



**ANALISIS DAN PERANCANGAN JASA LAYANAN FIRMA HUKUM  
BERBASIS *WEB* MENGGUNAKAN *FRAMEWORK LARAVEL***

**SKRIPSI**

**Oleh :**

Roberto Stephen  
20210700004

Program Studi Sistem Informasi  
Fakultas Sains dan Teknologi

**UNIVERSITAS BUDDHI DHARMA  
TANGERANG**

**2025**



**ANALISIS DAN PERANCANGAN JASA LAYANAN FIRMA HUKUM  
BERBASIS *WEB* MENGGUNAKAN *FRAMEWORK LARAVEL***

**SKRIPSI**

Diajukan sebagai salah satu syarat untuk mendapatkan gelar Sarjana Komputer  
pada Program Studi Sistem Informasi Fakultas Sains dan Teknologi  
Universitas Buddhi Dharma  
Jenjang Pendidikan Strata 1

**Oleh :**

Roberto Stephen  
20210700004

Program Studi Sistem Informasi  
Fakultas Sains dan Teknologi

**UNIVERSITAS BUDDHI DHARMA  
TANGERANG**

**2025**

## LEMBAR PERSEMBAHAN

**Na kāmatarante mahāntaḥ sveṣu karmasu**

*“Orang-orang hebat tidak pernah menunda tugasnya”*

Dengan mengucapkan puji syukur kepada Tuhan Yang Maha Esa, Skripsi ini kupersembahkan untuk:

1. Papa dan Mama tercinta yang telah membesarkan aku dan selalu membimbing, mendukung, memotivasi, memberi apa yang terbaik bagiku serta selalu mendoakan aku untuk meraih suksesku.
2. Adikku yang telah memberikan dukungan semangat serta dorongan yang senantiasa diberikan.
3. Teman-teman kelompok belajar yang selalu berjuang bersama.
4. Rekan-rekan dari William Widjaja & Partners Lawfirm yang selalu memberikan dukungan dan semangat.

**UNIVERSITAS BUDDHI DHARMA**  
**LEMBAR PERNYATAAN KEASLIAN SKRIPSI**

Yang bertanda tangan dibawah ini.

NIM : 20210700004  
Nama : Roberto Stephen  
Jenjang Studi : Strata I  
Program Studi : Sistem Informasi  
Peminatan : E-Business

Dengan ini saya menyatakan bahwa:

1. Laporan Skripsi ini adalah asli dan belum pernah diajukan untuk mendapat gelar akademik Sarjana atau kelengkapan studi, baik di Universitas Buddhi Dharma maupun di Perguruan Tinggi lainnya.
2. Laporan Skripsi ini saya buat sendiri tanpa bantuan dari pihak lain, kecuali arahan dosen pembimbing.
3. Dalam Laporan Skripsi ini tidak terdapat karya atau pendapat yang telah ditulis atau dipublikasikan orang lain, kecuali secara tertulis dengan jelas dan dicantumkan sebagai acuan dalam naskah dengan disebutkan nama pengarang dan dicantumkan daftar pustaka.
4. Dalam Laporan Skripsi ini tidak terdapat pemalsuan (kebohongan) seperti buku, artikel, jurnal, data sekunder, pengolahan data, dan pemalsuan tanda tangan dosen atau Ketua Program Studi Universitas Buddhi Dharma yang dibuktikan dengan keasliannya.
5. Lembar pernyataan ini saya buat dengan sesungguhnya, tanpa paksaan dan apabila dikemudian hari atau pada waktu lainnya terdapat penyimpangan dan ketidak benaran dalam pernyataan ini, saya bersedia menerima sanksi akademik berupa pencabutan gelar akademik yang telah saya peroleh karena Laporan Skripsi ini serta sanksi lainnya sesuai dengan peraturan dan norma yang berlaku.

Tangerang, 06-08-2025

Yang membuat pernyataan,

  
  
Roberto Stephen  
20210700004

## UNIVERSITAS BUDDHI DHARMA

### LEMBAR PERSETUJUAN PUBLIKASI KARYA ILMIAH

Yang bertanda tangan dibawah ini.

NIM : 20210700004  
Nama : Roberto Stephen  
Jenjang Studi : Strata I  
Program Studi : Sistem Informasi  
Peminatan : E-Business

Dengan ini menyetujui untuk memberikan izin kepada pihak Universitas Buddhi Dharma, Hak Bebas Royalti Non – Eksklusif (Non-exclusive Royalty-Free Right) atas karya ilmiah kami yang berjudul: “ANALISIS DAN PERANCANGAN JASA LAYANAN FIRMA HUKUM BERBASIS *WEB* MENGGUNAKAN *FRAMEWORK LARAVEL*”, beserta alat yang diperlukan (apabila ada). Dengan Hak Bebas Royalti Non-Eksklusif ini pihak Universitas Buddhi Dharma berhak menyimpan, mengalih-media atau format-kan, mengelolanya dalam pangkalan data (database), mendistribusikannya, mendistribusikannya di internet atau media lain untuk kepentingan akademis tanpa perlu meminta izin dari saya selama tetap mencantumkan nama saya sebagai penulis atau pencipta karya ilmiah tersebut. Saya bersedia untuk menanggung secara pribadi, tanpa melibatkan pihak Universitas Buddhi Dharma, segala bentuk tuntutan hukum yang timbul atas pelanggaran Hak Cipta dalam karya ilmiah saya ini.

Demikian pernyataan ini saya buat dengan sebenarnya.

Tangerang, 06-08-2025

Yang membuat pernyataan,



Roberto Stephen

**UNIVERSITAS BUDDHI DHARMA**

**LEMBAR PENGESAHAN PEMBIMBING**

**ANALISIS DAN PERANCANGAN JASA LAYANAN FIRMA HUKUM BERBASIS**

***WEB MENGGUNAKAN FRAMEWORK LARAVEL***

Dibuat Oleh:

NIM : 20210700004

Nama : Roberto Stephen

Telah disetujui untuk dipertahankan di hadapan Tim Penguji Ujian Komprehensif

Program Studi Sistem Informasi

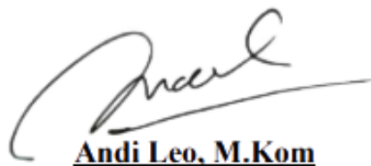
Peminatan E-Bisnis

Tahun Akademik 2024/2025

Tangerang, 10 Juli 2025

Disahkan Oleh,

**Pembimbing**



**Andi Leo, M.Kom**

NUPTK : 9337740641137083

**UNIVERSITAS BUDDHI DHARMA**

**LEMBAR PENGESAHAN SKRIPSI**

**ANALISIS DAN PERANCANGAN JASA LAYANAN FIRMA HUKUM BERBASIS**

***WEB MENGGUNAKAN FRAMEWORK LARAVEL***

Dibuat Oleh:

NIM : 20210700004

Nama : Roberto Stephen

Telah disetujui untuk dipertahankan di hadapan Tim Penguji Ujian Komprehensif

Program Studi Sistem Informasi

Peminatan E-Bisnis

Tahun Akademik 2024/2025

Tangerang, 06-08-2025

Disahkan Oleh,

**Dekan,**



Dr. Yakub, M.Kom., M.M.

NUPTK: 1836747648130172

**Ketua Program Studi,**



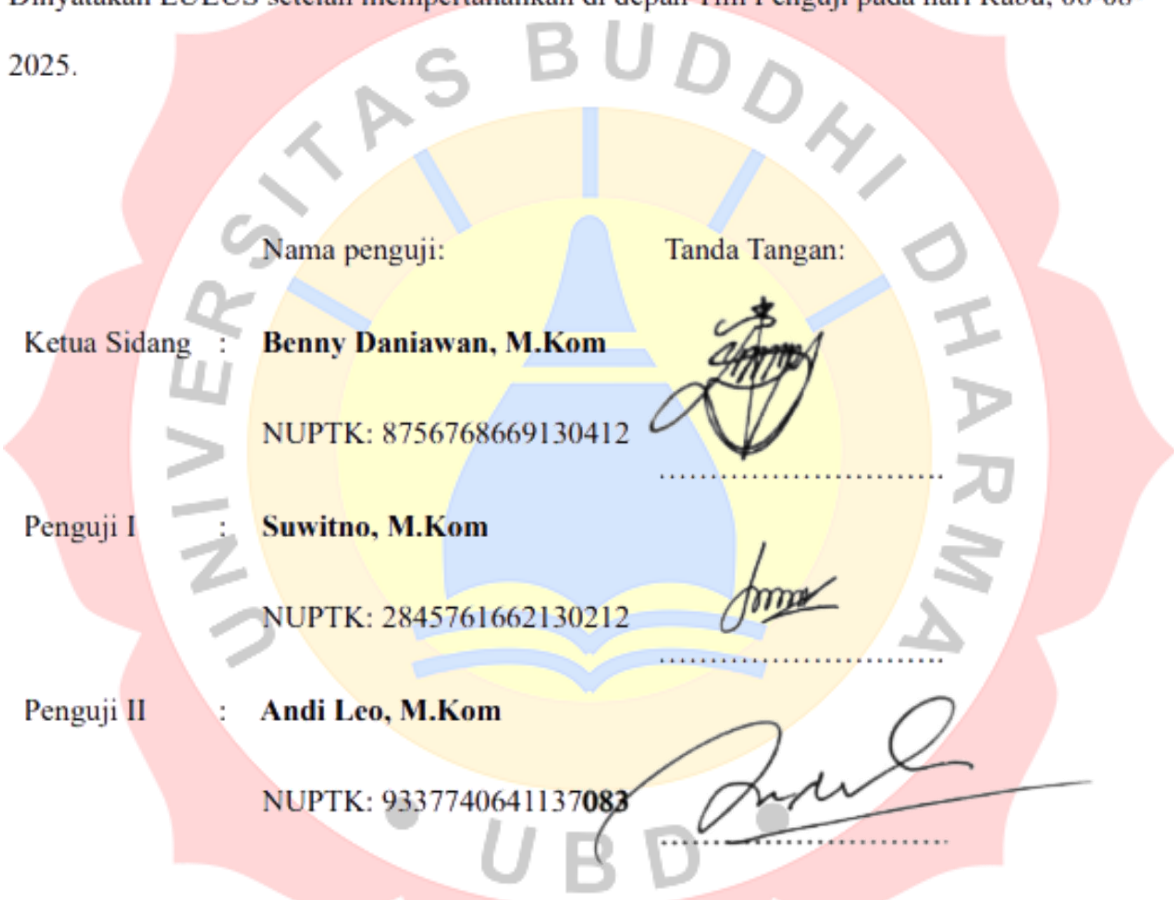
Benny Daniawan, M.Kom

NUPTK: 8756768669130412

## LEMBAR PENGESAHAN TIM PENGUJI

Nama : Roberto Stephen  
NIM : 20210700004  
Fakultas : Sains dan Teknologi  
Judul Skripsi : ANALISIS DAN PERANCANGAN JASA LAYANAN HUKUM  
BERBASIS *WEB* MENGGUNAKAN *FRAMEWORK LARAVEL*

