	24
3.3.3	<i>Sprint Backlog</i> .....	48
3.3.4	<i>Daily Standup</i> .....	49
3.3.5	<i>Finished Work</i> .....	50
<b>IV. HASIL DAN PEMBAHASAN .....</b>		<b>51</b>
4.1	Hasil .....	51
4.1.1	Pengkodean ( <i>Coding</i> ).....	51
4.1.2	Pengujian ( <i>Testing</i> ).....	133
4.2	Pembahasan.....	151
4.2.1	Metode <i>Agile Scrum</i> .....	151
4.2.2	Sistem Informasi <i>Point of Sales</i> PT Cakra Radha Mustika.....	152

<b>V.</b>	<b>SIMPULAN DAN SARAN .....</b>	<b>157</b>
5.1	Simpulan .....	157
5.2	Saran.....	157
	<b>DAFTAR PUSTAKA .....</b>	<b>158</b>
	<b>LAMPIRAN.....</b>	<b>162</b>

## DAFTAR GAMBAR

Gambar 2.1 Diagram Stuktur <i>Clean Architecture</i> .....	14
Gambar 3.1 Tahapan Penelitian .....	20
Gambar 3.2 <i>Product Backlog</i> .....	25
Gambar 3.3 <i>Use Case Diagram</i> Setelah Pengguna Login .....	26
Gambar 3.4 <i>Activity Diagram Login</i> .....	27
Gambar 3.5 <i>Activity Diagram Logout</i> .....	27
Gambar 3.6 <i>Activity Diagram</i> Pembuatan Pesanan .....	29
Gambar 3.7 <i>Activity Diagram</i> Melihat Transaksi .....	30
Gambar 3.8 <i>Activity Diagram</i> Pembuatan Settlement .....	31
Gambar 3.9 <i>Activity Diagram</i> Melihat Promo .....	32
Gambar 3.10 <i>Activity Diagram</i> Mengatur Promo .....	33
Gambar 3.11 <i>Activity Diagram</i> Melihat Stok.....	34
Gambar 3.12 <i>Activity Diagram</i> Pembuatan <i>Purchase Order</i> .....	35
Gambar 3.13 <i>Activity Diagram</i> Pembuatan <i>Good Receipt Note</i> .....	36
Gambar 3.14 <i>Activity Diagram</i> Melihat <i>Other Issue</i> .....	37
Gambar 3.15 <i>Activity Diagram</i> Mengatur <i>Other Issue</i> .....	38
Gambar 3.16 <i>Activity Diagram</i> Mengatur <i>Brand</i> .....	39
Gambar 3.17 <i>Activity Diagram</i> Mengatur Produk .....	41
Gambar 3.18 <i>Activity Diagram</i> Mengatur Kategori <i>Channel</i> .....	42
Gambar 3.19 <i>Activity Diagram</i> Mengatur <i>Channel</i> .....	43
Gambar 3.20 <i>Activity Diagram</i> Mengatur Administrasi Karyawan.....	45
Gambar 3.21 <i>Entity Relationship Diagram</i> Dengan <i>Foreign Key</i> .....	46
Gambar 3.22 <i>Entity Relationship Diagram</i> Tanpa <i>Foreign Key</i> .....	47
Gambar 3.23 <i>Entity Relationship Diagram</i> Master Data <i>Global</i> .....	48
Gambar 4.1 Implementasi <i>Repository Save Token</i> .....	51

Gambar 4.2 Implementasi <i>Usecase Login</i> .....	52
Gambar 4.3 Implementasi <i>Usecase Logout</i> .....	54
Gambar 4.4 Implementasi <i>Handler Login</i> .....	55
Gambar 4.5 Implementasi <i>Handler Logout</i> .....	56
Gambar 4.6 Implementasi <i>Repository List Users</i> .....	57
Gambar 4.7 Implementasi <i>Repository Create User</i> .....	58
Gambar 4.8 Implementasi <i>Repository Get By Id Users</i> .....	59
Gambar 4.9 Implementasi <i>Repository Update Users</i> .....	60
Gambar 4.10 Implementasi <i>Repository Delete User</i> .....	61
Gambar 4.11 Implementasi <i>Repository Create User Role</i> .....	62
Gambar 4.12 Implementasi <i>Usecase List Users</i> .....	63
Gambar 4.13 Implementasi <i>Usecase Create User</i> .....	64
Gambar 4.14 Implementasi <i>Usecase Get By Id Users</i> .....	65
Gambar 4.15 Implementasi <i>Usecase Update Users</i> .....	67
Gambar 4.16 Implementasi <i>Usecase Delete User</i> .....	69
Gambar 4.17 Implementasi <i>Usecase Create User Role</i> .....	70
Gambar 4.18 Implementasi <i>Handler List Users</i> .....	71
Gambar 4.19 Implementasi <i>Handler Create Users</i> .....	72
Gambar 4.20 Implementasi <i>Handler Get By Id Users</i> .....	73
Gambar 4.21 Implementasi <i>Handler Update Users</i> .....	74
Gambar 4.22 Implementasi <i>Handler Delete User</i> .....	75
Gambar 4.23 Implementasi <i>Handler Create User Role</i> .....	76
Gambar 4.24 Implementasi <i>Repository List Product</i> .....	77
Gambar 4.25 Implementasi <i>Repository Check Stok</i> .....	78
Gambar 4.26 Implementasi <i>Repository List Product Cart</i> .....	79
Gambar 4.27 Implementasi <i>Repository List Promotion</i> .....	80
Gambar 4.28 Implementasi <i>Repository List Payment Method</i> .....	82
Gambar 4.29 Implementasi <i>Usecase List Product</i> .....	83
Gambar 4.30 Implementasi <i>Usecase Check Stok</i> .....	84
Gambar 4.31 Implementasi <i>Usecase List Product Cart</i> .....	85
Gambar 4.32 Implementasi <i>Usecase List Promotion</i> .....	86
Gambar 4.33 Implementasi <i>Usecase List Payment Method</i> .....	87

Gambar 4.34 Implementasi <i>Handler List Product</i> .....	89
Gambar 4.35 Implementasi <i>Handler Check Stok</i> .....	90
Gambar 4.36 Implementasi <i>Handler List Product Cart</i> .....	92
Gambar 4.37 Implementasi <i>Handler List Promotion</i> .....	93
Gambar 4.38 Implementasi <i>Handler List Payment Method</i> .....	93
Gambar 4.39 Implementasi <i>Handler Create Order</i> .....	94
Gambar 4.40 Implementasi <i>Repository Create Channel Product</i> .....	95
Gambar 4.41 Implementasi <i>Repository List Channel Product</i> .....	97
Gambar 4.42 Implementasi <i>Repository Get By Id Channel Product</i> .....	99
Gambar 4.43 Implementasi <i>Repository Update Channel Product</i> .....	101
Gambar 4.44 Implementasi <i>Repository Delete Channel Product</i> .....	101
Gambar 4.45 Implementasi <i>Usecase List Channel Product</i> .....	102
Gambar 4.46 Implementasi <i>Usecase Create Channel Product</i> .....	104
Gambar 4.47 Implementasi <i>Usecase Get By Id Channel Product</i> .....	106
Gambar 4.48 Implementasi <i>Usecase Delete Channel Product</i> .....	107
Gambar 4.49 Implementasi <i>Handler Create Channel Product</i> .....	108
Gambar 4.50 Implementasi <i>Handler List Channel Product</i> .....	109
Gambar 4.51 Implementasi <i>Handler Get By Id Channel</i> .....	110
Gambar 4.52 Implementasi <i>Handler Update Channel</i> .....	111
Gambar 4.53 Implementasi <i>Handler Delete Channel Product</i> .....	112
Gambar 4.54 Implementasi <i>Repository List Transaction</i> .....	114
Gambar 4.55 Implementasi <i>Repository Get By Id Transaction</i> .....	115
Gambar 4.56 Implementasi <i>Repository Transaction Settle</i> .....	117
Gambar 4.57 Implementasi <i>Repository List Report Transaction One Date</i> .....	117
Gambar 4.58 Implementasi <i>Repository List Report Transactions Many Date</i> ....	118
Gambar 4.59 Implementasi <i>Usecase List Transaction</i> .....	119
Gambar 4.60 Implementasi <i>Usecase Get By Id Transaction</i> .....	121
Gambar 4.61 Implementasi <i>Usecase Transaction Settle</i> .....	122
Gambar 4.62 Implementasi <i>Usecase Transaction Cancel</i> .....	124
Gambar 4.63 Implementasi <i>Usecase List Report Transaction</i> .....	126
Gambar 4.64 Implementasi <i>Handler List Transaction</i> .....	127
Gambar 4.65 Implementasi <i>Handler Get By Id Transaction</i> .....	129

Gambar 4.66 Implementasi <i>Handler Transaction Settle</i> .....	130
Gambar 4.67 Implementasi <i>Handler Transaction Cancel</i> .....	131
Gambar 4.68 Implementasi <i>Handler List Report Transaction</i> .....	132
Gambar 4.69 <i>Response Login</i> .....	133
Gambar 4.70 <i>Response Logout</i> .....	134
Gambar 4.71 <i>Response List Users</i> .....	135
Gambar 4.72 <i>Response Create User</i> .....	136
Gambar 4.73 <i>Response Get By Id User</i> .....	137
Gambar 4.74 <i>Response Update User</i> .....	138
Gambar 4.75 <i>Response Delete User</i> .....	139
Gambar 4.76 <i>Response List Channel</i> .....	139
Gambar 4.77 <i>Response Create Channel Product</i> .....	140
Gambar 4.78 <i>Response Get By Id Channel</i> .....	141
Gambar 4.79 <i>Response List Order</i> .....	142
Gambar 4.80 <i>Response Promotion</i> .....	143
Gambar 4.81 <i>Response Payment Method</i> .....	144
Gambar 4.82 <i>Response Create Order</i> .....	145
Gambar 4.83 <i>Response List Transaction</i> .....	146
Gambar 4.84 <i>Response Report Transaction</i> .....	147
Gambar 4.85 <i>Response Get By Id Transaction</i> .....	148
Gambar 4.86 <i>Response Transaction Settlement</i> .....	149
Gambar 4.87 <i>Response Transaction Cancel</i> .....	150
Gambar Lampiran 1.1 Kode Fitur <i>Create Order</i> .....	167
Gambar Lampiran 2.1 Tampilan Halaman <i>Order</i> .....	168
Gambar Lampiran 2.2 Tampilan Halaman <i>Transaction</i> .....	168
Gambar Lampiran 2.3 Tampilan Halaman <i>Transaction Detail</i> .....	169
Gambar Lampiran 2.4 Tampilan Halaman <i>Transaction Detail Invoice</i> .....	169
Gambar Lampiran 2.5 Tampilan Halaman <i>Transaction Detail Receipt</i> .....	170
Gambar Lampiran 2.6 Tampilan Halaman <i>Stock List</i> .....	170
Gambar Lampiran 2.7 Tampilan Halaman <i>Purchase Order</i> .....	171
Gambar Lampiran 2.8 Tampilan Halaman <i>Goods Receipt Note</i> .....	171
Gambar Lampiran 2.9 Tampilan Halaman <i>Brand</i> .....	172

Gambar Lampiran 2.10 Tampilan Halaman <i>Product List</i> .....	172
Gambar Lampiran 2.11 Tampilan Halaman <i>Promotion</i> .....	173
Gambar Lampiran 2.12 Tampilan Halaman <i>Channel</i> .....	173
Gambar Lampiran 2.13 Tampilan Halaman <i>Administration</i> .....	174
Gambar Lampiran 2.14 Tampilan Halaman <i>Other Issue</i> .....	174



**DAFTAR TABEL**

Tabel 3.1 Jadwal Penelitian.....	19
Tabel 3.2 Alat Penelitian .....	19
Tabel 3.3 <i>User Story</i> .....	21

## I. PENDAHULUAN

### 1.1 Latar Belakang

Dalam era bisnis yang terus berkembang, teknologi informasi telah menjadi elemen kunci dalam mengelola operasi perusahaan dengan lebih efisien dan efektif. Ketergantungan pada teknologi telah meningkat secara signifikan, terutama dalam konteks manajemen penjualan dan inventaris. Di tengah persaingan bisnis yang ketat, perusahaan-perusahaan mencari cara untuk memaksimalkan produktivitas, mengurangi biaya operasional, dan memberikan pengalaman pelanggan yang unggul.

Perusahaan PT Cakra Radha Mustika, berperan sebagai *Total Business Solution provider*, berfokus membantu perusahaan lain menjalankan proyek atau kampanye dari awal hingga akhir. Namun, dalam upaya menciptakan ekosistem yang terintegrasi, perusahaan ini menghadapi permasalahan spesifik dalam pengelolaan penjualan dan inventaris.

Tantangan yang dihadapi oleh PT Cakra Radha Mustika muncul dalam bentuk kebutuhan akan pengelolaan yang lebih efisien, pemantauan inventaris yang akurat, dan analisis data yang lebih mendalam untuk mendukung pengambilan keputusan bisnis yang lebih baik. Salah satu mitra bisnis mereka, yaitu Kalcare, memiliki akar sejarah yang kuat dalam perjuangan untuk memberikan akses kesehatan yang lebih baik bagi masyarakat Indonesia.

Kalcare merupakan bagian dari keluarga perusahaan Kalbe Farma yang berkomitmen untuk memberikan akses kesehatan yang terjangkau kepada masyarakat Indonesia. Awalnya didirikan oleh dr. Boenjamin Setiawan, Kalcare

telah berkembang dari sebuah garasi kecil menjadi perusahaan yang beroperasi di wilayah Asia Tenggara.

Salah satu inisiatif utama Kalcare adalah KlikDokter, platform digital terintegrasi yang memungkinkan individu untuk melakukan konsultasi online dengan dokter, mendapatkan resep, dan menggunakan alat kesehatan berbasis teknologi. Kalcare juga terus berinovasi dalam menyediakan solusi kesehatan yang komprehensif, termasuk layanan obat-obatan, produk kesehatan, kecantikan, serta teknologi kesehatan untuk menjaga kesehatan secara berkelanjutan.

Untuk mengatasi masalah yang dihadapi oleh Kalcare, PT Cakra Radha Mustika telah mengambil langkah untuk mengembangkan website POS yang sesuai dengan kebutuhan spesifik bisnis Kalcare. Kolaborasi ini menciptakan peluang untuk memahami bagaimana integrasi sistem POS berbasis Golang dengan kebutuhan spesifik bisnis Kalcare dapat menjadi studi kasus yang berharga. Dengan implementasi sistem *Point of Sales* (POS) berbasis Golang yang tepat, solusi yang diberikan oleh PT Cakra Radha Mustika diantaranya Otomatisasi Proses Penjualan, Sistem POS akan mengotomatisasi proses penjualan, termasuk pencatatan pesanan, pembayaran, dan pencatatan inventaris. Hal ini akan menghemat waktu dan mengurangi kesalahan manusia, pemantauan Inventaris *Real-Time*, Sistem POS akan memungkinkan Kalcare untuk memantau inventaris mereka secara *real-time*. Ini akan membantu mereka mengelola stok dengan lebih baik dan menghindari masalah seperti kehabisan stok atau stok berlebih, dan analisis Data yang Lebih Mendalam, Sistem POS akan mengumpulkan data penjualan yang berharga, yang dapat digunakan untuk analisis bisnis yang lebih mendalam. Ini akan membantu Kalcare dalam pengambilan keputusan yang lebih baik untuk mengoptimalkan operasional mereka.

Dalam konteks pengembangan sistem POS, penggunaan bahasa pemrograman Golang memiliki beberapa kelebihan yang signifikan diantaranya, berkinerja tinggi, efisiensi penggunaan memori, memiliki pustaka yang kuat yang digunakan dalam

berbagai fungsi balikan data serta dapat mudah untuk dimodifikasi dalam penggunaan struktur folder nya dengan penggunaan package sebagai kuncinya.

Pada implementasi sistem POS berbasis Golang untuk Kalcare, proyek ini telah dibagi menjadi dua komponen utama yaitu *frontend* dan *backend*. Pembagian ini memungkinkan pengembangan aplikasi menjadi lebih terstruktur dan efisien, Dalam proyek ini, penulis telah memfokuskan perannya pada pengembangan *backend* atau sisi server. Ini mencakup merancang dan mengimplementasikan berbagai fitur sesuai dengan kebutuhan bisnis yang sudah ditentukan

Dengan demikian, implementasi sistem POS berbasis Golang akan membantu Kalcare mengatasi masalah pengelolaan penjualan dan inventaris mereka, meningkatkan efisiensi operasional, dan memberikan pengalaman pelanggan yang lebih baik. Studi kasus ini akan memberikan wawasan berharga tentang bagaimana teknologi informasi, khususnya POS berbasis Golang, dapat menjadi alat yang efektif untuk meningkatkan kinerja perusahaan dan memenuhi kebutuhan pelanggan di era digital yang terus berkembang pesat.