Dinyatakan LULUS setelah mempertahankan di depan Tim Penguji pada hari Rabu, 06-08-2025.



	Nama penguji:	Tanda Tangan:
Ketua Sidang :	<b>Benny Daniawan, M.Kom</b> NUPTK: 8756768669130412	
Penguji I :	<b>Suwitno, M.Kom</b> NUPTK: 2845761662130212	
Penguji II :	<b>Andi Leo, M.Kom</b> NUPTK: 9337740641137083	

Mengetahui,

**Dekan Fakultas Sains dan Teknologi**



Dr. Yakub, M.Kom, M.M

NUPTK: 1836747648130172

## KATA PENGANTAR

Dengan mengucapkan Puji Syukur kepada Tuhan Yang Maha Esa, yang telah memberikan Rahmat dan karunia-Nya kepada penulis sehingga dapat menyusun dan menyelesaikan Skripsi ini dengan judul **ANALISIS DAN PERANCANGAN JASA LAYANAN FIRMA HUKUM BERBASIS *WEB* MENGGUNAKAN *FRAMEWORK LARAVEL***. Tujuan utama dari pembuatan Skripsi ini adalah sebagai salah satu syarat kelengkapan dalam menyelesaikan Program Sarjana Program Studi Sistem Informasi di Universitas Buddhi Dharma. Dalam penyusunan Skripsi ini Saya banyak menerima bantuan dan dorongan baik moril maupun materiil dari berbagai pihak, maka pada kesempatan ini penulis menyampaikan rasa terima kasih yang sebesar-besarnya kepada:

1. Ibu Dr. Limajatini, S.E., M.M., B.K.P. sebagai Rektor Universitas Buddhi Dharma.
2. Bapak Dr. Yakub, M.Kom., M.M. Dekan Fakultas Sains dan Teknologi.
3. Bapak Benny Daniawan, M.Kom., sebagai Ketua Program Studi Sistem Informasi
4. Bapak Andi Leo, S.Kom., M.Kom., sebagai pembimbing yang telah membantu dan memberikan dukungan serta harapan untuk menyelesaikan penulisan Skripsi ini.
5. Orang tua dan keluarga yang selalu memberikan dukungan baik moril dan materiil.
6. Pak William Widjaja sebagai pemilik William Widjaja & Partners Lawfirm.
7. Teman-teman yang selalu membantu dan memberikan semangat.
8. Yovinda Ardelia Wangsa yang selalu membantu dan memberikan semangat.

Serta semua pihak yang terlalu banyak untuk disebutkan satu-persatu sehingga terwujudnya penulisan ini. Saya menyadari bahwa penulisan Skripsi ini masih belum sempurna, untuk itu Saya mohon kritik dan saran yang bersifat membangun demi kesempurnaan penulisan di masa yang akan datang. Akhir kata semoga Skripsi ini dapat berguna bagi Saya khususnya dan bagi para pembaca yang berminat pada umumnya.

Tangerang, 06-08-2025

Penulis

# ANALISIS DAN PERANCANGAN JASA LAYANAN FIRMA HUKUM BERBASIS WEB MENGGUNAKAN *FRAMEWORK LARAVEL*.

239 Halaman + xix / 42 Tabel / 134 Gambar / 10 Lampiran

## ABSTRAK

Era digital telah membawa perubahan besar dalam berbagai sektor, termasuk layanan hukum. Penelitian ini bertujuan untuk menganalisis dan merancang sistem layanan firma hukum berbasis *web* yang memanfaatkan *framework Laravel* sebagai basis pengembangannya. Dengan pendekatan arsitektur *Model-View-Controller (MVC)*, penelitian ini mengevaluasi kebutuhan pengguna dan mengidentifikasi fitur-fitur penting yang harus dimiliki sistem, seperti konsultasi hukum *online*, manajemen kasus, dan sistem penjadwalan otomatis. Metodologi penelitian menggunakan pendekatan *Agile* dengan empat sprint selama delapan minggu, yang memungkinkan pengembangan iteratif dan responsif terhadap perubahan kebutuhan. Analisis kebutuhan sistem dilakukan menggunakan *PIECES Framework* terhadap 20 responden yang terdiri dari 10 staff firma hukum, 5 klien, dan 5 praktisi hukum eksternal. Perancangan sistem menggunakan *Unified Modeling Language (UML)* meliputi *use case diagram*, *activity diagram*, *sequence diagram*, dan *class diagram*. Hasil penelitian menunjukkan bahwa penerapan arsitektur *MVC* dalam *framework Laravel* memungkinkan pengelolaan logika bisnis, antarmuka pengguna, dan data secara terorganisir. Analisis *PIECES* menunjukkan tingkat kepuasan pengguna rata-rata 3.93 (kategori puas) dengan *domain efficiency* memperoleh skor tertinggi 4.03. Prototipe yang dihasilkan berhasil diuji menggunakan metode *BlackBox* dan *User Acceptance Testing (UAT)* terhadap 10 responden dengan tingkat penerimaan 92.3%, menunjukkan kemampuan sistem untuk meningkatkan efektivitas operasional firma hukum sekaligus memberikan pengalaman pengguna yang lebih baik bagi klien. Sistem ini berhasil menyediakan fitur katalog layanan, manajemen pesanan, dan dashboard administratif yang terintegrasi untuk mendukung digitalisasi layanan hukum.

**Kata Kunci:** Manajemen Kasus Hukum, *Framework Laravel*, Firma Hukum, MVC

# ***ANALYSIS AND DESIGN OF WEB-BASED LEGAL FIRM SERVICES USING THE LARAVEL FRAMEWORK***

***239 Pages + xix / 42 Tables / 134 Images / 10 References***

## ***ABSTRACT***

*The digital era has brought significant changes across various sectors, including legal services. This study aims to analyze and design a web-based legal firm service system utilizing the Laravel framework as its development foundation. By adopting the Model-View-Controller (MVC) architecture approach, this research evaluates user requirements and identifies essential system features, such as online legal consultations, case management, and automated scheduling systems. The research methodology employs an Agile approach with four sprints over eight weeks, enabling iterative development and responsiveness to changing requirements. System requirements analysis was conducted using the PIECES Framework involving 20 respondents consisting of 10 legal firm staff, 5 clients, and 5 external legal practitioners. System design utilizes Unified Modeling Language (UML) including use case diagrams, activity diagrams, sequence diagrams, and class diagrams. The research findings indicate that implementing MVC architecture within the Laravel framework enables organized management of business logic, user interfaces, and data. PIECES analysis shows an average user satisfaction level of 3.93 (satisfied category) with the efficiency domain achieving the highest score of 4.03. The resulting prototype was successfully tested using BlackBox methods and User Acceptance Testing (UAT) involving 10 respondents with an acceptance rate of 92.3%, demonstrating the system's capability to enhance legal firm operational effectiveness while providing a better user experience for clients. This system successfully provides integrated service catalog features, order management, and administrative dashboard to support legal service digitalization.*

***Keyword:*** Case Management System, Laravel Framework, Law Firm, MVC

## DAFTAR ISI

<b>LEMBAR PERSEMBAHAN .....</b>	<b>i</b>
<b>LEMBAR PERNYATAAN KEASLIAN SKRIPSI.....</b>	<b>ii</b>
<b>LEMBAR PERSETUJUAN PUBLIKASI KARYA ILMIAH.....</b>	<b>iii</b>
<b>LEMBAR PENGESAHAN PEMBIMBING.....</b>	<b>iv</b>
<b>LEMBAR PENGESAHAN SKRIPSI.....</b>	<b>v</b>
<b>LEMBAR PENGESAHAN TIM PENGUJI .....</b>	<b>vi</b>
<b>KATA PENGANTAR .....</b>	<b>vii</b>
<b>ABSTRAK.....</b>	<b>viii</b>
<b><i>ABSTRACT</i>.....</b>	<b>ix</b>
<b>DAFTAR ISI .....</b>	<b>x</b>
<b>DAFTAR TABEL.....</b>	<b>xiii</b>
<b>DAFTAR GAMBAR.....</b>	<b>xv</b>
<b>DAFTAR LAMPIRAN.....</b>	<b>xix</b>
<b>BAB I PENDAHULUAN .....</b>	<b>1</b>
1.1 Latar Belakang Masalah .....	1
1.2 Identifikasi Masalah.....	3
1.3 Ruang Lingkup Penelitian .....	4
1.4 Tujuan Dan Manfaat Penelitian .....	6
1.4.1 Tujuan Penelitian.....	6
1.4.2 Manfaat Penelitian.....	6
1.5 Sistematika Penulisan .....	7
<b>BAB II LANDASAN TEORI.....</b>	<b>9</b>
2.1 Analisis.....	9
2.2 Perancangan .....	9
2.3 Sistem Informasi .....	10
2.4 Informasi.....	10
2.5 Data .....	11
2.6 Firma Hukum.....	11
2.7 Jasa.....	12
2.8 Layanan.....	13
2.9 Internet .....	13
2.10 <i>Website</i> .....	14