## **1.2 Tujuan Penelitian**

Tujuan yang ingin dicapai adalah merancang dan membangun *backend* website *Point of Sales* (POS) menggunakan bahasa pemrograman Golang

## **1.3 Rumusan Masalah**

Rumusan masalah yang dapat diidentifikasi dalam penelitian ini adalah bagaimana merancang dan membangun *backend* website *Point of Sales* (POS).

## **1.4 Batasan Masalah**

Dalam penelitian ini, pembatasan masalah meliputi hal-hal sebagai berikut:

1. Spesifikasi aplikasi berdasarkan permintaan user Kalcare dengan pertimbangan developer PT Cakra Radha Mustika

2. *Output* dari penelitian ini adalah sebuah *backend application programming interface* (API). Penelitian ini tidak menyediakan *output* berupa tampilan antarmuka atau *frontend*

### 1.5 Sistematika Penulisan

Sistematika penulisan skripsi / tugas akhir ini terdiri dari 5 (lima) bab sebagai berikut:

**BAB I : PENDAHULUAN**

Dalam bab ini memuat latar belakang, tujuan penelitian, rumusan masalah, batasan masalah dan sistematika penulisan

**BAB II : TINJAUAN PUSTAKA**

Dalam bab ini memuat penjelasan tentang *point of sales* (POS), PT Cakra Radha Mustika, sistem informasi, *website*, *backend*, *application programming interface* (API), *database*, *postgreSQL*, *golang*, *visual studio code*, *postman*, *agile software development method*, *unified modelling language* (UML), *framework fiber*, *google cloud platform*, *clean architecture* dan penelitian terdahulu

**BAB III : METODOLOGI PENELITIAN**

Dalam bab ini memuat waktu dan tempat penelitian, tahapan penelitian, dan tahapan penelitian secara terperinci, metode yang digunakan adalah metode agile scrum yang terdiri dari *product backlog*, *sprint planning meeting*, *sprint backlog*, *daily standup* yang membahas *sprint review* dan *sprint retrospective*

**BAB IV : HASIL DAN PEMBAHASAN**

Dalam bab ini memuat hasil dan pembahasan yang didapat dalam penelitian ini

**BAB V****: KESIMPULAN DAN SARAN**

Dalam bab ini memuat kesimpulan dari hasil penelitian dan saran-saran mengenai perbaikan dan pengembangan lebih lanjut.

## II. TINJAUAN PUSTAKA

### 2.1 *Point of Sales (POS)*

Sistem POS (*point of sale*) adalah sebuah sistem aplikasi yang diterapkan pada bisnis minimarket ataupun pertokoan untuk menangani pengolahan data transaksi pembelian (*purchases*), transaksi penjualan eceran (*retails*), transaksi retur pembelian (*purchase returns*), dan pelaporan transaksi (*reporting*) yang secara umum penting dibutuhkan dalam pengambilan keputusan strategik oleh para pebisnis swalayan, organisasi, atau perusahaan yang berskala kecil dan menengah. [1]

Setiap POS terdiri dari *hardware* berupa (Terminal/PC, *Receipt Printer*, *Cash Drawer*, Terminal pembayaran, *Barcode Scanner*) dan *software* berupa (*Inventory Management*, Pelaporan, *Purchasing*, *Customer Management*, Standar Keamanan Transaksi, *Return Processing*) dimana kedua komponen tersebut digunakan untuk setiap proses transaksi. *Point of sales* dapat berupa sebuah *checkout counters* dalam sebuah toko atau tempat usaha dimana transaksi penjualan terjadi. [2]

### 2.2 **PT Cakra Radha Mustika**

PT Cakra Radha Mustika adalah perusahaan berbasis *Total Business Solution* yang berfokus pada membantu perusahaan lain dalam mengelola proyek dan kampanye dari awal hingga selesai. Rentang layanan mereka yang komprehensif mencakup membangun dan memperluas kehadiran merek perusahaan untuk mencapai audiens yang lebih luas, mengembangkan strategi generasi prospek yang efektif, menjalankan program akuisisi, dan menjaga inisiatif retensi pelanggan. Untuk menciptakan ekosistem yang sepenuhnya terintegrasi, PT Cakra Radha Mustika memanfaatkan pengalamannya dalam integrasi pelanggan melalui program

keanggotaan, menyediakan *Pusat Layanan Pelanggan* yang didedikasikan dan layanan telemarketing, serta menyediakan *Customer Data Integrators* dan *Orchestrators* yang mampu menggabungkan data dari berbagai sumber. Fokus inti mereka adalah meningkatkan interaksi dan komunikasi dengan pelanggan melalui saluran seperti *WhatsApp*, *Media Sosial*, *SMS*, *Email*, *Telepon*, dan *Aplikasi/Situs Web*. Dengan tujuan visi, PT Cakra Radha Mustika bertujuan untuk menjadi solusi utama dalam Manajemen Hubungan Pelanggan, menawarkan layanan terintegrasi yang meningkatkan proposisi nilai pelanggan dan merek dalam pasar. Misi mereka adalah menjadi integrator utama dan orchestrator dalam pengumpulan dan analisis data, memungkinkan peningkatan nilai pelanggan sepanjang masa. Pada akhirnya, mereka bercita-cita untuk menjadi penyedia solusi CRM Total yang unggul untuk merek, sambil juga berkontribusi pada kesejahteraan masyarakat dengan menyediakan informasi kesehatan dan pendidikan.[3]

### **2.3 Sistem Informasi**

Sistem informasi adalah suatu sistem yang terdiri dari elemen-elemen yang saling terkait dan berinteraksi untuk mengumpulkan, mengolah, menyimpan, mengambil, dan menyebarkan informasi guna mendukung pengambilan keputusan, pengendalian, dan operasi suatu organisasi atau bisnis. Sistem informasi merupakan suatu kombinasi teratur dari orang-orang, *hardware*, *software*, jaringan komunikasi dan sumber daya data yang mengumpulkan, mengubah, dan menyebarkan informasi dalam sebuah organisasi. [4]

Sistem informasi dapat dipahami sebagai sekumpulan subsistem yang saling berhubungan, berkumpul bersama-sama dan membentuk satu kesatuan, saling berinteraksi dan bekerja sama antara bagian satu dengan yang lainnya dengan cara-cara tertentu untuk melakukan fungsi pengolahan data, menerima masukan (*input*) berupa data-data, kemudian mengolahnya (*process*), dan menghasilkan keluaran (*output*) berupa informasi sebagai dasar bagi pengambilan keputusan yang berguna dan mempunyai nilai nyata yang dapat dirasakan akibatnya baik pada saat itu atau dimasa mendatang. [5]



## 2.4 Website

*Website* merupakan kumpulan halaman yang digunakan untuk menampilkan berbagai informasi seperti teks, gambar, animasi, dan suara, baik yang bersifat statis maupun dinamis yang membentuk satu rangkaian yang saling terhubung. [6]

*Website* adalah kumpulan halaman yang memuat informasi berupa data digital berupa teks, gambar, video, audio dan animasi lainnya, yang disampaikan melalui koneksi internet. [7]

Sebuah situs web terdiri dari serangkaian halaman, yang masing-masing memiliki topik terkait dan dapat berisi gambar, video, animasi, atau jenis objek lainnya. [8]

## 2.5 Backend

*Backend* adalah tempat di mana proses pada suatu sistem informasi atau aplikasi berjalan, data dapat ditambahkan, diubah maupun dihapus. *Backend* biasanya mengurus segala jenis proses yang tidak berhubungan langsung dengan pengguna, seperti server dan basis data. *Backend* dibutuhkan dalam pengembangan sistem dan manajemen data pada sistem. [9]

Back-end merupakan program yang berjalan pada sisi server untuk berinteraksi langsung dengan basis data dan melaksanakan proses *logic* dari suatu sistem web. Salah satu teknologi antarmuka aplikasi atau layanan dari suatu program adalah *Application Programming Interface* (API). API untuk layanan web yang menggunakan protokol HTTP adalah *Representational State Transfer* (REST). Penggunaan REST API sebagai *back-end* layanan web memungkinkan layanan web diakses oleh sistem lain tanpa batasan bahasa, *environment*, maupun platform dari sisi *front-end*. [10]

## 2.6 Application Programming Interface (API)

*Application Programming Interface* (API) adalah konsep fitur antarmuka pemrograman aplikasi yang memungkinkan pihak lain untuk mengakses dan menggunakan aplikasi tanpa mengubah struktur kode utama atau basis data sistem

dan memfasilitasi komunikasi antar sistem meskipun terdapat berbagai perbedaan platform [11]. Layanan web adalah API yang menyediakan akses pengguna ke proses pencarian informasi. Melalui arsitektur *Representational State Transfer* (REST) yang berjalan di atas *Hypertext Transfer Protocol* (HTTP), itu termasuk file *Javascript Object Notation* (JSON) yang disajikan kepada pengguna yang mengakses API. [12]

## 2.7 Database

*Database* secara umum dapat diartikan sebagai lokasi penyimpanan data yang menggantikan file dokumen tradisional. Basis data didefinisikan sebagai kumpulan informasi yang saling terkait dan deskripsi informasi yang dirancang untuk memenuhi kebutuhan informasi suatu organisasi. [13]

*Database*, atau yang disebut basis data, adalah kumpulan informasi yang terhubung. Data biasanya terdapat dalam tabel-tabel yang dihubungkan bersama menggunakan *field*/kolom pada setiap tabel yang ada. [14]

## 2.8 PostgreSQL

PostgreSQL adalah *database* yang dikembangkan oleh Departemen Ilmu Komputer di University of California di Berkeley. Banyak platform mendukung PostgreSQL dan bebas royalti. PostgreSQL adalah *database relasional* yang mendukung perintah SQL dan PL/pgSQL (*Procedural Language*). Selain itu, PostgreSQL memiliki komunitas yang besar, dokumentasi yang lengkap, dan didukung oleh beberapa bahasa pemrograman seperti C++, Java, Perl, PHP, Python dan Tcl. [15]

PostgreSQL adalah DBMS *open source* yang mendukung berbagai bahasa SQL dan menawarkan beberapa fitur modern seperti: pertanyaan kompleks; kunci asing; pemicu; Perspektif; integritas transaksi; dan kontrol bersamaan *multiverse*. [16]

## 2.9 Golang

Bahasa pemrograman Go, juga dikenal sebagai Golang, adalah bahasa pemrograman sumber terbuka dengan gaya sintaksis yang mirip dengan bahasa C

dan C++ yang diterbitkan oleh Google. Didesain oleh tiga ilmuwan komputer Google, Robert Griesemer, Ken Thompson, dan Rob Pike, bahasa ini awalnya dimaksudkan untuk memperbaiki kebocoran memori yang telah lama membatasi C++. Resmi dirilis ke publik pada tahun 2009, Go menawarkan beberapa fitur hebat: sederhana, sehingga mudah dipelajari; secara sintaksis mudah dan manajemen memori yang baik; perakitan cepat; kebutuhan (struktural dan prosedural) yang mendukung konkurensi; itu menampilkan pengetikan statis, yang menjadikan Go bahasa pemrograman yang aman; gratis dan *open source* dengan dokumentasi lengkap; dan mendukung kompilasi silang (terjemahan lintas sumber daya). [17]

### **2.10 Visual Studio Code**

*Visual Studio Code (VS Code)* adalah editor teks yang ringan dan kuat dari Microsoft untuk sistem multi-pengguna. Editor teks ini mendukung *JavaScript*, *Typescript*, *Node.js*, dan bahasa pemrograman lainnya langsung melalui plugin yang dapat diinstal melalui pasar *Visual Studio Code*. Editor teks *VS Code* juga merupakan versi *open source*, di mana Anda dapat melihat kode sumber dan berkontribusi untuk pengembangannya. Kode sumber *VS Code* juga bisa dilihat di *link Github*. Hal ini juga yang membuat *VS Code* menjadi favorit di antara software developer karena software developer dapat berpartisipasi dalam proses pengembangan *VS Code* di masa mendatang. [18]

### **2.11 Postman**

*Postman* adalah aplikasi (dalam bentuk *add-on*) untuk browser *Chrome* yang bertindak sebagai klien REST untuk menguji REST API. *Postman* juga dapat diunduh sebagai aplikasi desktop. *Postman* adalah platform *GUI* yang kuat yang membuat pengembangan API lebih cepat dan lebih mudah, mulai dari membangun API hingga menguji, mendokumentasikan, dan berbagi. *Postman* direkomendasikan untuk sistem operasi *Mac*, *Windows*, atau *Linux*. [19]

### **2.12 Agile Software Development Methods**

*Agile methodology* adalah pendekatan dalam pengembangan perangkat lunak yang menekankan pada kolaborasi tim, adaptasi terhadap perubahan, dan pengiriman produk yang bernilai bagi pelanggan. *Agile methodology* membantu mengurangi kegagalan produk dan memungkinkan proyek untuk beradaptasi dengan perubahan yang terjadi. *Scrum*, di sisi lain, adalah kerangka kerja yang mengadopsi metodologi *agile*. *Scrum* membantu tim dalam menghasilkan produk yang sesuai dengan tujuan mereka melalui solusi adaptif dan kolaborasi tim. Terdapat beberapa alasan mengapa menggunakan *scrum* diantaranya, *Scrum* memungkinkan tim untuk beradaptasi dengan perubahan yang terjadi selama pengembangan produk. Dalam *Scrum*, perubahan kebutuhan atau prioritas dapat diakomodasi dengan mudah melalui *Sprint Planning* dan *Sprint Review*, *Scrum* mendorong kolaborasi yang intens antara anggota tim pengembangan. Tim bekerja bersama untuk mencapai tujuan yang ditetapkan dalam *Sprint* dan berkomunikasi secara teratur melalui *Daily Scrum*, dan *Scrum* menerapkan prinsip transparansi dan inspeksi yang memungkinkan tim untuk melihat kemajuan proyek secara jelas. Melalui *Sprint Review* dan *Sprint Retrospective*. [20]

### **2.13 Unified Modelling Language**

*UML* adalah salah satu *tool/model* untuk merancang pengembangan software yang berbasis *object-oriented*. *UML* sendiri juga memberikan standar penulisan sebuah sistem *blueprint*, yang meliputi konsep proses bisnis, penulisan kelas-kelas dalam bahasa program yang spesifik, skema database, dan komponen yang diperlukan dalam sistem software. [21]

*Unified Modeling Language (UML)* adalah alat untuk memvisualisasikan dan mendokumentasikan hasil analisis dan desain yang menyertakan sintaks untuk sistem pemodelan visual. Ini juga merupakan seperangkat konvensi pemodelan yang digunakan untuk mendefinisikan atau menggambarkan sistem perangkat lunak dalam kaitannya dengan objek.. [22]

*UML* memiliki fungsi untuk membantu pendeskripsian dan desain system perangkat lunak, khususnya system yang dibangun menggunakan pemrograman berorientasi objek. *UML* diciptakan dari penggabungan banyak bahasa pemodelan grafis berorientasi objek yang berkembang pesat pada akhir tahun 1980-an dan awal tahun 1990-an. Penggunaan *UML* tidak terbatas untuk metodologi tertentu, walaupun pada kenyataannya *UML* ini paling banyak digunakan pada metodologi berorientasi objek. Jenis-jenis Diagram *Unified Model Language (UML)* yaitu *use case diagram* berfungsi untuk mendeskripsikan sebuah interaksi antara satu atau lebih *actor* dengan system informasi yang akan dibangun, dan dapat menggambarkan fungsi apa saja yang ada pada sebuah sistem informasi, *class diagram* merupakan pembentukan utama dari system berorientasi objek yang mempresentasikan suatu *class* beserta dengan atribut dan operasinya. *activity diagram* menggambarkan *workflow* (aliran kerja) atau aktivitas dari sebuah system yang ada pada perangkat lunak. [23]

#### **2.14 Framework Fiber**

*Fiber* adalah kerangka web terinspirasi *Express* yang dibangun di atas *Fasthttp*, mesin HTTP tercepat untuk Go. Dirancang untuk memudahkan pengembangan cepat tanpa mempertimbangkan alokasi memori dan kinerja. [24]

Secara umum, "*Fiber*" adalah sebuah framework web yang digunakan untuk mengembangkan aplikasi berbasis web dengan menggunakan bahasa pemrograman *Go* (atau disebut juga *Golang*). *Fiber* didesain untuk menjadi ringan, cepat, dan efisien dalam menangani permintaan HTTP, serta memiliki banyak fitur yang memudahkan pengembangan aplikasi web, seperti routing yang cepat, dukungan untuk middleware, dan manajemen permintaan HTTP yang lebih baik. *Fiber* sangat cocok digunakan untuk mengembangkan aplikasi web yang membutuhkan kinerja tinggi dan waktu respon yang rendah.

#### **2.15 Google Cloud Platform**

*Google Cloud Platform* atau *GCP* adalah cloud publik layanan komputasi dari *Google LLC* yang terdiri dari berbagai layanan. Platform ini dari *Google*

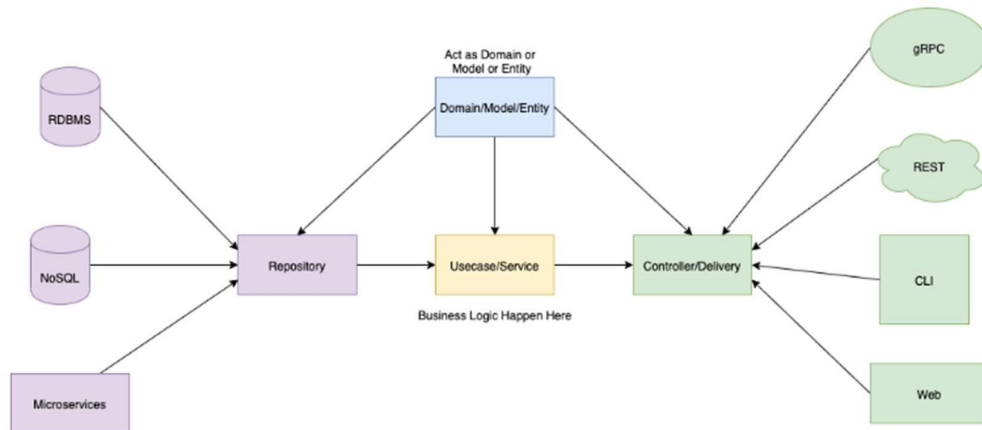
menyediakan berbagai layanan hosting mulai dari komputasi, penyimpanan, dan pengembangan aplikasi itu berjalan di perangkat keras *Google*. *Google Cloud Platform* dapat diakses melalui perangkat lunak pengembang, administrator cloud, dan TI lainnya profesional menggunakan internet publik atau melalui koneksi jaringan khusus. [25]

*Cloud* memungkinkan bisnis untuk mendapatkan sumber daya yang mereka butuhkan tanpa harus memiliki perangkat keras di tempat. Ini berarti bisnis dapat menyebarkan layanan di mana pun dan kapan pun mereka mau. *Cloud computing* dapat mengurangi biaya dengan memungkinkan perangkat keras dikonsolidasikan dan dikelola di lebih sedikit lokasi. Ini memberikan *skalabilitas* untuk mendukung beban kerja yang berfluktuasi, dan dapat memberdayakan kolaborasi di antara tim dan lokasi jarak jauh. [26]

### **2.16 Clean Architecture**

*Clean Architecture* adalah pendekatan dalam rekayasa perangkat lunak yang diusulkan oleh Robert C. Martin. Ini adalah pendekatan berbasis komponen dalam Rekayasa Perangkat Lunak (CBSE) yang membantu memisahkan aspek-aspek yang terkait dengan fungsionalitas yang bergantung pada platform dan yang tidak bergantung pada platform. Robert C. Martin mengatakan bahwa menggunakan arsitektur berlapis adalah titik awal yang baik, tetapi ketika perangkat lunak semakin besar dan kompleks, masalahnya adalah adanya tiga kontainer besar kode yang tidak memisahkan kode dengan benar. Selain itu, *Clean Architecture* adalah pendekatan yang berpusat pada domain, yang memungkinkan untuk menjelaskan semua elemen dari domain tersebut. Salah satu fitur penting dari *Clean Architecture* adalah kemampuannya untuk mengganti antarmuka pengguna (*UI*) dan sumber data tanpa masalah. Ini memungkinkan pengujian aturan bisnis tanpa bergantung pada *UI*, database, layanan, atau dependensi eksternal lainnya. Sebagai hasilnya, ini menghasilkan kode sumber dengan logika domain yang dikelilingi oleh komponen infrastruktur. [28]

Pada penelitian kali ini saya menggunakan *clean architecture* karena memudahkan pengembangan sistem Informasi untuk memisahkan domain atau *models response* dari komponen infrastruktur sehingga tidak bergantung satu sama lainnya, komponen infrastruktur dalam penelitian ini mengacu pada beberapa layer diantaranya *repository*, *usecase*, dan *handler*, berikut merupakan diagram dari struktur *clean architecture* yang saya gunakan



Gambar 2.1 Diagram Struktur *Clean Architecture*

### 2.17 Penelitian Terdahulu

Beberapa Penelitian sudah pernah membahas tentang Sistem Informasi *Point of Sales* (POS). Namun masing-masing penelitian memiliki beberapa kekurangan dan kelebihan. Penelitian-penelitian tersebut dijadikan referensi untuk mengembangkan situs web dan perbandingan untuk menghasilkan sistem web yang diharapkan

Penelitian dengan judul *Design of Web-Based Point of Sale Information System at Inti Peraga Mandiri*. Diambil dari *Proceeding International Applied Business and Engineering Conference*, diteliti oleh Faisal Amir, Riyanto, Fadhilah Oriyasmu, Fadli Fadilillah, dan Hafif Saputra pada tahun 2022 di Indonesia yang membahas tentang perancangan dan pengembangan sistem informasi *Point of Sale* (POS) berbasis web untuk Inti Peraga Mandiri, sebuah bisnis penjualan alat peraga sekolah. *Sistem ini* dirancang dengan tujuan utama untuk mengotomatisasi proses

pengumpulan data manual, mengurangi risiko kesalahan informasi, dan memberikan dukungan yang berarti bagi pemilik bisnis dalam mengelola operasi sehari-hari mereka. Sistem POS yang dirancang mencakup beragam fitur yang mendukung fungsionalitas yang diperlukan, seperti memberikan diskon pada transaksi, menyediakan ringkasan laporan secara tahunan, memberikan visualisasi data melalui fitur grafik, serta menyediakan tampilan halaman login, dashboard yang memberikan gambaran data penjualan, halaman input penjualan untuk memasukkan daftar pembelian produk, dan halaman laporan yang memungkinkan pengguna untuk menginputkan data penjualan berdasarkan periode waktu tertentu. [29]

Penelitian dengan judul *Point of Sales System in InHome Café Website Using Agile Methodology*. Diambil dari *Journal of Innovation and Community Engagement (ICE)*, diteliti oleh Irma Amelia Dewi, Yusup Miftahuddin, Muhammad Assidiq Fattah, Cikal Bingah Palenda, dan Syahrul Fathrurrahman Erawan pada tahun 2021 di Indonesia yang membahas pengembangan dan implementasi sistem *Point of Sales (POS)* untuk *InHome Café* menggunakan metodologi *Agile*, termasuk pengumpulan persyaratan, *product backlog*, *sprint backlog*, analisis dan desain sistem, jurnal tersebut juga menyajikan implementasi sukses dari sistem *POS*, yang menghasilkan peningkatan efisiensi dan akurasi dalam transaksi penjualan. Adapun fitur utama dalam penelitian jurnal di atas meliputi Sistem *Point of Sales (POS)* yang mampu mengambil data dari database, mengelola data, memproses transaksi penjualan, mencetak faktur, mengelola data pendapatan dari penjualan, dan melakukan proses pembelian serta penyimpanan data secara otomatis, Penerapan metodologi *Agile*, khususnya menggunakan framework *Scrum* dalam pengembangan sistem, adanya *workflow* sistem yang melibatkan pelanggan, kasir, pemilik, admin, dan superadmin. Pelanggan dapat melakukan pemesanan, kasir memproses pesanan, mencetak faktur, pemilik mengelola pesanan dan produk, admin mengelola laporan serta data master, dan superadmin memiliki akses penuh ke sistem, mengelola data pendapatan dari penjualan, dan melakukan proses pembelian serta penyimpanan data secara otomatis. [30]



Penelitian dengan judul *Point of Sales (POS) System Design Using Design Thinking Framework for Motorcycle Workshop*. Diambil dari *Journal of Information System and Informatics*, diteliti oleh Billy Macarius Sidhunata, Michael Kenang Gabbatha, Nicolas Arya Nanda Susilo, Pedro Manuel Lamberto Buu Sada, Biafra Daffa Farabi, Samuel Piolo, dan Yerik Afrianto Singgalen pada tahun 2023 di Indonesia yang membahas desain sistem manajemen modul *POS* yang dikembangkan untuk CV. Renaldi Motor Workshop. Penelitian ini bertujuan untuk meningkatkan efektivitas dan efisiensi perusahaan dengan mengoptimalkan teknologi digital dalam sistem pencatatan transaksi penjualan. Salah satu fitur utama yang dibahas dalam penelitian ini adalah antarmuka pengguna untuk halaman login dan halaman pengguna. Antarmuka ini dirancang untuk memenuhi kebutuhan pengguna dengan menampilkan data pesanan, mengambil pesanan, daftar master (kategori dan produk), laporan penjualan, dan sistem. Pengguna dapat menggunakan navigasi untuk mempercepat pencarian dan penyaringan data produk berdasarkan hari, bulan, dan tahun. Selain itu, pemilik bisnis dan super admin memiliki akses ke data produk dan riwayat transaksi untuk mengelola dan mengawasi kinerja karyawan. Fitur lain yang dibahas adalah halaman pesanan dan laporan penjualan. Halaman pesanan menampilkan informasi tentang jumlah pesanan dan status berdasarkan tanggal transaksi dan nomor faktur. Administrator dapat melihat, mengedit, dan menghapus data tersebut. Sedangkan laporan penjualan memudahkan pemilik bisnis dengan menampilkan informasi tentang tanggal transaksi, nomor faktur, total pesanan, dan pendapatan dalam rupiah. Keunggulan dari penelitian ini adalah penggunaan pendekatan *Design Thinking Framework* dalam merancang sistem *POS*. Pendekatan ini memastikan bahwa desain sistem memenuhi kebutuhan pengguna dan memberikan pengalaman pengguna yang baik. [31]

Penelitian dengan judul Rancang Bangun Sistem Informasi Software *Point of Sale (POS)* Dengan Metode *Waterfall* Berbasis Web. Diambil dari Jurnal Informatika Universitas Pamulang, diteliti oleh Putu Gede Surya Cipta Nugraha, Ni Wayan Wardani, dan I Wayan Sukarmayasa pada tahun 2021 di Indonesia yang membahas pengembangan sistem *point of sale (POS)* berbasis web menggunakan model *waterfall* Sistem *POS* ini dirancang untuk membantu pengusaha dalam mengelola

bisnis mereka dengan mengotomatisasi transaksi, manajemen inventaris, data pemasok, pembelian, manajemen pengguna, dan pembuatan laporan. Sistem ini dibangun menggunakan bahasa pemrograman PHP dan database MySQL. Fitur-fitur yang disajikan dalam sistem POS ini meliputi halaman *login* untuk pengguna, halaman penjualan untuk mencatat transaksi, halaman pemasok untuk mengelola data pemasok, halaman kategori untuk mengelompokkan produk, halaman produk untuk mengelola data produk, halaman pengguna untuk mengatur hak akses pengguna, halaman laporan untuk melihat laporan penjualan dan pembelian, serta halaman pembelian untuk mencatat pembelian produk. Selain itu, sistem ini juga memiliki desain arsitektur yang terstruktur, diagram *use case* yang jelas, dan model data fisik yang mencakup tabel-tabel untuk produk, pembelian, penjualan, kategori, pemasok, dan pengguna. [32]

Penelitian dengan judul Pengembangan Sistem Informasi *Point of Sales* (POS) Berbasis *Website* Untuk Manajemen *Home Industry* (Studi Kasus: Gelsey Real Surakarta). Diambil dari Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer, diteliti oleh Safria Isnibaiti, Issa Arwani, Widhy Hayuhardhika Nugraha Putra pada tahun 2022 di Indonesia yang membahas tentang pengembangan sistem informasi *Point of Sales* (POS) berbasis *website* untuk manajemen *Home Industry*. Tujuan dari penelitian ini adalah untuk mempermudah dalam merekap data penjualan secara rinci, jelas, dan efisien, serta terdapat beberapa fitur yang ada dalam penelitian pada jurnal ini antara lain, manajemen produk, manajemen pelanggan, manajemen penjualan, dan manajemen laporan. Penelitian pada jurnal ini dibuat menggunakan beberapa teknologi diantaranya framework laravel, bahasa pemrograman PHP, HTML, CSS dan juga JSON. [33]

Berdasarkan beberapa jurnal penelitian diatas hanya membuat sebuah fitur umum dalam sistem *point of sales* diantaranya fitur-fitur tersebut adalah pengelolaan data penjualan, laporan transaksi, autentikasi, pengelolaan produk, dan manajemen pengguna. Hal ini yang kemudian dikembangkan dalam penelitian ini yaitu menambahkan fitur-fitur pendukung diantaranya pembuatan melakukan transfer inventaris kepada outlet lainnya, melakukan pengembalian barang apabila tidak

sesuai atau rusak, pembuatan promosi per produk, melakukan *purchase order* kepada *supplier*, menambahkan *expired* pada masing-masing produk, serta dapat melihat stok barang yang terdapat digudang per outletnya, melakukan *grouping* produk, dan melakukan pencatatan pembukuan atau *settlement*.

### III. METODOLOGI PENELITIAN

#### 3.1 Waktu dan Tempat

Penelitian dilaksanakan mulai dari bulan Juli 2023 sampai dengan bulan Agustus 2023 yang bertempat di kantor PT Cakra Radha Mustika.

Tabel 3.1 Jadwal Penelitian

No	Aktivitas	Juli				Agustus			
		I	II	III	IV	I	II	III	IV
1	Product Backlog								
2	Sprint Planning Meeting								
3	Sprint Backlog								
4	Daily Standup								

#### 3.2 Alat dan Bahan Penelitian

##### 3.2.1 Alat Penelitian

Adapun alat yang digunakan pada penelitian ini adalah sebagai berikut:

Tabel 3.2 Alat Penelitian

No	Nama Alat	Spesifikasi	Deskripsi
1	Laptop	Prosesor AMD Ryzen 5 2500U, Memory 20GB, VGA AMD Radeon Vega 8, sistem operasi Windows	Perangkat keras untuk membangun aplikasi

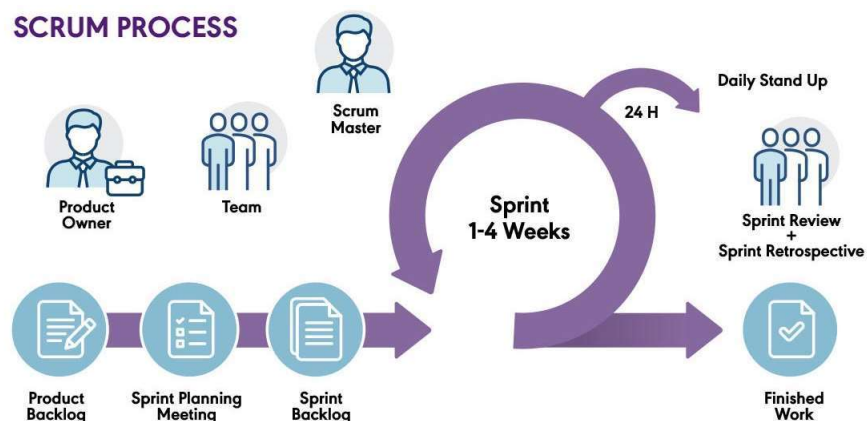
2	Visual Studio Code	Versi 1.81.1	Perangkat lunak untuk menuliskan baris kode aplikasi
3	Postman	Versi 10.17.5	Perangkat lunak yang digunakan untuk menguji <i>service</i> API
4	Golang	Versi 1.20.2	Bahasa pemrograman yang digunakan pada pembuatan <i>backend</i>
5	PostgreSQL	Versi 15	Sebuah sistem basis data yang menampung data aplikasi

### 3.2.2 Bahan Penelitian

Bahan penelitian yang digunakan dalam penelitian adalah sebuah *Product Requirement Document* (PRD) yang sudah disusun oleh tim *Product Management* dan sudah mencakup kebutuhan dari user.

### 3.3 Tahapan Penelitian

Rancang bangun *backend website point of sales* ini dilakukan dengan berbagai tahapan dengan tujuan untuk memudahkan penulisan penelitian. Penelitian dilakukan menggunakan metode *Agile Scrum*. Adapun tahapan yang dilakukan dalam penelitian dapat dilihat pada gambar berikut.



Gambar 3.1 Tahapan Penelitian

### 3.3.1 *Product Backlog*

Dalam metodologi *Agile Scrum*, tahap pertama ialah *Product Backlog*. Di sini, kebutuhan sistem muncul dari hasil identifikasi *user story*. Ini adalah daftar yang berisi semua pekerjaan yang dibutuhkan untuk mengembangkan produk, seperti fitur baru, perbaikan, dan tugas lainnya. Semua entri diurutkan berdasarkan nilai bisnis dan prioritasnya. *Product Backlog* berperan sebagai sumber utama kebutuhan dan tujuan bagi tim pengembangan.