2.11 <i>Online</i> .....	14
2.12 <i>Database</i> .....	15
2.13 <i>Framework</i> .....	15
2.14 <i>Laravel</i> .....	16
2.15 <i>XAMPP</i> .....	17
2.16 Metodologi <i>Agile</i> .....	17
2.17 <i>Use case</i> Diagram .....	19
2.18 <i>Activity</i> Diagram .....	21
2.19 Analisis Sistem.....	23
2.20 <i>Class</i> Diagram (Kelas Diagram).....	24
2.21 <i>Sequence</i> Diagram (Diagram Sekuen).....	26
2.21 Tinjauan Jurnal.....	28
<b>BAB III METODOLOGI PENELITIAN</b> .....	<b>32</b>
3.1 Teknik Pengumpulan Data Penelitian.....	32
3.1.1 Observasi Langsung .....	32
3.2 Metodologi Penelitian.....	33
3.2.1 Metodologi <i>Agile</i> .....	33
3.2.2 Langkah-langkah Implementasi <i>Agile</i> dalam Penelitian Ini .....	33
3.2.3 Manfaat Metodologi <i>Agile</i> .....	35
3.2.4 <i>Sprint</i> Cycle Diagram.....	36
3.2.5 <i>Kanban</i> Board Diagram.....	37
3.2.6 <i>Burndown</i> Chart .....	38
3.2.7 <i>User Story</i> Mapping .....	39
3.2.8 <i>Product Backlog</i> Diagram .....	40
3.3 Analisis Masalah .....	41
3.3.1 <i>PIECES Framework</i> .....	41
3.4 Requirement Elicitation .....	46
3.4.1 Tahap 1 .....	47
3.4.2 Tahap 2 .....	48
3.4.3 Tahap Final .....	49
3.5 Jadwal Penelitian .....	50
3.6 Kerangka Pemikiran .....	53
3.7 Prosedur Sistem Usulan .....	54
3.8 Rancangan Sistem Usulan .....	55
3.8.1 <i>Use Case</i> Diagram.....	55

3.8.2 <i>Use Case Diagram Scenario Admin, Customer</i> .....	58
3.8.3 <i>Activity Diagram</i> .....	65
3.8.4 <i>Sequence Diagram</i> .....	82
3.8.5 <i>Class Diagram</i> .....	105
3.8.6 <i>Rancangan Database</i> .....	106
3.8.7 <i>Desain Rancangan Sistem</i> .....	111
<b>BAB IV HASIL DAN PEMBAHASAN</b> .....	<b>152</b>
4.1 <i>Implementasi Sistem</i> .....	152
4.2 <i>Spesifikasi Software dan Hardware</i> .....	195
4.2.2 <i>Software</i> .....	195
4.2.2 <i>Hardware</i> .....	196
4.3 <i>Pengujian Blackbox</i> .....	197
4.3.1 <i>Pengujian Frontend (Customer)</i> .....	197
4.3.1 <i>Pengujian Backend (Admin)</i> .....	200
4.4 <i>UAT (User Acceptance Testing)</i> .....	204
4.4.1 <i>Hasil Analisis UAT</i> .....	208
<b>BAB V SIMPULAN DAN SARAN</b> .....	<b>215</b>
5.1 <i>Simpulan</i> .....	215
5.2 <i>Saran</i> .....	216
<b>DAFTAR PUSTAKA</b> .....	<b>217</b>
<b>DAFTAR RIWAYAT HIDUP</b> .....	<b>222</b>
<b>LAMPIRAN</b> .....	<b>223</b>

## DAFTAR TABEL

Tabel 2. 1 Simbol - Simbol Activity Diagram .....	22
Tabel 2. 2 Simbol - Simbol Use Case Diagram .....	20
Tabel 2. 3 Simbol - Simbol Class Diagram .....	24
Tabel 2. 4 Simbol - Simbol Sequence Diagram.....	26
Tabel 3. 1 Jawaban Responden Terhadap Variabel Performance .....	41
Tabel 3. 2 Jawaban Responden terhadap Information and Data.....	42
Tabel 3. 3 Jawaban Responden terhadap Variabel Economics .....	43
Tabel 3. 4 Jawaban Responden terhadap variabel Control and Security .....	44
Tabel 3. 5 Jawaban Responden terhadap Variabel Efficiency .....	45
Tabel 3. 6 Jawaban Responden terhadap variabel Service .....	45
Tabel 3. 7 Nilai Skor Rekap Variabel PIECES .....	46
Tabel 3. 8 Elicitation Tahap 1 .....	47
Tabel 3. 9 Elicitation Tahap 2 .....	48
Tabel 3. 10 Elicitation Tahap 3 .....	49
Tabel 3. 11 Prosedur Sistem Usulan .....	54
Tabel 3. 12 Scenario Login.....	58
Tabel 3. 13 Scenario Logout.....	58
Tabel 3. 14 Scenario Kelola Dashboard .....	59
Tabel 3. 15 Scenario Browse Service .....	59
Tabel 3. 16 Scenario Sort Services .....	60
Tabel 3. 17 Scenario Checkout.....	60
Tabel 3. 18 Scenario Add to Cart.....	61
Tabel 3. 19 Scenario Track Orders .....	61
Tabel 3. 20 Scenario Manage Wishlist .....	62
Tabel 3. 21 Scenario View Service Details.....	62
Tabel 3. 22 Scenario Apply Coupon.....	63
Tabel 3. 24 Struktur Data Addresses .....	106
Tabel 3. 25 Struktur Data <i>Customers</i> .....	107
Tabel 3. 26 Struktur Data Carts .....	107
Tabel 3. 27 Struktur Data Orders .....	108
Tabel 3. 28 Struktur Data Order_items.....	109
Tabel 3. 29 Struktur Data Services .....	109

Tabel 3. 30 Struktur Data Categories.....	110
Tabel 4. 1 Tabel <i>Software</i> Development Environment .....	195
Tabel 4. 2 Tabel <i>Software</i> Frontend Technologies .....	195
Tabel 4. 3 Tabel <i>Hardware</i> Development Machine .....	196
Tabel 4. 4 Tabel <i>Hardware</i> Production Server .....	196
Tabel 4. 5 Tabel <i>Hardware Deployment</i> Considerations .....	196
Tabel 4. 6 Tabel Pengujian <i>Blackbox</i> Frontend <i>Customer</i> .....	197
Tabel 4. 7 Tabel Pengujian <i>Blackbox</i> Backend <i>Admin</i> .....	200
Tabel 4. 8 <i>UAT</i> .....	204



## DAFTAR GAMBAR

Gambar 3. 1 Sprint Cycle Diagram .....	36
Gambar 3. 2 Kanban Board Diagram .....	37
Gambar 3. 3 Burndown Chart.....	38
Gambar 3. 4 User Story Mapping Diagram.....	39
Gambar 3. 5 Product <i>Backlog</i> Diagram.....	40
Gambar 3. 6 Gantt Chart.....	52
Gambar 3. 7 Use Case Diagram .....	55
Gambar 3. 8 Activity Diagram <i>Admin Login</i> .....	65
Gambar 3. 9 Activity Diagram Kelola Dashboard .....	66
Gambar 3. 11 Activity Diagram View Total Sales.....	67
Gambar 3. 12 Activity Diagram Kelola Sales .....	68
Gambar 3. 13 Activity Diagram Kelola Katalog .....	69
Gambar 3. 14 Activity Diagram Kelola Marketing .....	70
Gambar 3. 15 Activity Diagram Generate Reports.....	71
Gambar 3. 16 Activity Diagram Logout.....	72
Gambar 3. 17 Activity Diagram Register .....	73
Gambar 3. 18 Activity Diagram Login.....	74
Gambar 3. 19 Activity Diagram Browse Service .....	75
Gambar 3. 20 Activity Diagram Sort Services .....	76
Gambar 3. 21 Activity Diagram Add to Cart.....	77
Gambar 3. 22 Activity Diagram Manage Cart.....	78
Gambar 3. 23 Activity Diagram Manage Wishlist .....	79
Gambar 3. 24 Activity Diagram View Service Details.....	80
Gambar 3. 25 Activity Diagram Logout.....	81
Gambar 3. 26 Sequence Diagram Login .....	82
Gambar 3. 27 Sequence Diagram Kelola Dashboard .....	83
Gambar 3. 29 Sequence Diagram View Total Sales .....	84
Gambar 3. 30 Sequence Diagram Kelola Sales .....	85
Gambar 3. 31 Sequence Diagram Kelola Katalog.....	86
Gambar 3. 32 Sequence Diagram Kelola Marketing.....	87
Gambar 3. 33 Sequence Diagram Generate Reports .....	88
Gambar 3. 34 Sequence Diagram Logout .....	90