#### 3.3.1.1 *User Stories*

Pada bagian ini, menggambarkan kebutuhan dari penggunaan yang dituliskan dalam naratif sederhana agar kebutuhan pengguna dapat dipahami dengan jelas. Berikut merupakan tabel *user stories* pengguna.

Tabel 3.3 *User Story*

No	Aktor	<i>User Story</i>
1	Cashier	Sebagai Cashier, saya harus bisa melakukan autentikasi login sebagai role cashier
		Sebagai Cashier, saya harus bisa membuat order
		Sebagai Cashier, saya harus bisa melihat transaction
		Sebagai Cashier, saya harus bisa membuat settlement
		Sebagai Cashier, saya harus bisa melihat promotion yang ada
2	Admin	Sebagai Admin, saya harus bisa melakukan autentikasi login sebagai role admin
		Sebagai Admin, saya harus bisa melihat transaction
		Sebagai Admin, saya harus bisa membuat settlement
		Sebagai Admin, saya harus dapat melihat stock di outlet saya

		Sebagai Admin, saya harus dapat membuat purchase order product
		Sebagai Admin, saya harus dapat membuat good receipt note untuk mengisi expired product
		Sebagai Admin, saya harus bisa melihat promotion yang ada
3	Superadmin	Sebagai Superadmin, saya harus bisa melakukan autentikasi login sebagai role superadmin
		Sebagai Superadmin, saya harus bisa melihat transaction
		Sebagai Superadmin, saya harus bisa membuat settlement
		Sebagai Superadmin, saya harus bisa melihat semua stock dari semua outlet
		Sebagai Superadmin, saya harus dapat membuat purchase order product
		Sebagai Admin, saya harus dapat membuat good receipt note untuk mengisi expired product
		Sebagai Superadmin, saya harus bisa mengatur promosi yang berlaku
		Sebagai Superadmin, saya harus bisa mengatur brand yang produknya dijual di Kalcare
		Sebagai Superadmin, saya harus bisa mengatur produk yang dijual di Kalcare
		Sebagai Superadmin, saya harus bisa mengatur other issue
		Sebagai Superadmin, saya harus bisa mengatur category channel
		Sebagai Superadmin, saya harus bisa mengatur channel untuk grouping product
		Sebagai Superadmin, saya harus bisa mengatur administration employee

### 3.3.1.2 Analisis Kebutuhan

Berdasarkan hasil identifikasi *user stories*, maka didapatkan kebutuhan dari sistem sebagai berikut

1. Autentikasi Pengguna

Sistem harus menyediakan fitur autentikasi yang memungkinkan pengguna untuk masuk ke sistem dengan tiga peran berbeda, yaitu cashier, Admin, dan Superadmin

2. Pembuatan Pesanan

Cashier harus memiliki akses dan kemampuan untuk membuat pesanan melalui sistem.

3. Melihat Transaksi

Semua role harus dapat melihat daftar transaksi yang sedang berlangsung maupun transaksi yang telah selesai di sistem

4. Pembuatan Settlement

Sistem harus memungkinkan pengguna untuk membuat settlement terkait transaksi.

5. Melihat dan Mengatur Promosi

Cashier dan Admin hanya boleh memiliki akses untuk melihat promosi yang sedang berlangsung dalam sistem, sedangkan Superadmin dapat mengatur promosi yang berlangsung maupun yang akan datang

6. Melihat Stok

Admin dan Superadmin harus dapat melihat stok produk di masing-masing outlet atau secara keseluruhan, sesuai dengan peran masing-masing.

7. Pembuatan Purchase Order

Admin dan Superadmin harus memiliki kemampuan untuk membuat purchase order untuk produk yang diperlukan.

8. Pembuatan Good Receipt Note

Admin dan Superadmin harus memiliki akses untuk melakukan Good Receipt Note, yang memungkinkan mereka untuk menambahkan tanggal kedaluwarsa pada produk yang telah dibeli melalui purchase order.



#### 9. Melihat dan Mengatur Other Issue

Admin hanya dapat melihat daftar other issue yang ada, sedangkan Superadmin memiliki hak akses untuk mengatur dan menyelesaikan other issue.

#### 10. Mengatur Master Data Brand dan Product

Superadmin harus memiliki kemampuan untuk mengatur brand dan product yang dijual dalam sistem.

#### 11. Mengatur Kategori Channel dan Channel Product

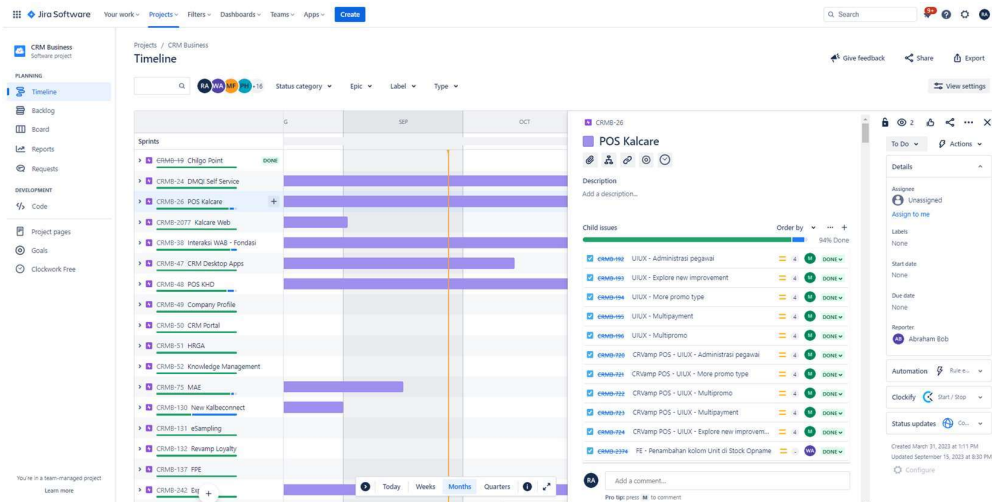
Superadmin harus dapat mengatur kategori channel yang digunakan dalam sistem, serta mengatur grouping produk pada kategori channel yang sudah dibuat

#### 12. Mengatur Administrasi Karyawan

Superadmin harus memiliki hak akses untuk mengatur informasi dan hak akses karyawan administrasi yang menggunakan sistem.

### **3.3.2 Sprint Planning Meeting**

Setelah *Product Backlog* terdefinisi, tim memasuki tahap *Sprint Planning Meeting* yang bertujuan menyusun rencana dan mengatur pekerjaan untuk periode *Sprint* tersebut. Item dari *Product Backlog* yang terpilih dipecah menjadi tugas-tugas terperinci, memberikan pemahaman mendalam kepada tim tentang pekerjaan yang harus dilakukan. Tim juga menggunakan diagram *use case* dan *activity* diagram serta *entity relationship diagram* untuk representasi visual yang jelas tentang pelaksanaan pekerjaan. Kombinasi perencanaan *Sprint* dan visualisasi diagram membantu tim dalam menjalankan pekerjaan dengan lebih terstruktur dan efisien.

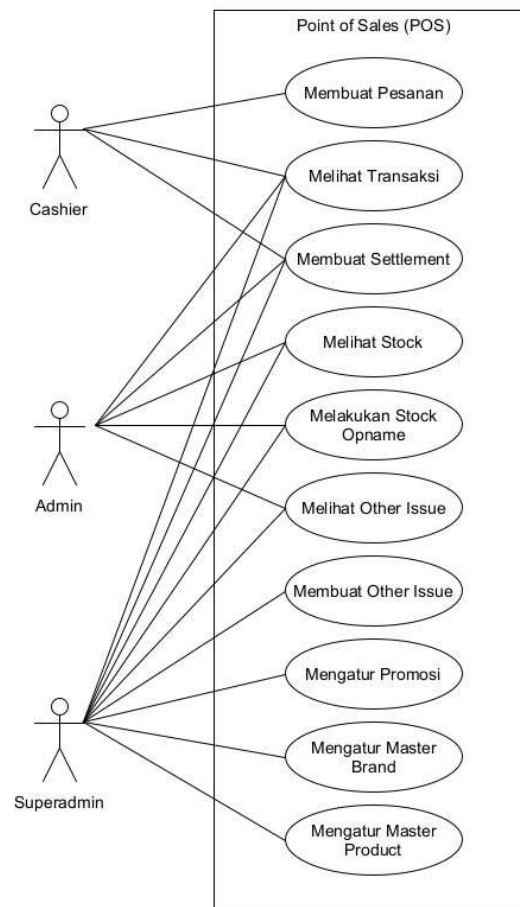


Gambar 3.2 *Product Backlog*

### 3.3.2.1 Use Case Diagram

*Use case diagram* digunakan untuk menggambarkan interaksi antara pengguna atau aktor yang terlibat dengan sistem aplikasi, serta menggambarkan berbagai fungsionalitas yang diperlukan oleh pengguna. *Use case diagram* dapat dilihat pada gambar berikut.

Pada gambar dibawah ini terdapat *usecase diagram* pengguna dengan 3 peran dalam aplikasi *point of sales*, diantaranya *cashier*, admin, dan superadmin. Pada peran *cashier* hanya dapat mengakses fitur membuat pesan, melihat transaksi dan membuat *settlement*. Pada peran admin memiliki tambahan akses yaitu melihat *stock*, melakukan *stock opname*, dan melihat *other issue*, lalu pada peran superadmin memiliki hak ases keseluruhan fitur kecuali pembuatan pesanan.



Gambar 3.3 Use Case Diagram Setelah Pengguna Login

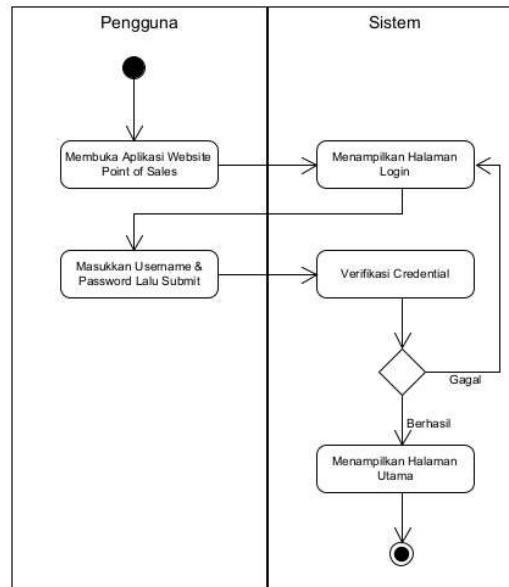
### 3.3.2.2 Activity Diagram

*Activity diagram* digunakan untuk menggambarkan aktivitas atau proses bisnis dari sebuah fitur didalam sistem. Berikut adalah *activity diagram* yang ada pada aplikasi *point of sales* (POS).

#### 1. Activity Diagram Autentikasi Pengguna

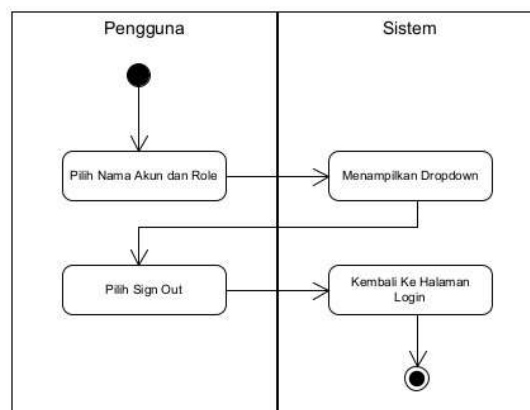
*Activity diagram* Autentikasi Pengguna dibagi menjadi dua, yaitu *login* dan *logout*. *Activity diagram login* menjelaskan alur proses pengguna untuk masuk ke dalam sistem. Pengguna memulai proses dengan masuk ke aplikasi, kemudian aplikasi akan menampilkan halaman *login*. Pengguna memasukkan *username* dan *password* lalu menekan tombol *submit*, setelah disubmit sistem akan memverifikasi data yang dikirim. Jika gagal maka, sistem akan tetap berada didalam halaman *login* dan

menampilkan pesan kesalahan. Jika berhasil, maka sistem akan masuk ke sistem dan menampilkan halaman utama aplikasi. *Activity diagram login* dapat dilihat pada gambar berikut.



Gambar 3.4 *Activity Diagram Login*

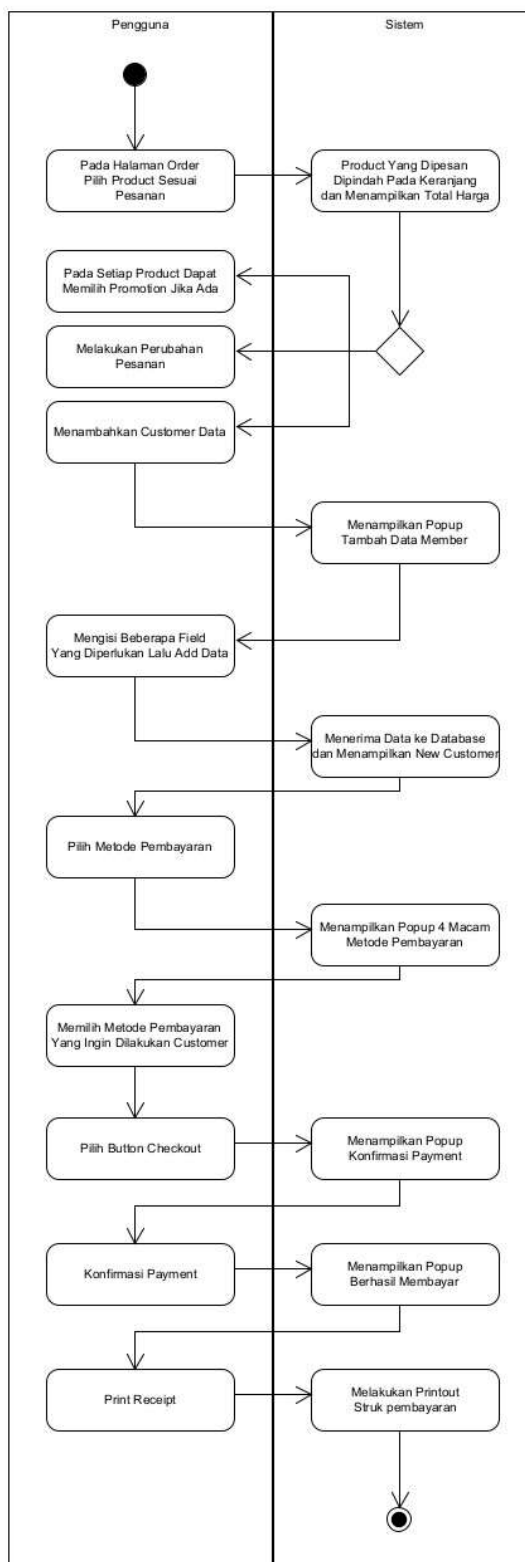
*Activity diagram logout* menjelaskan alur proses pengguna untuk keluar dari aplikasi. Pengguna memulai dengan memilih nama akun dan *role*, sistem akan menampilkan *dropdown* menu. Pengguna memilih tombol *sign out* lalu sistem akan menampilkan halaman *login*. *Activity diagram logout* dapat dilihat pada gambar berikut.



Gambar 3.5 *Activity Diagram Logout*

## 2. *Activity Diagram* Pembuatan Pesanan

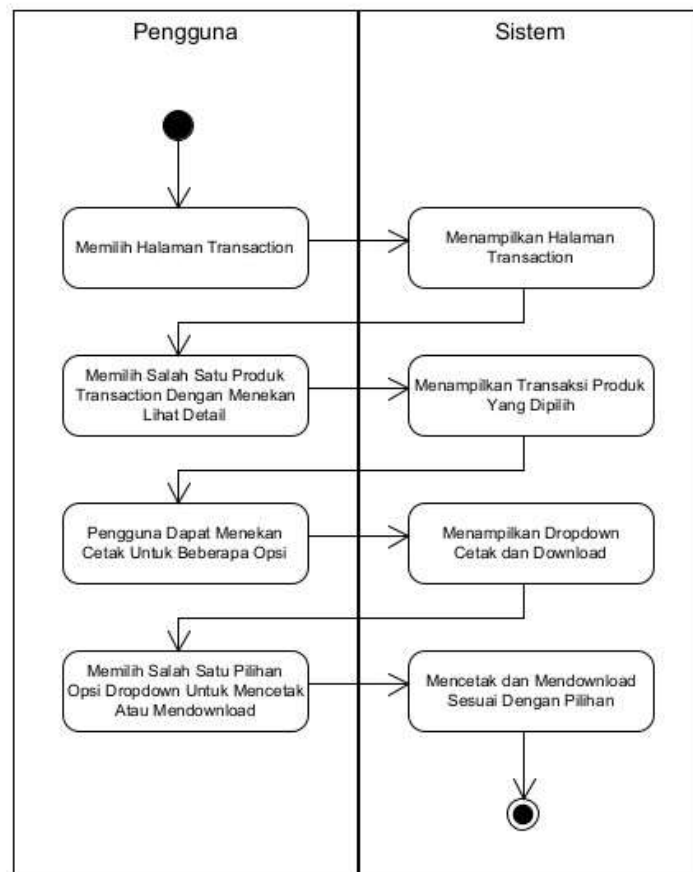
*Activity diagram* pembuatan pesanan ini dilakukan oleh peran kasir dengan alur proses pengguna sudah melakukan *login* terlebih dahulu, lalu pengguna masuk pada halaman *order*, pada halaman *order* kasir memilih produk yang sesuai dengan pesanan *customer*, kemudian sistem akan memindahkan barang tersebut kedalam keranjang dan menampilkan total harganya, lalu terdapat pilihan yaitu memilih memilih promo pada produk jika terdapat promo, melakukan perubahan data dan menambahkan *customer* data, pada *customer* data jika belum terdaftar sebagai *member* maka sistem akan menampilkan popup tambah data *member*, dan kasir mengisi beberapa *field* yang diperlukan untuk pendaftaran *member*, kemudian sistem akan menerima data tersebut dan menampilkan *new customer* pada pembayaran, lalu dilakukan pilih metode pembayaran, lalu sistem akan menampilkan popup yang berisikan 4 macam metode pembayaran, kasir memilih metode pembayaran yang ingin dilakukan oleh *customer* dan menekan tombol *checkout*, lalu sistem akan menampilkan popup konfirmasi *payment*, dan kasir mengkonfirmasi *payment*, sistem akan menampilkan popup berhasil melakukan pembayaran, dan kasir dapat melakukan *print struct*, maka sistem akan melakukan *printout* struk pembayaran. *Activity diagram* pembuatan pesanan dapat dilihat pada gambar berikut.



Gambar 3.6 Activity Diagram Pembuatan Pesanan

### 3. Activity Diagram Melihat Transaksi

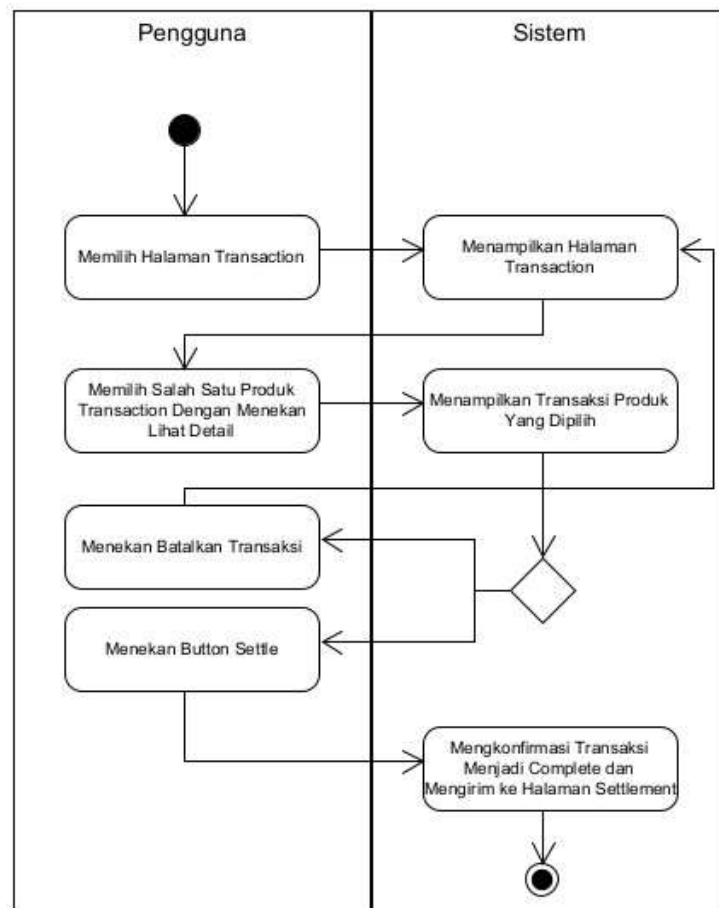
*Activity diagram* melihat transaksi ini dilakukan oleh peran kasir, admin dan superadmin dengan alur proses pengguna sudah melakukan *login* terlebih dahulu, pengguna memilih halaman transaksi, sistem akan menampilkan halaman transaksi sesuai dengan outlet nya, sedangkan jika peran superadmin dapat melihat semua transaksi dari seluruh outlet, lalu pengguna memilih salah satu produk transaksi dengan menekan lihat detail, sistem akan menampilkan transaksi detail dari transaksi yang dipilih, pengguna dapat menekan cetak dan akan diberikan beberapa pilihan, sistem akan menampilkan *dropdown* yang menampilkan beberapa pilihan untuk melakukan cetak *invoice*, cetak struk, ataupun *download invoice*, selanjutnya pengguna dapat memilih salah satu dari pilihan tersebut, maka sistem akan mencetak ataupun mendownload transaksi tersebut sesuai pilihan pengguna. *Activity diagram* melihat transaksi dapat dilihat pada gambar dibawah ini.



Gambar 3.7 Activity Diagram Melihat Transaksi

#### 4. Activity Diagram Pembuatan Settlement

Activity diagram pembuatan settlement ini dilakukan oleh peran kasir, admin dan superadmin dengan alur proses pengguna sudah melakukan *login* terlebih dahulu. Pengguna memilih halaman transaksi, sistem akan menampilkan halaman transaksi, pengguna dapat memilih salah satu produk yang ada pada halaman transaksi dengan menekan tombol lihat detail, sistem akan menampilkan halaman detail transaksi dari transaksi tersebut, pengguna dapat melakukan dua pilihan yaitu melakukan batal transaksi dan akan kembali kepada halaman transaksi atau menekan tombol *settle* untuk melakukan pembuatan *settlement*, sistem akan mengkonfirmasi transaksi menjadi selesai dan mengirim ke halaman settlement. Activity diagram pembuatan *settlement* dapat dilihat pada gambar dibawah ini.

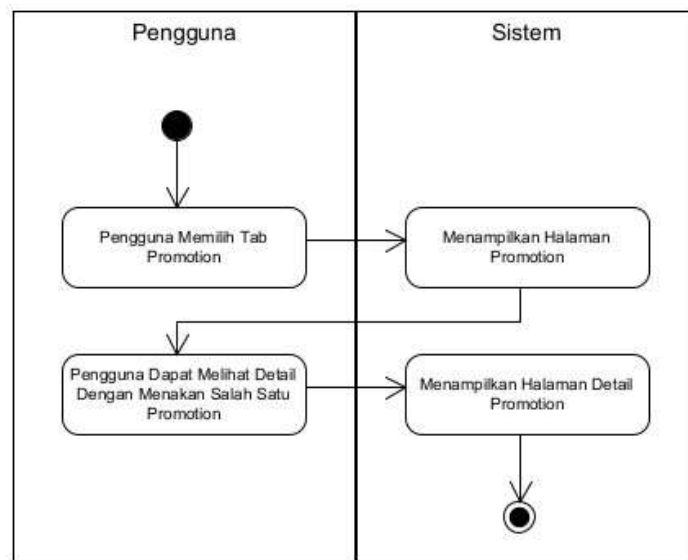


Gambar 3.8 Activity Diagram Pembuatan Settlement



### 5. Activity Diagram Melihat dan Mengatur Promosi

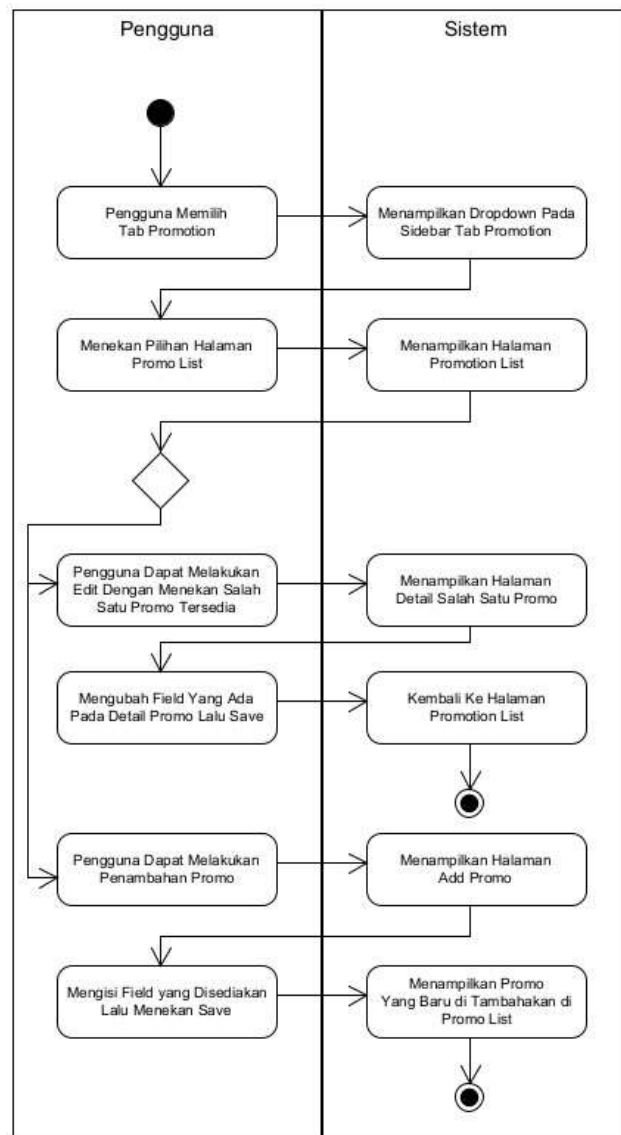
*Activity diagram* melihat dan mengatur promosi ini dilakukan oleh peran kasir, admin dan superadmin dengan terdapat pembatasan izin serta alur proses pengguna sudah melakukan *login* terlebih dahulu, Pada peran kasir dan admin hanya dapat melihat promosi yang berlangsung dengan alur pengguna memilih tab *promotion*, lalu sistem akan menampilkan halaman *promotion* yang berisikan promo yang sedang berlangsung dan yang sedang aktif maupun tidak aktif, pengguna dapat melihat detail promo dengan menekan salah satu promo tersebut, sistem akan menampilkan halaman detail promo tersebut. *Activity diagram* melihat promo dengan peran kasir dan admin dapat dilihat pada gambar dibawah ini.



Gambar 3.9 Activity Diagram Melihat Promo

Pada peran superadmin memiliki izin untuk mengatur promo yang sedang berlangsung maupun yang akan datang, dengan alur pengguna memilih tab *promotion*, sistem akan menampilkan *dropdown* pada *sidebar tab promotion*, pengguna dapat menekan pilihan halaman promo *list*, sistem akan menampilkan halaman promo *list*, dan diberikan beberapa pilihan untuk pengguna untuk melakukan pengubahan atau penambahan. Pada alur pengubahan, pengguna dapat menekan salah satu promo yang tersedia, sistem akan menampilkan halaman detail dari promo tersebut, pengguna dapat mengubah *field* yang ada sesuai yang

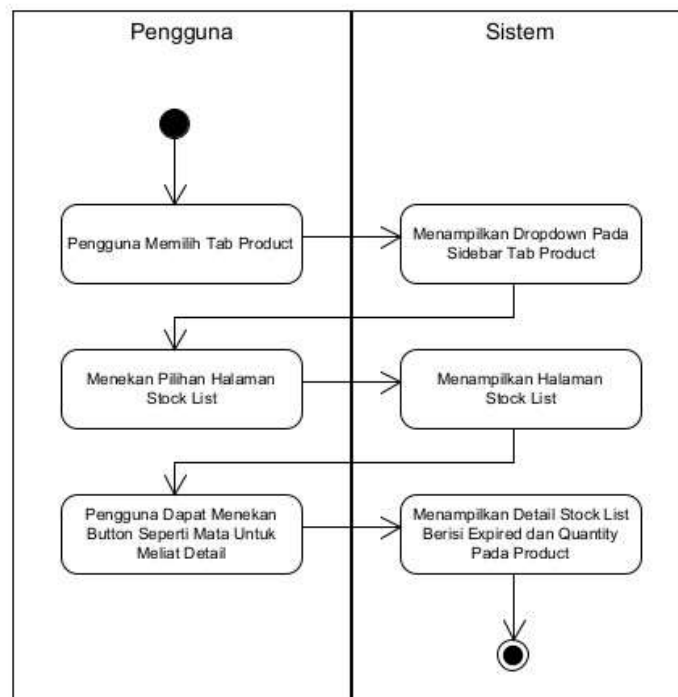
dibutuhkan lalu menekan tombol *save* jika sudah, sistem akan kembali ke halaman *promotion list* dengan pengubah yang sudah dilakukan. Pada alur penambahan, pengguna dapat menekan *add* promo, maka sistem akan menampilkan halaman *add* promo, pengguna diharuskan mengisi beberapa *field* yang diperlukan untuk membuat promo lalu menekan tombol *save* jika sudah, maka sistem akan menampilkan promo yang sudah dibuat pada halaman *promotion list*. *Activity Diagram* mengatur promo untuk superadmin dapat dilihat pada gambar dibawah ini.



Gambar 3.10 *Activity Diagram* Mengatur Promo

### 6. Activity Diagram Melihat Stok

*Activity diagram* melihat stok ini dilakukan oleh peran admin dan superadmin dengan alur proses pengguna sudah melakukan *login* terlebih dahulu, pengguna memilih halaman produk, sistem akan menampilkan *dropdown* pada *tab* produk, pengguna dapat menekan pilihan halaman stok, sistem akan menampilkan halaman stok yang berisikan stok produk sesuai dengan outlet nya, sedangkan jika peran superadmin dapat melihat semua stok produk dari seluruh outlet, pengguna dapat menekan *button* seperti gambar mata untuk melihat detail dari salah satu produk yang ada pada halaman stok, sistem akan menampilkan detail stok yang berisikan kadaluarsa produk, kuantitas dan unit dari produk stok tersebut. Activity Diagram melihat stok dapat dilihat pada gambar dibawah ini.

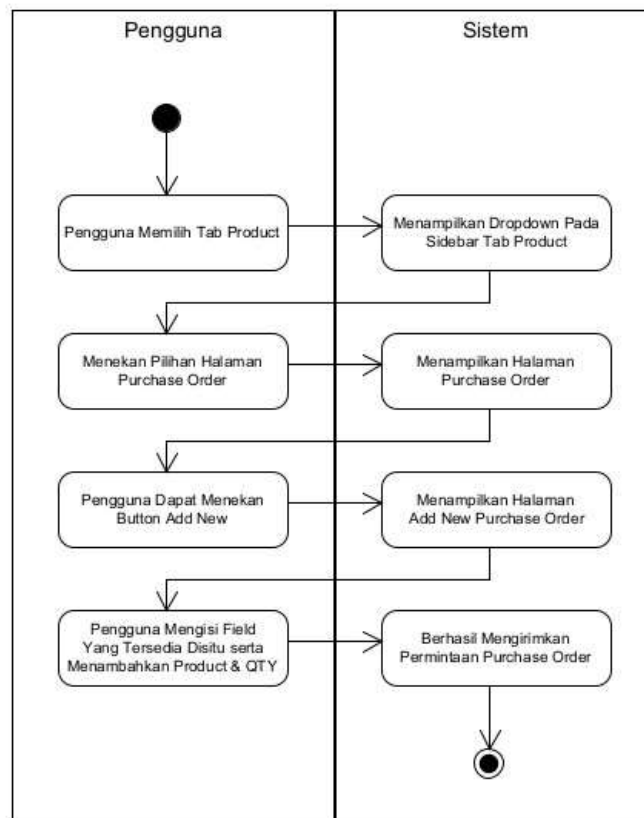


Gambar 3.11 Activity Diagram Melihat Stok

### 7. Activity Diagram Pembuatan Purchase Order

*Activity diagram* pembuatan Purchase Order ini dilakukan oleh admin dan superadmin dengan alur proses pengguna sudah melakukan *login* terlebih dahulu, pengguna memilih *tab* produk, sistem akan menampilkan *dropdown* pada *sidebar*

*tab* produk, pengguna dapat menekan pilihan halaman *purchase order*, sistem akan menampilkan halaman *purchase order*, pengguna dapat menekan tombol tambah baru, sistem akan menampilkan halaman untuk menambahkan *purchase order*, pengguna dapat mengisi beberapa *field* yang diperlukan dan menambahkan produk serta kuantitasnya yang dibutuhkan, sistem akan mengirimkan permintaan *purchase order*. *Activity diagram* pembuatan *purchase order* dapat dilihat pada gambar dibawah ini.