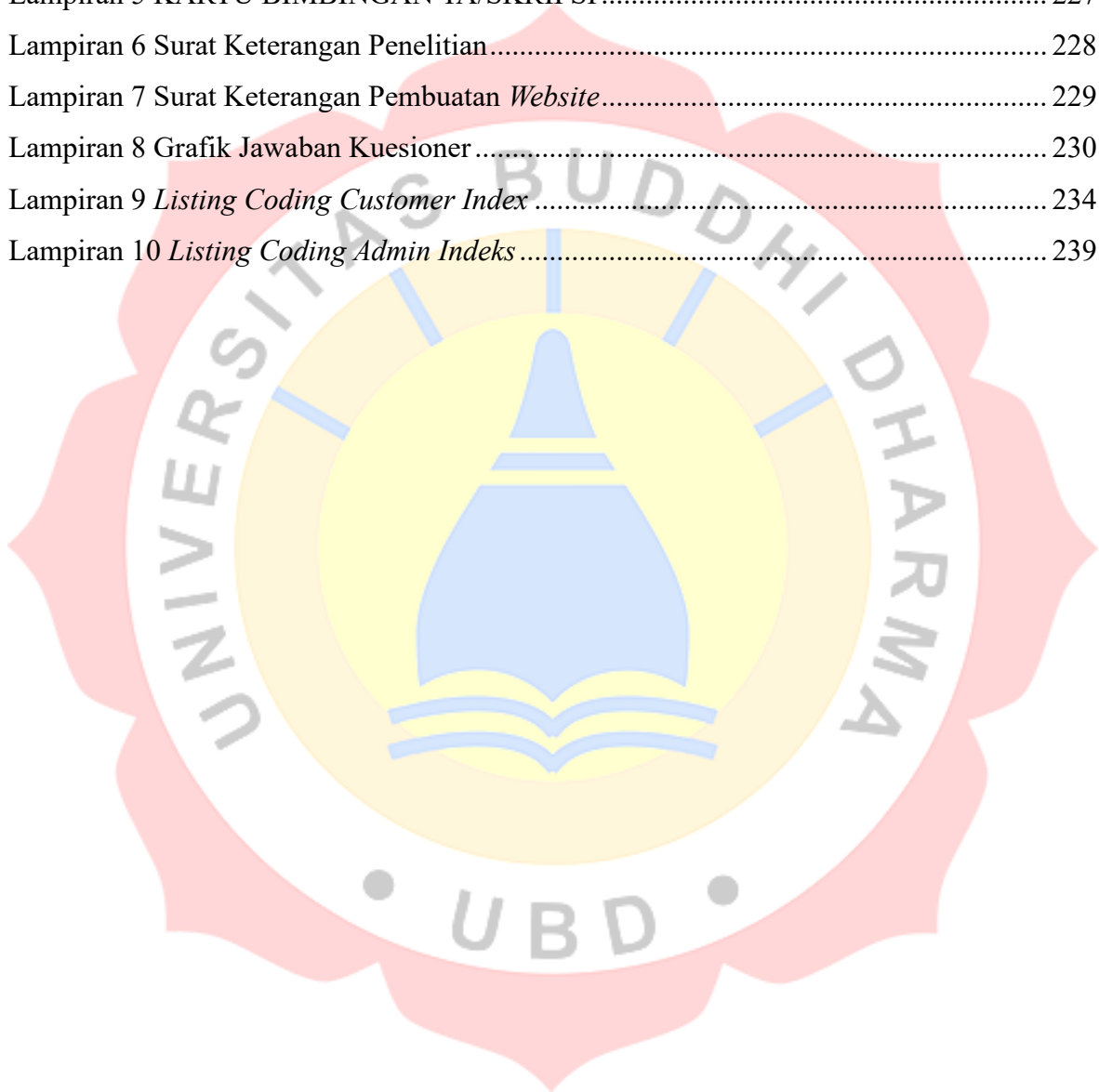
Gambar 3. 35 Sequence Diagram Register.....	91
Gambar 3. 36 Sequence Diagram Login .....	92
Gambar 3. 37 Sequence Diagram Browse Service.....	94
Gambar 3. 38 Sequence Diagram Sort Services.....	95
Gambar 3. 39 Sequence Diagram Filter Services.....	96
Gambar 3. 40 Sequence Diagram Checkout.....	97
Gambar 3. 41 Sequence Diagram Apply Coupon.....	98
Gambar 3. 42 Sequence Diagram Track Orders .....	99
Gambar 3. 43 Sequence Diagram Add to Cart .....	100
Gambar 3. 44 Sequence Diagram Manage Cart .....	101
Gambar 3. 45 Sequence Diagram Manage Wishlist.....	102
Gambar 3. 46 Sequence Diagram View Service Details .....	103
Gambar 3. 47 Sequence Diagram Logout .....	104
Gambar 3. 48 Class Diagram.....	105
Gambar 3. 49 Tampilan Login <i>Customer</i> .....	111
Gambar 3. 50 Tampilan Register <i>Customer</i> .....	112
Gambar 3. 51 Tampilan Profile <i>Customer</i> .....	113
Gambar 3. 52 Tampilan Dashboard <i>Customer</i> .....	114
Gambar 3. 53 Tampilan Service .....	115
Gambar 3. 54 Tampilan Katalog Service.....	116
Gambar 3. 55 Tampilan Our Client .....	117
Gambar 3. 56 Tampilan Footer.....	118
Gambar 3. 57 Tampilan Dashboard <i>Admin</i> .....	119
Gambar 3. 58 Tampilan Account Profile <i>Admin</i> .....	120
Gambar 3. 59 Tampilan Order <i>Admin</i> .....	121
Gambar 3. 60 Tampilan Shipments <i>Admin</i> .....	122
Gambar 3. 61 Tampilan Invoice <i>Admin</i> .....	123
Gambar 3. 62 Tampilan Refunds <i>Admin</i> .....	124
Gambar 3. 63 Tampilan Transactions <i>Admin</i> .....	125
Gambar 3. 64 Tampilan Create Transactions <i>Admin</i> .....	126
Gambar 3. 65 Tampilan Catalog Services <i>Admin</i> .....	127
Gambar 3. 66 Tampilan Create Services <i>Admin</i> .....	128
Gambar 3. 67 Tampilan Catalog Categories <i>Admin</i> .....	129
Gambar 3. 68 Tampilan Add Categories <i>Admin</i> .....	130

Gambar 3. 69 Lanjutan Add Categories <i>Admin</i> .....	131
Gambar 3. 70 Tampilan <i>Customers Admin</i> .....	132
Gambar 3. 71 Tampilan Create New <i>Customer</i> .....	133
Gambar 3. 72 Tampilan Groups <i>Admin</i> .....	134
Gambar 3. 73 Create New Group.....	135
Gambar 3. 74 Tampilan <i>Reviews Admin</i> .....	136
Gambar 3. 75 Tampilan CMS Promotions <i>Admin</i> .....	137
Gambar 3. 76 Tampilan Create Catalog Rule <i>Admin</i> .....	138
Gambar 3. 77 Tampilan Communications <i>Admin</i> .....	139
Gambar 3. 78 Tampilan Create Template <i>Admin</i> .....	139
Gambar 3. 79 Tampilan Search & SEO <i>Admin</i> .....	140
Gambar 3. 80 Tampilan Create URL Rewrite.....	140
Gambar 3. 81 Tampilan Reporting Sales <i>Admin</i> .....	141
Gambar 3. 82 Tampilan Reporting <i>Customers</i> .....	142
Gambar 3. 83 Tampilan Reporting Services.....	143
Gambar 3. 84 Tampilan Settings Locales.....	144
Gambar 3. 85 Tampilan Create Locale.....	145
Gambar 3. 86 Tampilan Settings Users.....	146
Gambar 3. 87 Tampilan Create User.....	147
Gambar 3. 88 Tampilan Settings Roles.....	148
Gambar 3. 89 Tampilan Create Role.....	149
Gambar 3. 90 Tampilan Tax Categories.....	150
Gambar 3. 91 Tampilan Create Tax Categories.....	151
Gambar 4. 1 Tampilan Login <i>Customer</i> .....	152
Gambar 4. 2 Tampilan Register <i>Customer</i> .....	153
Gambar 4. 3 Tampilan Profile <i>Customer</i> .....	154
Gambar 4. 4 Tampilan Dashboard <i>Customer</i> .....	155
Gambar 4. 5 Tampilan Dashboard <i>Customer</i> .....	156
Gambar 4. 6 Tampilan Katalog Service.....	157
Gambar 4. 7 Tampilan Our Client.....	158
Gambar 4. 8 Tampilan Footer.....	159
Gambar 4. 9 Tampilan Dashboard <i>Admin</i> .....	160
Gambar 4. 10 Tampilan Account Profile <i>Admin</i> .....	161
Gambar 4. 11 Tampilan Order <i>Admin</i> .....	162