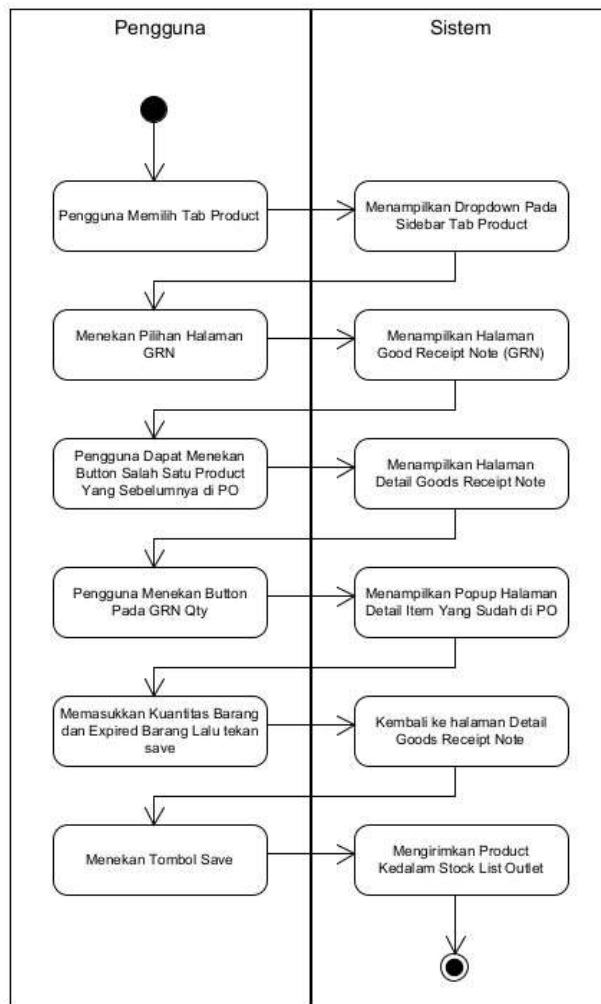


Gambar 3.12 *Activity Diagram* Pembuatan *Purchase Order*

#### 8. Pembuatan *Good Receipt Note*

*Activity diagram* pembuatan *Good Receipt Note* ini dilakukan oleh admin dan superadmin dengan alur proses pengguna sudah melakukan *login* terlebih dahulu dan juga sudah melakukan *purchase order*, selanjutnya pengguna memilih *tab* product, sistem akan menampilkan *dropdown* pada *sidebar tab* dari product, pengguna menekan pilihan halaman GRN, sistem akan menampilkan halaman

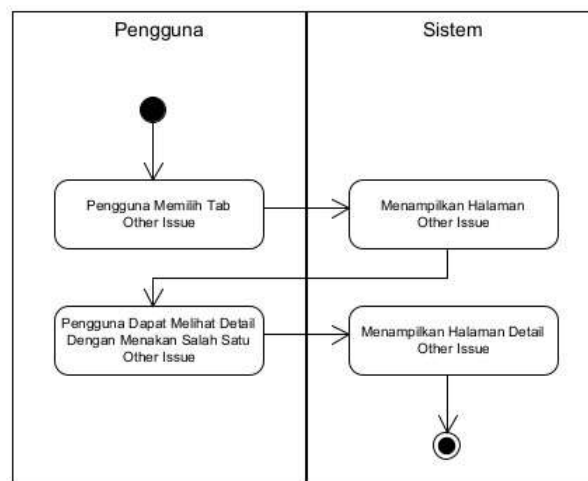
*Good Receipt Note (GRN)*, pengguna dapat menekan tombol salah satu product yang sebelumnya dilakukan *purchase order*, kemudian sistem akan menampilkan detail dari halaman *Goods Receipt Note*, pengguna dapat menekan tombol yang tersedia pada GRN Qty, maka sistem akan menampilkan sebuah popup detail item yang sudah dilakukan *purchase order*, pengguna dapat memasukkan kuantitas barang dengan *expired* barang yang sesuai dengan barang yang sudah dilakukan *purchase order* sebelumnya, lalu menekan tombol *save*, sistem akan kembali pada halaman detail GRN, pengguna dapat menekan tombol *save*, dan sistem akan mengirimkan *product* kedalam *stock list* dari outlet yang melakukan *purchase order* tersebut. *Activity diagram* pembuatan *good receipt note* ini dapat dilihat pada gambar dibawah ini.



Gambar 3.13 *Activity Diagram* Pembuatan *Good Receipt Note*

### 9. Activity Diagram Melihat dan Mengatur *Other Issue*

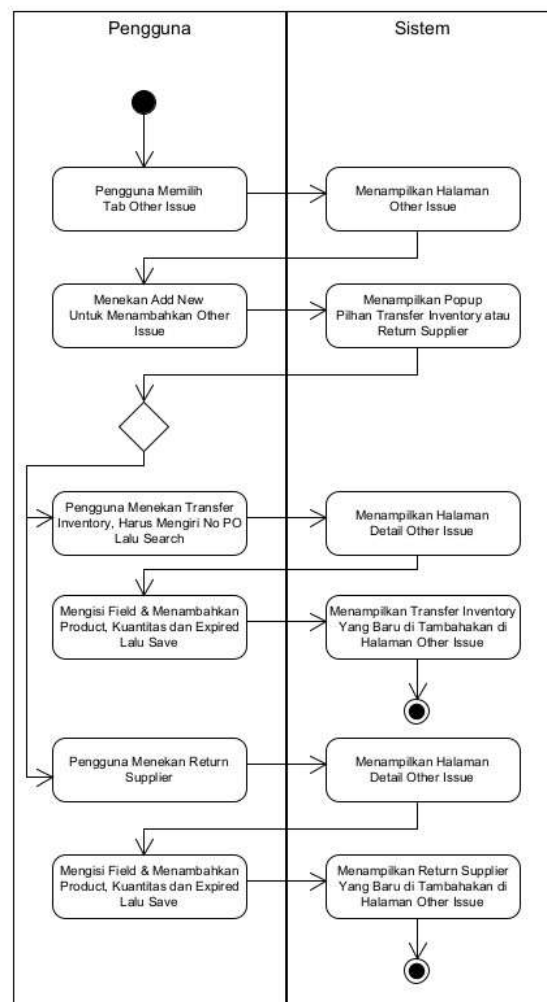
*Activity diagram* melihat dan mengatur *other issue* ini dilakukan oleh admin dan superadmin dengan terdapat pembatasan izin serta alur proses pengguna sudah melakukan *login* terlebih dahulu, peran admin hanya dapat melihat *other issue* yang ada, dengan alur proses pengguna memilih *tab other issue* pada *sidebar*, sistem akan menampilkan halaman *other issue* yang berisikan beberapa *issue* untuk melakukan *return supplier* ketika barang yang diterima terdapat kerusakan, ataupun *transfer inventory* yang dilakukan untuk mentransfer barang dari outlet ke outlet, lalu pengguna dapat melihat detail dengan menekan salah satu *other issue*, sistem akan menampilkan halaman detail dari *other issue* yang pengguna pilih. *Activity diagram* untuk melihat *other issue* yang dilakukan oleh admin dapat dilihat dibawah ini.



Gambar 3.14 *Activity Diagram* Melihat *Other Issue*

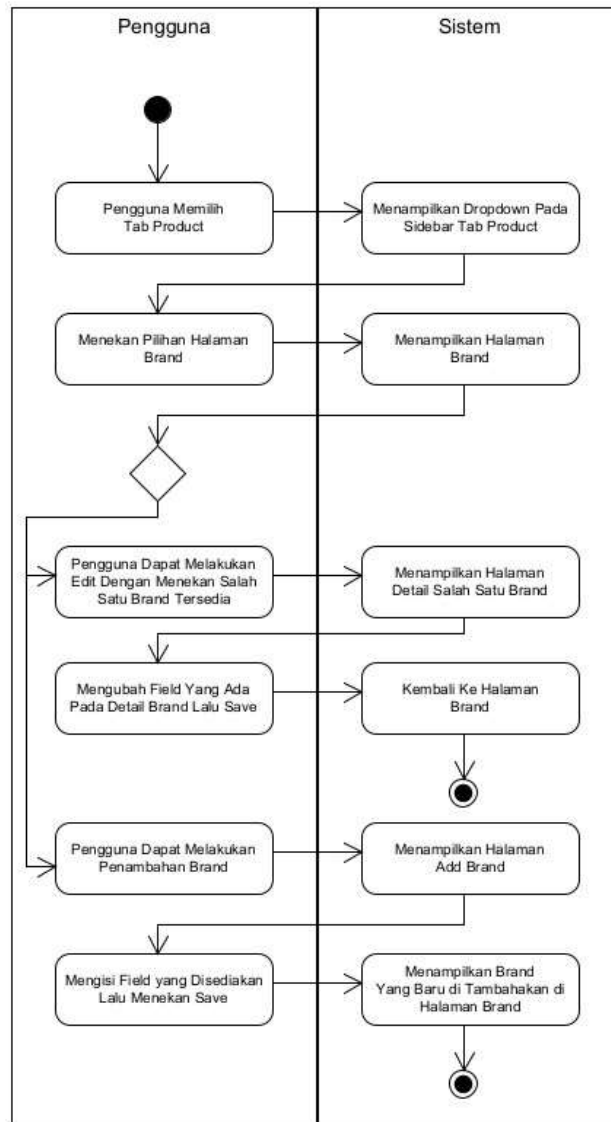
Pada peran superadmin memiliki izin untuk mengatur *other issue* yang akan dibuat, dengan alur pengguna memilih *tab other issue*, sistem akan menampilkan halaman *other issue*, pengguna dapat menekan *add new* untuk menambahkan *other issue*, sistem akan menampilkan popup untuk melakukan pilihan *transfer inventory* atau *return supplier*, pengguna diberikan beberapa pilihan untuk melakukan transfer atau *return*. Pada alur *transfer inventory*, pengguna dapat menekan *transfer inventory* dengan harus mengisi nomor *purchase order* yang dilakukan sebelumnya, sistem akan menampilkan halaman detail dari *other issue transfer inventory*,

pengguna dapat mengisi beberapa *field* dan menambahkan produk, kuantitas dan *expired* lalu menekan tombol *save*, sistem akan menampilkan *transfer inventory* yang baru ditambahkan pada halaman *other issue*. Pada alur *return supplier* pengguna dapat menekan tombol *return supplier* yang berguna untuk mengembalikan barang yang rusak ketika *purchase order* kepada *supplier*, kemudian sistem akan menampilkan halaman detail dari *other issue return supplier*, pengguna diharuskan mengisi beberapa *field* dan menambahkan produk, kuantitas serta tanggal kadaluarsa dari produk yang sudah di PO, lalu jika sudah pengguna menekan tombol *save*, maka sistem akan menampilkan *return supplier* yang baru ditambahkan pada halaman *other issue*. *Activity diagram* mengatur *other issue* yang dilakukan oleh superadmin dapat dilihat pada gambar dibawah ini.



Gambar 3.15 *Activity Diagram* Mengatur *Other Issue*

## 10. Activity Diagram Mengatur Brand



Gambar 3.16 Activity Diagram Mengatur Brand

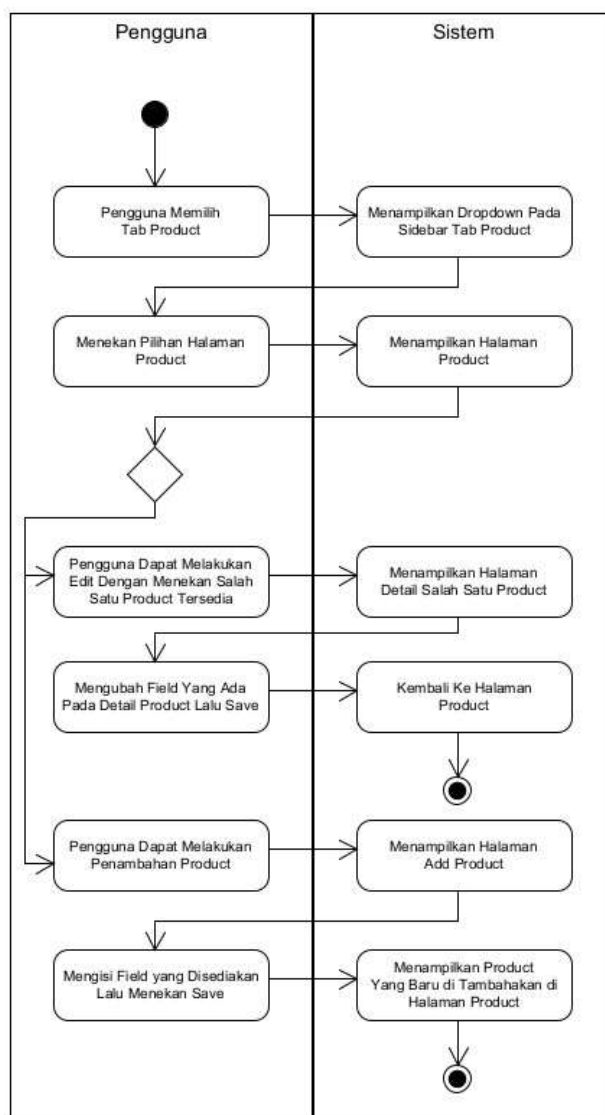
Pada *activity diagram* diatas mengatur *brand* hanya dapat dilakukan oleh superadmin dengan alur proses pengguna sudah melakukan *login* terlebih dahulu, Pengguna memilih *tab* produk, sistem akan menampilkan *dropdown* dari *sidebar* pada *tab* produk, pengguna memilih halaman *brand*, kemudian sistem akan menampilkan halaman *brand*, lalu terdapat 2 pilihan yang dapat dilakukan oleh pengguna yaitu melakukan pengubahan ataupun penambahan dari *brand*. Pada alur



pengubahan pengguna dapat menekan salah satu *brand* yang tersedia pada halaman, sistem akan menampilkan detail dari *brand* tersebut, lalu pengguna dapat melakukan perubahan dari *field* yang terdapat pada detail *brand* lalu menekan tombol *save*, sistem akan menyimpan perubahan dan kembali pada halaman *brand*. Pada alur penambahan *brand* pengguna dapat menekan tombol *add brand*, sistem akan menampilkan halaman *add brand*, lalu pengguna dapat mengisi *field* yang diperlukan untuk pembuatan *brand* lalu tekan tombol *save*, sistem akan menyimpan *brand* tersebut dan akan ditambahkan pada halaman *brand*.

#### 11. *Activity Diagram* Mengatur *Product*

*Activity diagram* mengatur produk hanya dapat dilakukan oleh superadmin dengan alur proses pengguna sudah melakukan *login* terlebih dahulu, Pengguna memilih *tab* produk, sistem akan menampilkan *dropdown* dari *sidebar* pada *tab* produk, pengguna memilih halaman produk, kemudian sistem akan menampilkan halaman produk, lalu terdapat 2 pilihan yang dapat dilakukan oleh pengguna yaitu melakukan pengubahan ataupun penambahan dari produk. Pada alur pengubahan pengguna dapat menekan salah satu produk yang tersedia pada halaman, sistem akan menampilkan detail dari produk tersebut, lalu pengguna dapat melakukan perubahan dari *field* yang terdapat pada detail produk lalu menekan tombol *save*, sistem akan menyimpan perubahan dan kembali pada halaman produk. Pada alur penambahan produk pengguna dapat menekan tombol *add* produk, sistem akan menampilkan halaman *add* produk, lalu pengguna dapat mengisi *field* yang diperlukan untuk pembuatan produk lalu tekan tombol *save*, sistem akan menyimpan produk tersebut dan akan ditambahkan pada halaman produk. *Activity diagram* mengatur produk dapat dilihat pada gambar dibawah ini.

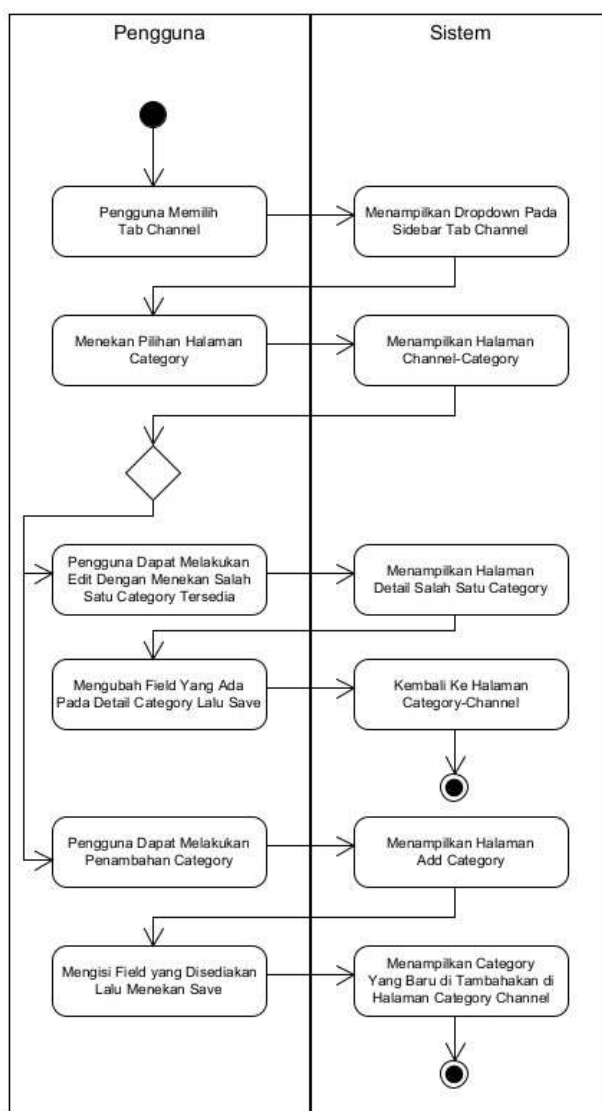


Gambar 3.17 Activity Diagram Mengatur Produk

## 12. Activity Diagram Mengatur Kategori Channel

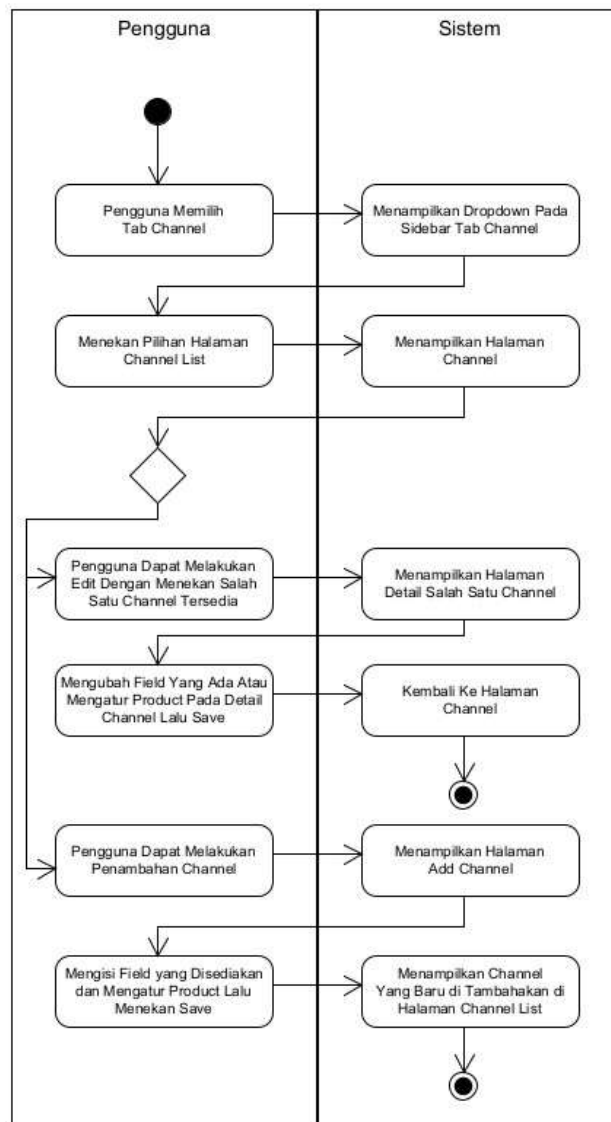
Activity diagram mengatur kategori channel hanya dapat dilakukan oleh superadmin dengan alur proses pengguna sudah melakukan login terlebih dahulu, Pengguna memilih tab channel, sistem akan menampilkan dropdown dari sidebar pada tab channel, pengguna memilih halaman category, kemudian sistem akan menampilkan halaman channel-category, lalu terdapat 2 pilihan yang dapat dilakukan oleh pengguna yaitu melakukan perubahan ataupun penambahan dari kategori. Pada alur perubahan pengguna dapat menekan salah satu kategori yang

tersedia pada halaman, sistem akan menampilkan detail dari kategori tersebut, lalu pengguna dapat melakukan perubahan dari *field* yang terdapat pada detail kategori lalu menekan tombol *save*, sistem akan menyimpan perubahan dan kembali pada halaman kategori *channel*. Pada alur penambahan kategori pengguna dapat menekan tombol *add* kategori, sistem akan menampilkan halaman *add* kategori, lalu pengguna dapat mengisi *field* yang diperlukan untuk pembuatan kategori lalu tekan tombol *save*, sistem akan menyimpan kategori tersebut dan akan ditambahkan pada halaman kategori. *Activity diagram* mengatur kategori *channel* dapat dilihat pada gambar dibawah ini.



Gambar 3.18 *Activity Diagram* Mengatur Kategori *Channel*

## 13. Activity Diagram Mengatur Channel



Gambar 3.19 Activity Diagram Mengatur Channel

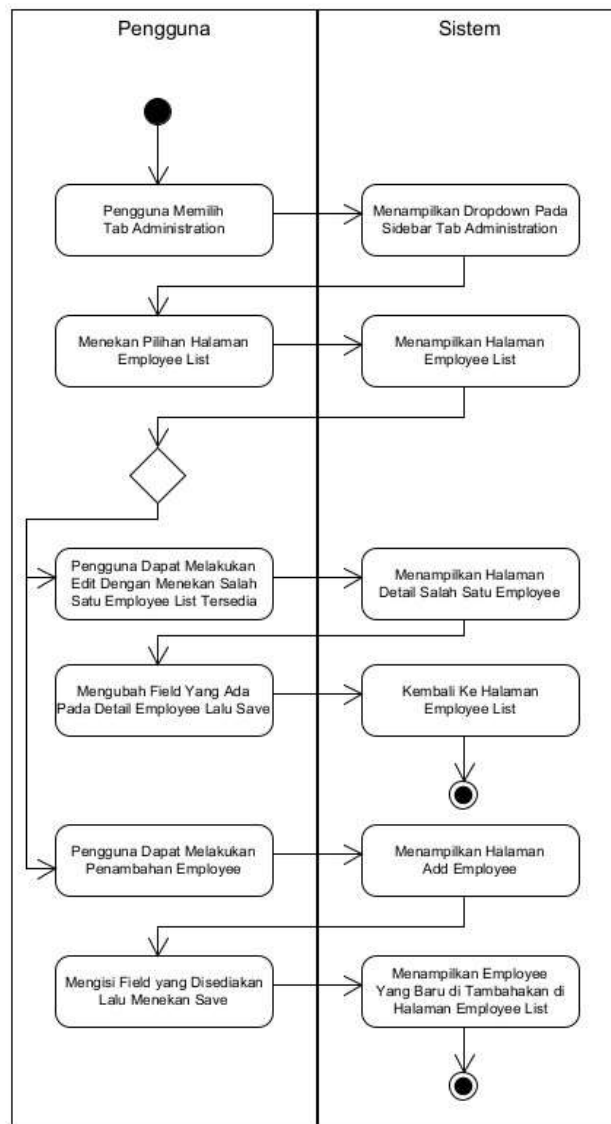
Pada *activity diagram* mengatur channel hanya dapat dilakukan oleh superadmin dengan alur proses pengguna sudah melakukan *login* terlebih dahulu, Pengguna memilih *tab channel*, sistem akan menampilkan *dropdown* dari *sidebar* pada *tab channel*, pengguna memilih halaman *channel list*, kemudian sistem akan menampilkan halaman *channel*, lalu terdapat 2 pilihan yang dapat dilakukan oleh pengguna yaitu melakukan pengubahan ataupun penambahan dari *channel*. Pada

alur perubahan pengguna dapat menekan salah satu *channel* yang tersedia pada halaman, sistem akan menampilkan detail dari *channel* tersebut, lalu pengguna dapat melakukan perubahan dari *field* yang terdapat pada detail *channel* dan menambahkan atau mengurangi beberapa produk sesuai dengan kebutuhan dari *channel* lalu menekan tombol *save*, sistem akan menyimpan perubahan dan kembali pada halaman *channel list*. Pada alur penambahan *channel* pengguna dapat menekan tombol *add channel*, sistem akan menampilkan halaman *add channel*, lalu pengguna dapat mengisi *field* yang diperlukan untuk pembuatan *channel* dan menambahkan beberapa produk yang diperlukan dalam proses tersebut lalu tekan tombol *save*, sistem akan menyimpan *channel* tersebut dan akan ditambahkan pada halaman *channel list*, *channel* bertujuan untuk melakukan *grouping* dari produk yang sudah ada dengan memanfaatkan kategori *channel* dan *master* data dari produk maupun *brand*.

#### 14. *Activity Diagram* Mengatur Administrasi Karyawan

*Activity diagram* mengatur administrasi karyawan hanya dapat dilakukan oleh superadmin dengan alur proses pengguna sudah melakukan *login* terlebih dahulu, Pengguna memilih *tab administration*, sistem akan menampilkan *dropdown* dari *sidebar* pada *tab administration*, pengguna memilih halaman *employee list*, kemudian sistem akan menampilkan halaman *employee list*, lalu terdapat 2 pilihan yang dapat dilakukan oleh pengguna yaitu melakukan perubahan ataupun penambahan dari *employee*. Pada alur perubahan pengguna dapat menekan salah satu *employee* yang tersedia pada halaman, sistem akan menampilkan detail dari *employee* tersebut, lalu pengguna dapat melakukan perubahan dari *field* yang terdapat pada detail *employee* sesuai dengan kebutuhan lalu menekan tombol *save*, sistem akan menyimpan perubahan dan kembali pada halaman *employee list*. Pada alur penambahan *employee* pengguna dapat menekan tombol *add employee*, sistem akan menampilkan halaman *add employee*, lalu pengguna dapat mengisi *field* yang diperlukan dan disesuaikan dengan kebutuhan lalu tekan tombol *save*, sistem akan menyimpan *employee* tersebut dan akan ditambahkan pada halaman *employee list*, administrasi karyawan bertujuan untuk mengatur informasi dan hak akses karyawan

administrasi yang menggunakan sistem. *Activity diagram* mengatur administrasi karyawan dapat dilihat pada gambar dibawah ini.

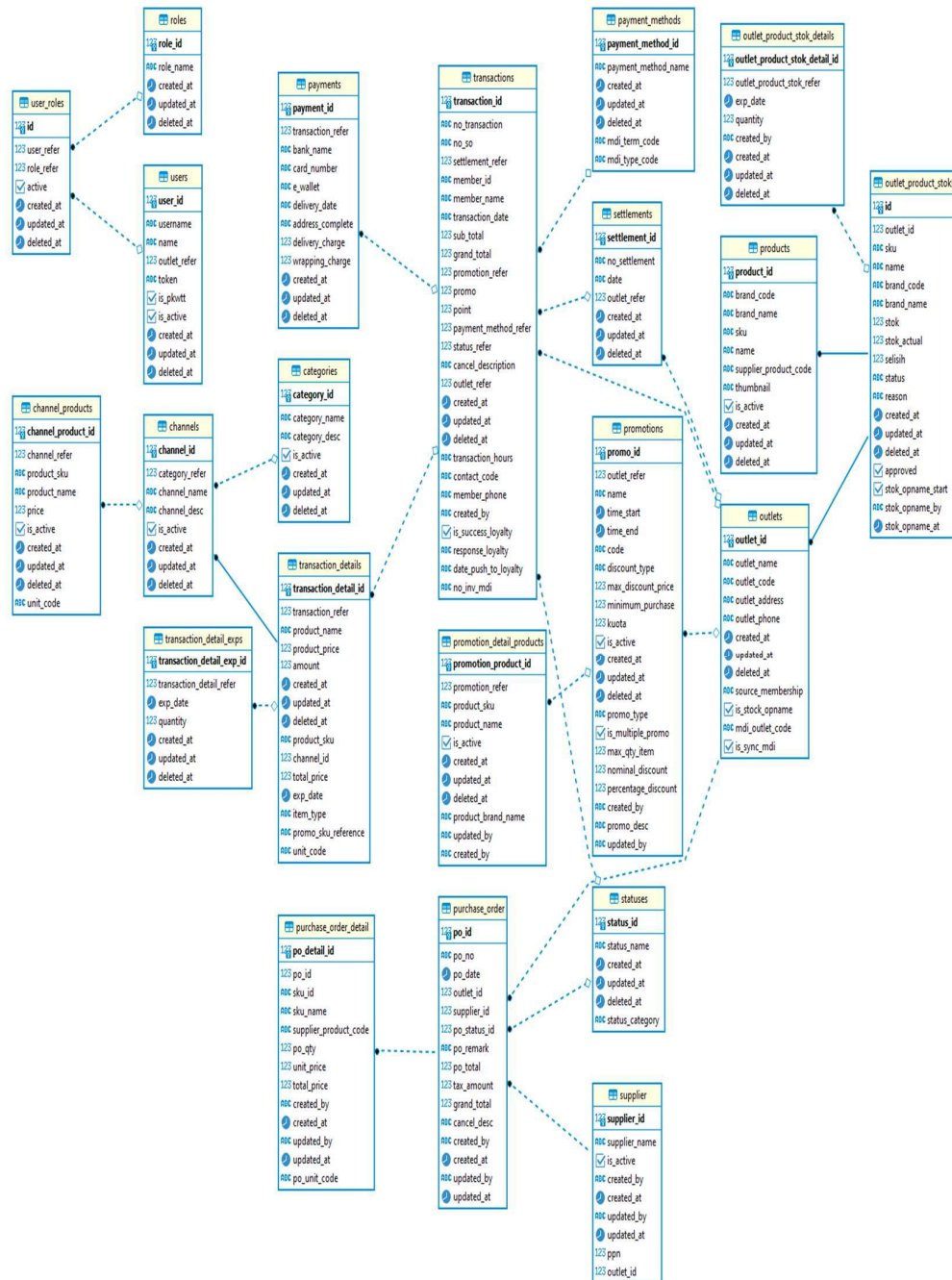


Gambar 3.20 *Activity Diagram* Mengatur Administrasi Karyawan

### 3.3.2.3 Perancangan *Database*

Perancangan *database* adalah proses merencanakan dan mengorganisir struktur data, tabel, relasi, serta atribut-atributnya dalam sebuah sistem informasi, yang bertujuan untuk mengoptimalkan penyimpanan, pengambilan, dan manipulasi data.

Perancangan database pada aplikasi *point of sales* (POS) ini dapat dilihat pada gambar dibawah ini.

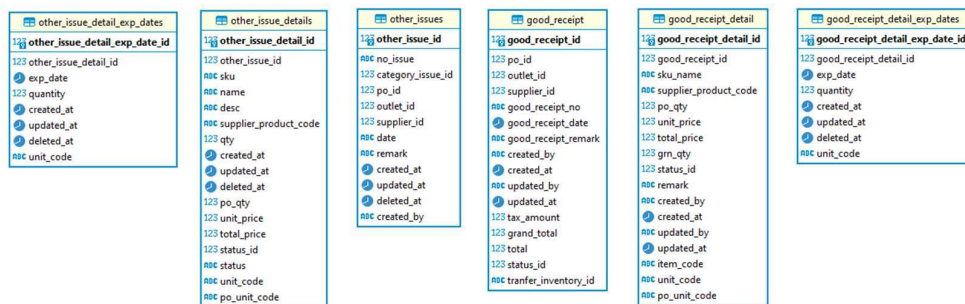


Gambar 3.21 Entity Relationship Diagram Dengan Foreign Key

Pada gambar *entity relationship diagram* diatas, hasil desain *database* yang telah disusun untuk mendukung pengembangan *backend* aplikasi *Point of Sales* (POS).

Desain *database* ini dibuat dengan menggunakan sistem manajemen *database* PostgreSQL. ERD ini mencakup beberapa tabel utama yang saling terhubung melalui relasi *foreign key*. Setiap tabel dalam desain ini merepresentasikan entitas atau objek yang berperan penting dalam aplikasi POS. Entitas-entitas ini mencakup informasi-informasi yang diperlukan untuk menjalankan berbagai fungsi dalam aplikasi, seperti menyimpan data produk, data pelanggan, transaksi penjualan, stok barang, dan sebagainya.