Gambar 4. 12 Tampilan Shipments <i>Admin</i> .....	163
Gambar 4. 13 Tampilan Invoice <i>Admin</i> .....	164
Gambar 4. 14 Tampilan Refunds <i>Admin</i> .....	165
Gambar 4. 15 Tampilan Transactions <i>Admin</i> .....	166
Gambar 4. 16 Tampilan Create Transactions <i>Admin</i> .....	167
Gambar 4. 17 Tampilan Catalog Products <i>Admin</i> .....	168
Gambar 4. 18 Tampilan Create Services <i>Admin</i> .....	169
Gambar 4. 19 Tampilan Catalog Categories <i>Admin</i> .....	170
Gambar 4. 20 Tampilan Add Categories <i>Admin</i> .....	171
Gambar 4. 21 Tampilan Lanjutan Add Categories <i>Admin</i> .....	172
Gambar 4. 22 Tampilan <i>Customers Admin</i> .....	173
Gambar 4. 23 Tampilan Create New <i>Customer</i> .....	174
Gambar 4. 24 Tampilan Groups <i>Admin</i> .....	175
Gambar 4. 25 Tampilan Create New Group .....	176
Gambar 4. 26 Tampilan <i>Reviews Admin</i> .....	177
Gambar 4. 27 Tampilan CMS Promotions <i>Admin</i> .....	178
Gambar 4. 28 Tampilan Create Catalog Rule <i>Admin</i> .....	179
Gambar 4. 29 Tampilan Communications <i>Admin</i> .....	180
Gambar 4. 30 Tampilan Create Template <i>Admin</i> .....	181
Gambar 4. 31 Tampilan Search & SEO <i>Admin</i> .....	182
Gambar 4. 32 Tampilan Create URL Rewrite .....	183
Gambar 4. 33 Tampilan Reporting Sales <i>Admin</i> .....	184
Gambar 4. 34 Tampilan Reporting <i>Customers</i> .....	185
Gambar 4. 35 Tampilan Reporting Services.....	186
Gambar 4. 36 Tampilan Settings Locale.....	187
Gambar 4. 37 Tampilan Create Locale .....	188
Gambar 4. 38 Tampilan Settings User.....	189
Gambar 4. 39 Tampilan Create User .....	190
Gambar 4. 40 Tampilan Settings Roles .....	191
Gambar 4. 41 Tampilan Create Role .....	192
Gambar 4. 42 Tampilan Tax Categories .....	193
Gambar 4. 43 Tampilan Create Tax Categories .....	194

## DAFTAR LAMPIRAN

Lampiran 1 <i>Requirement Elicitation 1</i> .....	223
Lampiran 2 <i>Requirement Elicitation 2</i> .....	224
Lampiran 3 <i>Requirement Elicitation 3</i> .....	224
Lampiran 4 <i>Requirement Elicitation 4</i> .....	226
Lampiran 5 KARTU BIMBINGAN TA/SKRIPSI .....	227
Lampiran 6 Surat Keterangan Penelitian .....	228
Lampiran 7 Surat Keterangan Pembuatan <i>Website</i> .....	229
Lampiran 8 Grafik Jawaban Kuesioner .....	230
Lampiran 9 <i>Listing Coding Customer Index</i> .....	234
Lampiran 10 <i>Listing Coding Admin Indeks</i> .....	239



# BAB I

## PENDAHULUAN

### 1.1 Latar Belakang Masalah

Perkembangan teknologi yang semakin pesat memiliki dampak yang signifikan bagi berbagai sektor, termasuk sektor hukum. Teknologi yang terus berkembang bertujuan untuk mempermudah aktivitas manusia dalam memproses, mengelola, dan menganalisis data maupun informasi (Siregar & Nasution, 2020). Transformasi digital dalam sektor hukum telah menjadi kebutuhan mendesak, terutama pasca pandemi *COVID-19* yang mendorong percepatan adopsi teknologi di berbagai bidang (Anjar Setiarma, 2023). Tidak hanya di dunia bisnis, instansi pemerintahan hingga firma hukum juga mengalami perubahan signifikan dalam proses pengolahan data dan pemberian layanan kepada klien.

Dalam dunia hukum, efektivitas dan akurasi dalam memberikan layanan menjadi faktor krusial. Proses *administrasi* yang masih dilakukan secara manual sering kali menghambat efektivitas firma hukum dalam menangani kasus dan memberikan konsultasi kepada klien (Muriyatmoko et al. 2020). Penelitian menunjukkan bahwa digitalisasi layanan hukum dapat meningkatkan aksesibilitas klien terhadap informasi hukum dan mempercepat proses penyelesaian kasus (Aziz, 2021). Oleh karena itu, pemanfaatan teknologi berbasis *web* menjadi solusi untuk meningkatkan efektivitas layanan hukum dan memberikan *competitive advantage* bagi firma hukum di era digital.

*Framework Laravel* menjadi salah satu teknologi yang dapat digunakan dalam pengembangan sistem berbasis *web*. *Laravel* menawarkan berbagai fitur yang mendukung keamanan, skalabilitas, dan kemudahan dalam pengelolaan data (Haryana, 2020). Penelitian sebelumnya menunjukkan bahwa penggunaan *framework Laravel* dalam pengembangan sistem informasi hukum dapat meningkatkan efektivitas pengelolaan data dan memberikan

antarmuka yang *user-friendly* (Dewi et al., 2020; Pasaribu et al., 2023). Dengan kemampuannya dalam mengelola *routing*, autentikasi pengguna, serta integrasi dengan berbagai sistem, *Laravel* menjadi pilihan yang tepat untuk membangun sistem jasa layanan hukum berbasis *web*.

Firma hukum sebagai penyedia layanan hukum dapat memanfaatkan *platform* berbasis *web* untuk menyediakan berbagai layanan seperti konsultasi hukum *online*, manajemen dokumen hukum, serta pemantauan kasus klien secara *real-time*. Studi yang dilakukan oleh Laaziri et al. (2019) menunjukkan bahwa *Laravel* memiliki keunggulan dalam pengembangan aplikasi *web* skala besar dengan dokumentasi yang komprehensif dan dukungan komunitas yang luas. Selain itu, aspek keamanan yang disediakan *Laravel*, seperti enkripsi data menggunakan algoritma *Bcrypt* dan proteksi *CSRF*, sangat relevan untuk layanan hukum yang membutuhkan tingkat keamanan tinggi (Kausar Bagwan & Swati Ghule, 2019). Dengan demikian, sistem berbasis *Laravel* ini diharapkan dapat meningkatkan aksesibilitas serta efektivitas dalam pengelolaan layanan hukum.

Berdasarkan permasalahan tersebut, maka diusulkan "Analisis dan Perancangan Jasa Layanan Firma Hukum Berbasis *Web* Menggunakan *Framework Laravel*", sehingga dapat membantu meningkatkan efektivitas serta kualitas layanan firma hukum dalam memberikan pelayanan terbaik kepada masyarakat. Penelitian ini diharapkan dapat memberikan kontribusi pada proses digitalisasi sektor hukum dan menjadi referensi bagi firma hukum lain yang ingin mengadopsi teknologi serupa untuk meningkatkan daya saing di era digital.

## 1.2 Identifikasi Masalah

Dalam pengembangan *website* jasa layanan firma hukum berbasis *web* menggunakan *framework Laravel*, terdapat beberapa masalah yang perlu diidentifikasi dan diselesaikan agar *website* dapat berfungsi secara optimal dan memberikan manfaat maksimal bagi firma hukum serta para kliennya:

1. Kurangnya digitalisasi pada firma hukum, Banyak firma hukum yang masih mengandalkan metode tradisional dalam memberikan layanan kepada klien, seperti pertemuan tatap muka dan manajemen dokumen fisik, yang menyebabkan keterbatasan dalam aksesibilitas dan Efektivitas pelayanan.
2. Kesulitan klien dalam mengakses layanan hukum secara *online*, Klien sering mengalami kendala dalam mencari informasi dan menggunakan jasa hukum secara mudah dan cepat karena kurangnya *platform* berbasis *web* yang terintegrasi dengan layanan hukum.
3. Belum optimalnya penggunaan teknologi oleh firma hukum, Meskipun teknologi informasi berkembang pesat, belum semua firma hukum memanfaatkan sistem berbasis *web* yang canggih untuk mendukung operasional mereka, yang dapat mengurangi daya saing di era digital.
4. Kurangnya *platform* yang dapat mengelola aspek operasional firma hukum, Firma hukum membutuhkan sebuah sistem yang tidak hanya memfasilitasi interaksi dengan klien tetapi juga mampu mengelola dokumen, komunikasi, dan konsultasi secara *online* dengan efektif dan aman.
5. Kebutuhan akan pengembangan sistem yang andal, aman, dan Efektif, Diperlukan suatu *framework* yang dapat memberikan solusi pengembangan *website* jasa layanan hukum yang responsif, aman, dan mudah diimplementasikan, sehingga dapat mendukung pengelolaan layanan hukum secara optimal.

### 1.3 Ruang Lingkup Penelitian

Penelitian ini berfokus pada situs jasa layanan firma hukum berbasis *web* menggunakan *framework Laravel*. Adapun ruang lingkup penelitian ini meliputi:

1. Pengembangan sistem berbasis *web*, Penelitian ini akan merancang dan mengembangkan *website* yang dirancang khusus untuk mendukung layanan hukum secara *online*. Pengembangan sistem ini akan difokuskan pada penggunaan *framework Laravel*.
2. Analisis kebutuhan pengguna, Penelitian akan melakukan analisis terkait kebutuhan firma hukum dan klien dalam mengakses dan menyediakan layanan hukum secara digital, mencakup fitur-fitur utama yang diperlukan seperti konsultasi *online*, manajemen dokumen, dan komunikasi langsung dengan klien.
3. Perancangan *user interface (UI)* dan *user experience (UX)*, Penelitian ini mencakup perancangan antarmuka pengguna yang mudah digunakan dan memberikan pengalaman pengguna yang nyaman, dengan fokus pada aksesibilitas dan kemudahan navigasi.
4. Keamanan dan skalabilitas *system*, Penelitian akan mempertimbangkan aspek keamanan data, mengingat layanan hukum memerlukan perlindungan informasi yang tinggi. Selain itu, *website* akan dirancang agar dapat menangani beban pengguna yang besar dan dapat berkembang sesuai dengan kebutuhan firma hukum.

#### 4.1 Aspek Keamanan (*Security*):

##### A. Enkripsi Data *End-to-End*:

- I. Implementasi *SSL/TLS Certificate* untuk enkripsi data *in-transit*
- II. Enkripsi *database* menggunakan *AES-256* untuk data *at-rest*

III. *Hashing password* menggunakan algoritma *Bcrypt* dengan *salt random*

B. Autentikasi dan Otorisasi Berlapis:

I. *Multi-factor Authentication (MFA)* untuk akun *admin*

II. *Role-based Access Control (RBAC)* untuk pembatasan akses fitur

III. *Session management* dengan *timeout* otomatis untuk keamanan

C. Proteksi Aplikasi:

I. *Cross-Site Request Forgery (CSRF) protection*

II. *SQL Injection prevention* melalui *prepared statements*

III. *Cross-Site Scripting (XSS) filtering* pada semua *input user*

IV. *Rate limiting* untuk mencegah *brute force attacks*

D. Compliance dan Audit:

I. Implementasi *audit trail* untuk *tracking* semua aktivitas *user*

II. *Compliance* dengan regulasi *Perlindungan Data Pribadi (PDP)*  
Indonesia

III. *Regular security scanning* dan *vulnerability assessment*

## 1.4 Tujuan Dan Manfaat Penelitian

### 1.4.1 Tujuan Penelitian

1. Menganalisis kebutuhan pengguna, mengidentifikasi dan menganalisis kebutuhan firma hukum serta klien dalam penggunaan layanan hukum berbasis *web* untuk meningkatkan efektivitas pelayanan.
2. Merancang *website* jasa layanan firma hukum, mengembangkan desain dan fitur *website* yang dapat memenuhi kebutuhan pengguna serta meningkatkan pengalaman pengguna dalam mengakses layanan hukum secara *online*.
3. Mengimplementasikan *framework Laravel*, menggunakan *framework Laravel* dalam pengembangan *website* untuk memastikan keamanan, Efektivitas, dan kemudahan dalam pemeliharaan sistem.
4. Mengujicoba dan mengevaluasi *system*, melakukan pengujian terhadap *website* yang dikembangkan untuk memastikan fungsionalitas dan kinerja sistem sesuai dengan harapan serta kebutuhan pengguna.

### 1.4.2 Manfaat Penelitian

1. Meningkatkan kesiapan dan aksesibilitas layanan hukum, *Website* yang dirancang memungkinkan klien dan firma hukum mengakses informasi, konsultasi, dan layanan hukum dengan cepat, sehingga meminimalkan potensi permasalahan akibat keterlambatan informasi atau komunikasi.
2. Menyediakan sistem layanan hukum yang terintegrasi, Hasil perancangan berupa *website* dengan fitur lengkap (katalog layanan, manajemen kasus, konsultasi *online*, sistem pembayaran terintegrasi) yang dapat menjadi solusi proaktif dalam mendukung digitalisasi proses bisnis firma hukum.

3. Menjamin keamanan dan keberlanjutan sistem, Implementasi *framework Laravel* menghasilkan sistem yang memiliki autentikasi berlapis, enkripsi data, dan arsitektur MVC, sehingga tidak hanya aman tetapi juga mudah dikembangkan di masa depan untuk mencegah gangguan operasional.
4. Memvalidasi kesesuaian sistem dengan kebutuhan, Pengujian Black Box dan User Acceptance Testing (UAT) memastikan sistem memenuhi kebutuhan fungsional dan non-fungsional, serta memberikan masukan untuk penyempurnaan sebelum digunakan secara luas.

## **1.5 Sistematika Penulisan**

### **BAB I PENDAHULUAN**

Pada bab ini akan membahas tentang Latar belakang Masalah, Identifikasi Masalah, Ruang Lingkup Penelitian, Tujuan dan Manfaat Penelitian dan Sistematika Penulisan

### **BAB II LANDASAN TEORI**

Pada bab ini akan membahas Teori yang digunakan untuk penelitian (diagram-diagram, simbol-simbol, rumus-rumus, gambar-gambar, dll), Teori Metode yang digunakan, Penelitian Yang Relevan.

### **BAB III METODOLOGI PENELITIAN**

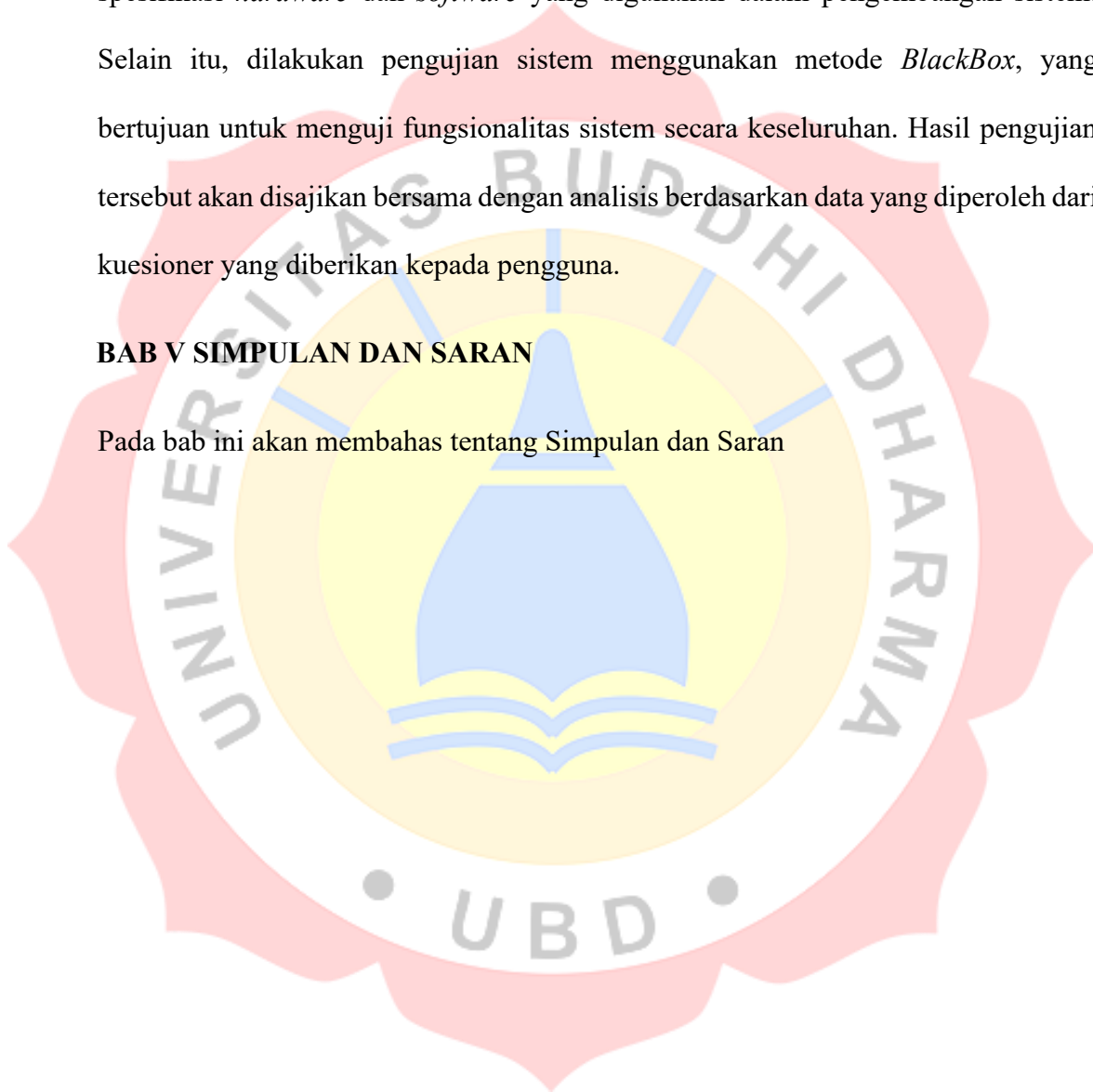
Pada bab ini akan membahas tentang Gambaran Umum Perusahaan, Teknik Pengumpulan Data Penelitian, Prosedur Sistem Berjalan, Dokumentasi *Input* dan *Output*, Analisis Masalah, identifikasi Kebutuhan Sistem, Metode, *Requirement Elicitation*, Jadwal Penelitian dan Kerangka Pemikiran.

## **BAB IV HASIL DAN PEMBAHASAN**

Pada bab ini, akan dijelaskan mengenai implementasi sistem yang telah dilakukan serta pembahasan hasil pengujian sistem yang dilakukan. Penjelasan meliputi tampilan sistem yang dapat dilihat pada screenshot yang disertakan, serta spesifikasi *hardware* dan *software* yang digunakan dalam pengembangan sistem. Selain itu, dilakukan pengujian sistem menggunakan metode *BlackBox*, yang bertujuan untuk menguji fungsionalitas sistem secara keseluruhan. Hasil pengujian tersebut akan disajikan bersama dengan analisis berdasarkan data yang diperoleh dari kuesioner yang diberikan kepada pengguna.

## **BAB V SIMPULAN DAN SARAN**

Pada bab ini akan membahas tentang Simpulan dan Saran



## **BAB II**

### **LANDASAN TEORI**

#### **2.1 Analisis**

Analisis adalah proses memecah informasi atau sistem yang kompleks menjadi komponen yang lebih kecil dan lebih mudah dimengerti untuk mendapatkan wawasan dan membuat keputusan yang tepat. Ini melibatkan pemeriksaan hubungan dan interaksi antara komponen-komponen ini untuk memahami keseluruhan sistem dengan lebih baik (Hendrawan et al., 2020).

Tujuan utama analisis adalah untuk memahami struktur atau fungsi yang mendasari suatu sistem atau masalah. Pemahaman ini dapat membantu dalam mengidentifikasi pola, penyebab, dan efek, yang sangat penting untuk pengambilan keputusan dan pemecahan masalah.