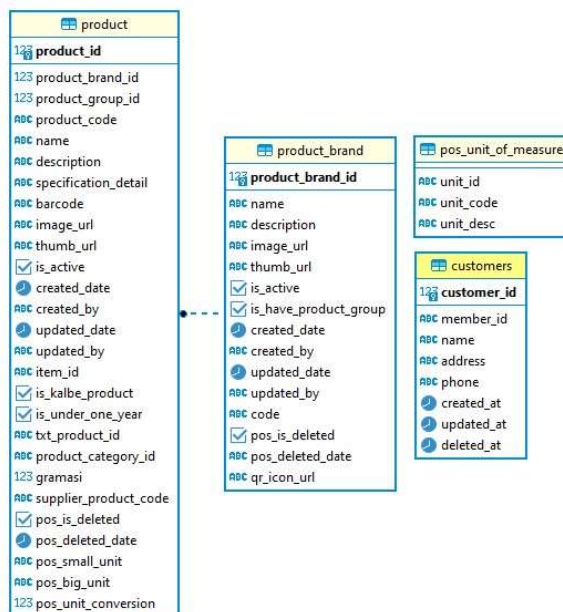
Desain *database* ini memastikan bahwa hubungan antara entitas-entitas tersebut dijaga dengan baik melalui penggunaan kunci asing (*foreign key*), sehingga memungkinkan integritas data dan konsistensi dalam aplikasi.



Gambar 3.22 Entity Relationship Diagram Tanpa Foreign Key

Pada gambar diatas merupakan gambar dari ERD yang dibangun tanpa adanya penggunaan kunci asing (*foreign key*) secara langsung. Namun, hubungan antara tabel-tabel tersebut tetap terjalin melalui kueri SQL menggunakan operasi *JOIN*. Terdapat beberapa tabel, seperti yang terkait dengan fitur "Other Issue" dan "Good Receipt Note" yang memerlukan penggunaan *po\_id* sebagai kunci referensi dalam proses penghubungannya melalui kueri SQL.





Gambar 3.23 Entity Relationship Diagram Master Data Global

Pada gambar diatas merupakan *entity relationship diagram* yang telah diimplementasikan dalam tiga database terpisah: tabel *product*, *product\_brand*, dan *pos\_unit\_of\_measures*. Pendekatan ini bertujuan untuk mengisolasi data referensi dari data transaksional utama, sehingga memudahkan manajemen data yang lebih efisien dan aman. Data yang ditempatkan secara terpisah ini termasuk data yang jarang berubah dan dapat digunakan oleh berbagai bagian dari aplikasi. Penggunaan database master data yang terpisah memungkinkan penekanan pada pemeliharaan, pencadangan (*backup*), dan pemulihan data referensi tanpa terganggu oleh operasi harian dalam *database* transaksional. Selain itu, data referensi yang memiliki frekuensi perubahan yang rendah dapat disimpan dengan lebih terorganisir dan terjaga keamanannya. Hal ini menghasilkan manfaat tambahan, seperti pengelolaan data yang lebih efisien, dan memungkinkan akses yang aman dan terkendali ke data referensi oleh berbagai bagian dari aplikasi atau sistem.

### 3.3.3 Sprint Backlog

*Sprint backlog* merupakan salah satu elemen penting dalam kerang kerja pengembangan perangkat lunak *scrum* yang berisikan daftar tugas atau pekerjaan yang harus diselesaikan oleh tim pengembangan selama satu sprint tertentu. Sprint

*backlog* dibuat oleh *Product Owner* dan tim pengembangan bekerja sama dalam mengidentifikasi dan memprioritaskan item-item pekerjaan yang akan dimasukkan ke dalam *Sprint backlog*. *Sprint backlog* membantu tim pengembangan fokus pada tujuan *Sprint* dan menghindari gangguan selama periode tersebut. Ini juga memungkinkan tim untuk membuat rencana yang lebih rinci tentang bagaimana mereka akan mencapai tujuan mereka dalam *Sprint* tersebut. Pada pengkodean pada tahapan *sprint backlog* ini menerapkan konsep *clean architecture* dalam pengkodean tugas-tugas tersebut.

### **3.3.4 Daily Standup**

*Daily Scrum* adalah pertemuan harian yang singkat, biasanya berlangsung 15 menit atau kurang. Ini diadakan setiap hari kerja selama *Sprint* dengan tujuan meningkatkan komunikasi, koordinasi, dan kolaborasi dalam tim pengembangan serta membantu tim untuk tetap terfokus pada pencapaian tujuan *Sprint* dan mengidentifikasi hambatan apa pun yang mungkin menghambat kemajuan pengembangan *daily standup* dihadiri oleh Semua anggota tim pengembangan dan *scrum master* yang terlibat dalam *Sprint*. Berikut merupakan kegiatan *daily standup* yang dihadiri setiap harinya dapat dilihat dibawah ini.

Pada dasarnya, *Daily Stand-Up* adalah pertemuan harian singkat yang merupakan komponen utama dalam siklus *Sprint* di dalam kerangka kerja Scrum. Namun, ada dua elemen penting yang menjadi bagian dari siklus *Sprint* yang lebih besar, yaitu *Sprint Review* dan *Sprint Retrospective*.

#### **3.3.4.1 Sprint Review**

*Sprint review* merupakan pertemuan yang diadakan di akhir setiap *sprint* untuk meninjau dan mengevaluasi hasil kerja yang telah diselesaikan oleh tim pengembangan selama *sprint* tersebut, ini membantu memastikan bahwa tujuan *Sprint* telah tercapai dan memberikan kesempatan untuk memperbarui *Product Backlog* berdasarkan umpan balik dari pemangku kepentingan. *Sprint review* diadakan sekali pada akhir setiap *sprint* dan dihadiri oleh seluruh tim pengembangan dan *scrum master*.

#### **3.3.4.1 *Sprint Retrospective***

*Sprint Retrospective* adalah pertemuan yang diadakan setelah *Sprint Review*. Tujuannya adalah untuk memeriksa dan mengevaluasi proses pengembangan, praktik kerja, dan perubahan yang dapat dilakukan untuk meningkatkan produktivitas dan efektivitas tim dalam *Sprint* berikutnya. *Sprint Retrospective* diadakan sekali pada akhir setiap *Sprint*, biasanya setelah *Sprint Review*. *Sprint Retrospective* melibatkan seluruh tim pengembangan dan dapat dihadiri oleh *Scrum Master*.

#### **3.3.5 *Finished Work***

*Finished Work* merupakan tahap akhir dari sebuah *sprint* di mana produk yang sudah dapat digunakan diserahkan kepada pengguna melalui *product manager*. Produk ini harus berfungsi dengan baik, stabil, dan memenuhi kebutuhan yang telah ditetapkan sebelumnya.

## V. SIMPULAN DAN SARAN

### 5.1 Simpulan

Berdasarkan hasil dari penelitian yang telah disampaikan, didapatkan kesimpulan sebagai berikut:

1. Penelitian yang dilakukan berhasil membangun *backend* sistem informasi *Point of Sales* (POS) menggunakan bahasa pemrograman golang dengan Framework *Fiber* serta dilakukan pengujian dengan melakukan pengiriman endpoint pada aplikasi postman dan mendapatkan *response* yang sesuai dengan parameter yang dikirimkan
2. Sistem Informasi *Point of Sales* (POS) berhasil menyediakan 12 fitur dan digunakan pada kasir Kalcare.
3. Pengujian difokuskan secara lebih mendalam pada proses pemesanan untuk memastikan respons yang sesuai dengan parameter yang dikirim, serta pada pengurangan stok di outlet dengan penyesuaian jumlah yang dikeluarkan.
4. Sistem *Point of Sales* (POS) ini telah aktif digunakan oleh pengguna untuk menjalankan transaksi penjualan barang di masing-masing outlet Kalcare

### 5.2 Saran

Adapun saran untuk pengembangan selanjutnya adalah sebagai berikut:

1. Melakukan pengembangan aplikasi lebih lanjut menggunakan *microservices* dengan teknologi *gRPC* dan *GraphQL* untuk mendapatkan balasan data yang lebih ringan dengan konsep *transcoding binary* dibandingkan *RESTful API* dengan format *JSON*.
2. Melakukan pengembangan pada proses pembayaran dengan memanfaatkan *payment gateway*.

## DAFTAR PUSTAKA

- [1] J. Sugihartono, K. I. Satoto, and E. D. Widiyanto, “Pembuatan Aplikasi Point of Sale Toko Cabang Perusahaan Torani Menggunakan Framework CodeIgniter,” *J. Teknol. dan Sist. Komput.*, vol. 3, no. 4, p. 445, 2015, doi: 10.14710/jtsiskom.3.4.2015.445-455.
- [2] B. Loardy, B. Bunawan, and P. Hartono, “Aplikasi Point Of Sales Yang Terhubung Dengan Electronic Data Capture,” *no. Lc*, no. Lc, pp. 3–6, 2010.
- [3] PT Cakra Radha Mustika, “Company Profile,” 2022. <https://cakraradhamustika.com/aboutus> (accessed Sep. 06, 2023).
- [4] A. Elisabet Yunaeti and I. Rita, *Pengantar Sistem Informasi*. Yogyakarta: Andi Offset, 2017.
- [5] E. Sutanta, *Basis Data Dalam Tinjauan Konseptual*. Yogyakarta: Andi Offset, 2011.
- [6] A. Susanto, “Rancang Bangun Sistem Informasi Penjualan Aksesoris Hanphone Berbasis Web Pada Dazzle Cellular Semarang,” *J. Din. Inform.*, vol. 5, no. 1, pp. 1–6, 2013.
- [7] R. Abdulloh, *Web Programming is Easy*. Jakarta: PT Elex Media Komputindo, 2015.
- [8] Y. Susilowati, *Modul E-Commerce - Teaching Factory For Students*. Mutiara Publisher, 2019.
- [9] R. Pangestika and R. T. Dirgahayu, “Pengembangan Back-end Sistem Informasi Pendataan Sekolah Desa Komunitas Pendar Foundation,” *Automata*, vol. 1, no. 2, pp. 184–189, 2020.
- [10] A. Mubariz, D. Nur, E. Tungadi, and M. N. Y. Utomo, “Perancangan Back-End Server Menggunakan Arsitektur Rest dan Platform Node.JS ( Studi Kasus : Sistem Pendaftaran Ujian Masuk Politeknik Negeri Ujung Pandang

- ),” *Semin. Nas. Tek. Elektro dan Inform.*, pp. 72–77, 2020.
- [11] D. Wijonarko and B. W. R. Mulya, “Pengembangan Antarmuka Pemrograman Aplikasi Menggunakan Metode RESTful pada Sistem Informasi Akademik Politeknik Kota Malang,” *Smatika J.*, vol. 8, no. 02, pp. 63–66, 2018, doi: 10.32664/smatika.v8i02.202.
- [12] A. B. Warsito, A. Ananda, and D. Triyanjaya, “Penerapan Data JSON Untuk Mendukung Pengembangan Aplikasi Pada Perguruan Tinggi Dengan Teknik Restfull Dan Web Service,” *Technomedia J.*, vol. 2, no. 1, pp. 26–36, 2017, doi: 10.33050/tmj.v2i1.313.
- [13] E. C. Foster and S. Godbole, *Database systems: A pragmatic approach*. 2016.
- [14] A. Prayitno and Y. Safitri, “Pemanfaatan Sistem Informasi Perpustakaan Digital Berbasis Website Untuk Para Penulis,” *JSE – Indones. J. Softw. Eng.*, vol. Volume 1 N, p. 10, 2015.
- [15] D. A. B. Prasetyo, “Implementasi Information Schema Database Pada PostgreSQL untuk Pembuatan Tabel Informasi dengan Menggunakan Python Di PT XYZ,” *JATISI (Jurnal Tek. Inform. dan Sist. Informasi)*, vol. 9, no. 3, pp. 1961–1972, 2022, doi: 10.35957/jatisi.v9i3.2221.
- [16] T. S. Waruwu, “Implementasi Postgresql sebagai Sistem Manajemen Basis Data pada Pendaftaran Mahasiswa Baru Berbasis Web,” *J. Mahajana Inf.*, vol. 4, no. 1, p. 59, 2019.
- [17] S. Ni Kadek Dwi, P. Dian, and M. K. Tubagus, “Implementation of Golang and ReactJS in the Covid-19 Vaccination Reservation System,” *Adi J. Recent Innov.*, vol. 5, no. 1, pp. 1–12, 2023, doi: 10.34306/ajri.v5i1.877.
- [18] Y. A. Permana and R. Puji, “Perancangan Sistem Informasi Penjualan Perumahan Menggunakan Metode SDLC Pada PT. Mandiri Land Prosperous Berbasis Mobile,” *Vol. 10 Nomor 2 Desember 2019 ISSN 2407-3903*, vol. 84, no. 10, pp. 1511–1518, 2019, doi: 10.1134/s0320972519100129.
- [19] M. Laya and B. S. Arifin, “Web Service Processor sebagai Penghubung Sistem Kiosk Medicom dengan SIM RS Kanker Dharmais,” *Multinetics*, vol. 3, no. 2, p. 49, 2017, doi: 10.32722/vol3.no2.2017.pp49-56.

- [20] O. C. Resmi Rachmawati, K.W. Deyana, M. F. Wifda, F. Arna, and R. Hestiasari, "Implementing Agile Scrum Methodology in The Development of SICITRA Mobile Application," *J. RESTI (Rekayasa Sist. dan Teknol. Informasi)*, vol. 7, no. 1, pp. 41–50, 2023, doi: 10.29207/resti.v7i1.4688.
- [21] F.- Sonata, "Pemanfaatan UML (Unified Modeling Language) Dalam Perancangan Sistem Informasi E-Commerce Jenis Customer-To-Customer," *J. Komunika J. Komunikasi, Media dan Inform.*, vol. 8, no. 1, p. 22, 2019, doi: 10.31504/komunika.v8i1.1832.
- [22] Haviluddin, "Memahami Penggunaan UML ( Unified Modelling Language )," *Memahami Pengguna. UML (Unified Model. Lang.*, vol. 6, no. 1, pp. 1–15, 2011, [Online]. Available: <https://informatikamulawarman.files.wordpress.com/2011/10/01-jurnal-informatika-mulawarman-feb-2011.pdf>.
- [23] K. Nistrina and L. Sahidah, "Unified Modelling Language (Uml) Untuk Perancangan Sistem Informasi Penerimaan Siswa Baru Di Smk Marga Insan Kamil," *J. Sist. Inf.*, vol. 04, pp. 12–23, 2022.
- [24] Fenny, "An Express-inspired web framework written in Go," *Creative Commons*, 2019. <https://gofiber.io/> (accessed Sep. 24, 2023).
- [25] D. Desmulyati and M. R. Perdana Putra, "Load Balance Design of Google Cloud Compute Engine VPS with Round Robin Method in PT. Lintas Data Indonesia," *Sinkron*, vol. 3, no. 2, p. 147, 2019, doi: 10.33395/sinkron.v3i2.10064.
- [26] I. Barokah and A. Asriyanik, "Analisis Perbandingan Serverless Computing Pada Google Cloud Platform," *J. Teknol. Inform. dan Komput.*, vol. 7, no. 2, pp. 169–187, 2021, doi: 10.37012/jtik.v7i2.662.
- [27] N. Ramsari and A. Ginanjar, "Implementasi Infrastruktur Server Berbasis Cloud Computing Untuk Web Service Berbasis Teknologi Google Cloud Platform," *Conf. Senat. STT Adisutjipto Yogyakarta*, vol. 7, 2022, doi: 10.28989/senatik.v7i0.472.
- [28] S. Daniel, E. R. Alix, and F. Hector, "Towards a Clean Architecture for Android Apps using Model Transformations," *IAENG Int. J. Comput. Sci.*, vol. 49, no. 1, 2022.

- [29] F. Amir, F. Oriyasmi, F. Fadilillah, H. Saputra, and A. Y. Street, "Design of Web-Based Point of Sale Information System at Inti Peraga Mandiri Inti Peraga Mandiri," no. November, pp. 17–19, 2022.
- [30] I. A. Dewi, Y. Miftahuddin, M. A. Fattah, C. B. Palenda, and S. F. Erawan, "Point of Sales System in InHome Café Website using Agile Methodology," *J. Innov. Community Engagem.*, vol. 1, no. 1, pp. 01–19, 2021, doi: 10.28932/jice.v1i1.3321.
- [31] B. M. Sidhunata *et al.*, "Point of Sales ( POS ) System Design using Design Thinking Framework for Motorcycle Workshop," vol. 5, no. 3, pp. 874–886, 2023, doi: 10.51519/journalisi.v5i3.515.
- [32] P. G. S. C. Nugraha, "Rancang Bangun Sistem Informasi Software Point of Sale (Pos) Dengan Metode Waterfall Berbasis Web," *JST (Jurnal Sains dan Teknol.*, vol. 10, no. 1, pp. 92–103, 2021, doi: 10.23887/jstundiksha.v10i1.29748.
- [33] S. Isnibaiti, I. Arwani, W. Hayuhardhika, and N. Putra, "Pengembangan Sistem Informasi Point of Sales (POS) berbasis Website untuk Manajemen Home Industry (Studi Kasus: Gelsey Real Surakarta)," vol. 6, no. 4, pp. 1789–1796, 2022, [Online]. Available: <http://j-ptiik.ub.ac.id>.