#### **2.2 Perancangan**

Merancang adalah proses pembuatan rencana atau cetak biru untuk konstruksi atau pengembangan suatu objek, sistem, atau struktur. Ini melibatkan konseptualisasi dan perencanaan elemen dan tata letak untuk mencapai tujuan atau fungsi tertentu.

Tujuan utama perancangan adalah untuk menerjemahkan ide dan persyaratan ke dalam bentuk nyata yang dapat diimplementasikan. Ini bertujuan untuk memenuhi kebutuhan pengguna dan memecahkan masalah spesifik melalui solusi kreatif dan fungsional.

Merancang sering melibatkan penggunaan diagram dan model untuk memetakan kebutuhan pengguna dan fungsi sistem. Alat umum termasuk Diagram Kasus Penggunaan, Diagram Aktivitas, dan Diagram Kelas, yang membantu memvisualisasikan desain dan memastikannya selaras dengan kebutuhan pengguna.

Perancangan diterapkan di berbagai bidang, termasuk teknologi, arsitektur, dan teknik, untuk menciptakan sistem, layanan, dan struktur yang Efektif, ramah pengguna, dan menyenangkan secara estetika. Dalam teknologi, perancangan sangat penting untuk mengembangkan antarmuka pengguna dan arsitektur sistem (Hendrawan et al., 2020).

### **2.3 Sistem Informasi**

Sistem informasi dapat didefinisikan sebagai seperangkat komponen terkoordinasi yang mengumpulkan, memproses, menyimpan, dan menyebarkan informasi untuk mendukung pengambilan keputusan, koordinasi, kontrol, analisis, dan visualisasi dalam suatu organisasi. Sistem ini mengintegrasikan perangkat keras, perangkat lunak, data, prosedur, dan orang-orang untuk mengelola dan memproses informasi secara efektif (Santoso, n.d. 2020).

Komponen utama sistem informasi meliputi perangkat keras (perangkat fisik), perangkat lunak (program dan aplikasi), data (informasi dan *database*), prosedur (kebijakan dan proses), dan orang-orang (pengguna dan profesional TI) (Santoso, n.d. 2020). Komponen-komponen ini bekerja sama untuk melakukan aktivitas *input*, pemrosesan, *output*, penyimpanan, dan kontrol.

### **2.4 Informasi**

Informasi adalah data yang telah diolah sehingga memiliki makna dan dapat digunakan untuk pengambilan keputusan (Agustine Bacsafr & Mustika Kusumawardani, 2022). Dalam konteks sistem informasi, informasi yang dihasilkan berasal dari pengolahan data mentah seperti data klien, kasus, dokumen hukum, atau status layanan. Informasi ini memiliki nilai bagi pengguna karena memberikan pemahaman yang lebih dalam mengenai kondisi atau situasi tertentu.

Dalam layanan firma hukum berbasis *web*, informasi menjadi aspek krusial karena klien dan pengacara membutuhkan informasi yang akurat, tepat waktu, dan dapat diakses dengan mudah. Misalnya, informasi terkait jadwal persidangan, status perkembangan kasus, atau dokumen yang perlu ditandatangani oleh klien dapat disampaikan melalui sistem *web*. Informasi yang akurat memungkinkan pengambilan keputusan yang lebih baik, terutama dalam konteks yang sensitif seperti hukum (Agustine Bacsafra & Mustika Kusumawardani, 2022).

Lebih jauh lagi, penyajian informasi dalam sistem *web* harus didukung oleh antarmuka yang mudah dipahami dan sistem navigasi yang baik. Informasi yang disajikan secara tepat, lengkap, dan aman akan meningkatkan Efektivitas dan kepercayaan klien terhadap firma hukum.

## **2.5 Data**

Data mengacu pada fakta dan angka mentah yang belum diproses yang dikumpulkan dari berbagai sumber. Ini bisa berupa angka, teks, gambar, atau bentuk informasi lainnya yang dapat direkam dan dianalisis (Al Qardh et al., n.d. 2021).

Data dapat dikategorikan ke dalam berbagai jenis, seperti kualitatif dan kuantitatif. Data kualitatif bersifat deskriptif dan konseptual, sedangkan data kuantitatif bersifat numerik dan dapat diukur (Al Qardh et al., n.d. 2021).

## **2.6 Firma Hukum**

Firma hukum adalah entitas bisnis yang dibentuk oleh satu atau lebih pengacara untuk terlibat dalam praktik hukum. Layanan utama yang ditawarkan oleh firma hukum adalah untuk memberi tahu klien tentang hak dan tanggung jawab hukum mereka dan untuk mewakili mereka dalam kasus perdata atau pidana, transaksi bisnis, dan hal-hal lain di mana nasihat hukum dan bantuan lainnya dicari.

Tujuan utama firma hukum adalah untuk memberikan layanan hukum kepada individu, bisnis, dan entitas pemerintah. Layanan ini termasuk menawarkan nasihat hukum, menyusun dokumen hukum, dan mewakili klien dalam proses hukum.

Firma hukum dapat bervariasi dalam ukuran dari satu pengacara hingga perusahaan besar dengan ratusan pengacara. Mereka dapat diatur sebagai kemitraan, perusahaan, atau perseroan terbatas, tergantung pada yurisdiksi dan preferensi perusahaan.

Firma hukum menawarkan berbagai layanan, termasuk litigasi, pekerjaan transaksional, dan layanan konsultasi. Mereka mungkin mengkhususkan diri dalam bidang hukum tertentu seperti hukum perusahaan, hukum keluarga, kekayaan intelektual, atau hukum pidana (Natacia & Mailoa, n.d. 2020).

## **2.7 Jasa**

Jasa mengacu pada kualitas menjadi sangat baik atau layak, terutama agar pantas mendapatkan pujian atau penghargaan. Ini sering digunakan untuk mengevaluasi nilai atau kelayakan individu, ide, atau sistem berdasarkan kualitas dan efektivitas yang melekat.

Tujuan menilai prestasi adalah untuk mengakui dan menghargai keunggulan, memastikan bahwa individu atau sistem yang menunjukkan kualitas dan efektivitas tinggi diakui dan diberi insentif. Hal ini dapat mendorong motivasi dan peningkatan di berbagai bidang.

Jasa diterapkan di berbagai bidang, termasuk pendidikan, pekerjaan, dan teknologi, untuk memastikan bahwa keputusan dibuat berdasarkan kualitas dan efektivitas. Misalnya, dalam teknologi, prestasi dapat memandu pemilihan kerangka kerja atau sistem yang paling memenuhi kebutuhan pengguna dan tujuan proyek.

Hasil evaluasi prestasi adalah identifikasi individu atau sistem berkualitas tinggi yang dapat berfungsi sebagai tolok ukur atau model bagi orang lain. Hal ini dapat mengarah pada peningkatan standar dan praktik di berbagai domain (Natacia & Mailoa, n.d. 2020).

## **2.8 Layanan**

Layanan adalah penawaran tidak berwujud yang melibatkan perbuatan, kinerja, atau upaya yang tidak dapat dimiliki secara fisik. Ini dirancang untuk memberikan nilai kepada konsumen dengan memenuhi kebutuhan atau persyaratan tertentu.

Tujuan utama dari suatu layanan adalah untuk memberikan nilai kepada konsumen dengan menangani kebutuhan spesifik atau memecahkan masalah tertentu. Layanan sering disesuaikan untuk memenuhi kebutuhan unik setiap konsumen, memastikan kepuasan dan efektivitas (Natacia & Mailoa, n.d. 2020).

Layanan dicirikan oleh sifat tidak berwujud, artinya tidak dapat disentuh atau dimiliki seperti layanan fisik. Mereka juga mudah rusak, karena tidak dapat disimpan untuk digunakan nanti, dan seringkali memerlukan interaksi langsung antara penyedia dan konsumen.

Layanan diterapkan di berbagai sektor, termasuk perawatan kesehatan, pendidikan, teknologi, dan sistem hukum. Misalnya, dalam konteks sistem hukum, layanan mungkin melibatkan penyediaan akses ke informasi hukum dan dokumentasi kepada publik

## **2.9 Internet**

Internet adalah jaringan global yang luas dari jaringan komputer yang saling berhubungan yang berkomunikasi menggunakan protokol standar. Ini memungkinkan pertukaran data dan informasi di seluruh dunia, menghubungkan jutaan jaringan swasta, publik, akademik, bisnis, dan pemerintah (Azriel et al., n.d. 2020).

Ini berfungsi sebagai infrastruktur komunikasi yang mendukung berbagai layanan seperti email, transfer file, dan World Wide *Web* (WWW). Infrastruktur ini memungkinkan komunikasi yang mulus dan berbagi data antara pengguna dan perangkat secara global.

Internet beroperasi pada satu set protokol standar, terutama Internet Protocol (IP) dan Transmission *Control* Protocol (TCP) (Azriel et al., n.d. 2020). Protokol ini memastikan bahwa data ditransmisikan secara akurat dan Efektif antara jaringan dan perangkat yang berbeda.

### **2.10 Website**

Situs *web* adalah kumpulan halaman *web* terkait, biasanya diidentifikasi oleh nama domain umum, dan diterbitkan pada setidaknya satu server *web*. Situs *web* dapat diakses melalui Internet menggunakan browser *web* (Cakti & Gunawan, n.d. 2022).

Situs *web* terdiri dari beberapa elemen seperti teks, gambar, video, dan konten multimedia lainnya. Elemen-elemen ini diatur ke dalam halaman *web* yang dihubungkan bersama melalui hyperlink.

Situs *web* melayani berbagai tujuan, termasuk menyediakan informasi, memfasilitasi komunikasi, menawarkan layanan, dan mengaktifkan *e-commerce* (Cakti & Gunawan, n.d. 2022). Mereka bisa bersifat pribadi, komersial, pemerintah, atau nirlaba.

### **2.11 Online**

*Online* mengacu pada keadaan terhubung ke Internet atau jaringan, memungkinkan pertukaran data dan komunikasi secara *real-time*. Konektivitas ini memungkinkan akses ke berbagai sumber daya dan layanan digital (Satria & Susanti, 2024).

Menjadi *online* memungkinkan pengguna untuk mengakses beragam informasi dan sumber daya, termasuk situs *web*, *database*, dan perpustakaan digital. Konektivitas ini sangat penting untuk penelitian, pendidikan, dan tetap mendapat informasi.

Banyak aplikasi dirancang untuk digunakan secara *online*, memanfaatkan teknologi *web* untuk menyediakan layanan seperti manajemen data, pembuatan konten, dan alat kolaborasi. Contohnya termasuk klien email berbasis *web*, layanan penyimpanan cloud, dan suite kantor *online* (Satria & Susanti, 2024).

### **2.12 Database**

Basis data merupakan komponen yang sangat penting dan memerlukan perhatian yang serius. Fungsinya sebagai wadah untuk menyimpan data atau informasi yang akan digunakan di masa depan sangatlah penting. Menurut definisi, "Sistem Basis Data adalah sistem komputer yang dirancang untuk menyimpan data yang telah diproses atau informasi, serta memastikan bahwa informasi tersebut dapat diakses sesuai kebutuhan." Dari pengertian ini, dapat disimpulkan bahwa basis data adalah koleksi tabel yang berisi data dan fakta, berperan sebagai sumber informasi yang tersimpan dalam bentuk digital dan dapat diakses menggunakan perangkat lunak komputer. Hal ini sangat membantu dalam memudahkan aktivitas untuk mendapatkan informasi yang lebih terperinci.

### **2.13 Framework**

Kerangka kerja adalah seperangkat alat, perpustakaan, konvensi, dan praktik terbaik yang memberikan dasar untuk mengembangkan aplikasi perangkat lunak. Ini dirancang untuk menyederhanakan proses pengembangan dengan menyediakan komponen pra-bangun dan pendekatan terstruktur untuk pengkodean.

Tujuan utama dari kerangka kerja adalah untuk merampingkan proses pengembangan dengan menawarkan kode yang dapat digunakan kembali dan cara standar untuk

membangun aplikasi. Ini membantu pengembang fokus pada aspek unik dari proyek mereka daripada menciptakan kembali fungsionalitas umum

Kerangka kerja dicirikan oleh kemampuannya untuk menegakkan arsitektur atau pola desain tertentu, yang membantu menjaga konsistensi dan kualitas di berbagai proyek. Mereka sering menyertakan pustaka untuk menangani tugas-tugas umum seperti akses *database*, otentikasi pengguna, dan validasi *input* (Hendrawan et al., 2020).

#### **2.14 Laravel**

*Laravel* adalah kerangka aplikasi *web* dengan sintaks ekspresif dan elegan. Ini dirancang untuk membuat proses pengembangan lebih mudah dan lebih Efektif dengan menyediakan seperangkat alat dan sumber daya untuk membangun aplikasi *web* modern.

Tujuan utama *Laravel* adalah untuk menyederhanakan pengembangan aplikasi *web* dengan menawarkan sintaks yang bersih dan lugas, yang membantu pengembang fokus pada fungsionalitas inti aplikasi mereka daripada berurusan dengan tugas berulang.

*Laravel* mencakup fitur-fitur seperti sistem *routing* yang kuat, ORM yang kuat (Object-Relational Mapping) yang disebut Eloquent, dan mesin templating yang disebut Blade. Fitur-fitur ini membantu merampingkan proses pengembangan dan meningkatkan kualitas aplikasi secara keseluruhan.

Menggunakan *Laravel* dapat menyebabkan peningkatan produktivitas dan pemeliharaan aplikasi *web*. Ini mempromosikan praktik terbaik dan menyediakan pendekatan terstruktur untuk pengkodean, yang membantu pengembang membangun aplikasi yang dapat diskalakan dan aman secara Efektif (Hendrawan et al., 2020).

## 2.15 XAMPP

XAMPP adalah paket tumpukan solusi server *web* lintas *platform* gratis dan *open-source* yang dikembangkan oleh Apache Friends (Al-Zikri et al., 2019). Ini terdiri dari Apache HTTP Server, MariaDB (sebelumnya MySQL), dan penafsir untuk skrip yang ditulis dalam PHP dan Perl . Berikut adalah Komponen XAMPP :

1. Apache, Apache HTTP Server adalah server *web* yang kuat, fleksibel, dan sesuai dengan HTTP/1.1. Ini adalah komponen inti dari XAMPP yang menangani permintaan HTTP dan melayani halaman *web*.
2. MariaDB/MySQL, XAMPP termasuk MariaDB, sistem manajemen basis data relasional *open-source* yang populer, yang merupakan pengganti *drop-in* untuk MySQL.
3. PHP, PHP adalah bahasa scripting *open-source* yang banyak digunakan yang sangat cocok untuk pengembangan *web* dan dapat disematkan ke dalam HTML.
4. Perl, Perl adalah bahasa pemrograman yang sangat mampu dan kaya fitur dengan lebih dari 30 tahun pengembangan.

## 2.16 Metodologi Agile

Metode *Agile* adalah pendekatan dinamis untuk pengembangan perangkat lunak yang menekankan fleksibilitas dan kemampuan beradaptasi. Ini memungkinkan perubahan pada setiap tahap proses pengembangan, membuatnya sangat cocok untuk lingkungan di mana persyaratan diharapkan berkembang.

Metode *Agile* menawarkan beberapa keuntungan, termasuk peningkatan fleksibilitas, waktu pengiriman yang lebih cepat, dan peningkatan kepuasan pelanggan. Namun, mereka

juga datang dengan tantangan, seperti potensi kompromi pada kualitas karena siklus pengembangan yang cepat (Nova et al., 2022)

Penelitian melakukan Tinjauan Literatur Sistematis (SLR) untuk menganalisis penggunaan metode *Agile* pada tahun 2021. Tinjauan ini mengidentifikasi bahwa metode *Agile*, terutama *scrum*, banyak digunakan di berbagai sektor, termasuk bisnis, pendidikan, dan pemerintah, untuk mengembangkan sistem informasi berbasis *web*.

Dalam implementasi metodologi *Agile*, terdapat 5 tahapan utama yang membentuk siklus pengembangan iteratif:

1. Perencanaan Sprint (*Sprint Planning*) Tahap perencanaan yang menentukan fitur dan fungsi yang akan dikembangkan dalam setiap iterasi. Tim mengidentifikasi *user stories*, memprioritaskan *backlog*, dan menentukan *sprint goal* yang spesifik dan terukur.
2. Pengembangan dan Implementasi Sprint (*Sprint Development*) Fase aktif pengembangan dimana tim melakukan coding, desain, dan implementasi fitur sesuai dengan rencana sprint. Tahap ini melibatkan *daily standup meetings* untuk memantau progress dan mengatasi hambatan.
3. Pengujian dan Umpan Balik (*Testing and Feedback*) Proses pengujian berkelanjutan yang melibatkan *unit testing*, *integration testing*, dan pengumpulan *feedback* dari *stakeholder*. Tahap ini memastikan kualitas *deliverable* sebelum masuk ke tahap *review*.
4. Evaluasi Sprint (*Sprint Review*) Sesi demonstrasi dan evaluasi hasil sprint kepada *stakeholder*. Tim menunjukkan fitur yang telah diselesaikan, mengumpulkan *feedback*, dan menilai pencapaian *sprint goal* untuk menentukan langkah selanjutnya.

5. Retrospektif Sprint (*Sprint Retrospective*) Refleksi internal tim untuk mengidentifikasi apa yang berjalan baik, apa yang perlu diperbaiki, dan tindakan perbaikan untuk sprint berikutnya. Tahap ini fokus pada *continuous improvement* proses pengembangan.

Kelima tahapan ini membentuk siklus iteratif yang berkelanjutan, memungkinkan tim untuk beradaptasi dengan perubahan kebutuhan dan meningkatkan kualitas produk secara bertahap melalui *feedback loop* yang konsisten.

### **2.17 Use case Diagram**

*Use case* atau diagram *use case* merupakan metode umum dalam pengembangan sistem informasi yang bertujuan untuk menggambarkan perilaku sistem yang akan dibuat. Konsep "perilaku" atau "kelakuan" dalam konteks ini mengacu pada aktivitas atau fungsi sistem yang melibatkan interaksi antara pengguna atau aktor dengan sistem itu sendiri.

Pada dasarnya, *Use case* adalah sebuah alat untuk secara terstruktur menjelaskan bagaimana interaksi antara satu atau lebih aktor dengan sistem informasi yang sedang dikembangkan atau akan dibangun. Aktor dalam konteks *use case* dapat berupa individu, sistem eksternal, atau entitas lain yang terlibat langsung dalam proses.




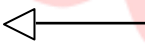
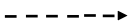
Dalam penggunaan *Use Case*, kita mengidentifikasi berbagai aksi atau kegiatan yang akan dilakukan oleh aktor terhadap sistem. *Use case* memberikan cara untuk menggambarkan secara visual interaksi antara aktor dan sistem, serta menyajikan skenario-skenario yang menjelaskan bagaimana sistem akan merespon aksi-aksi tersebut.


Diagram *use case* digunakan untuk menggambarkan secara visual hubungan antara aktor dan *use case*, serta interaksi di antara keduanya. Umumnya, aktor direpresentasikan

sebagai kotak persegi panjang, sementara *use case* direpresentasikan sebagai oval. Garis penghubung menunjukkan hubungan antara aktor dan *use case*.

Dengan menggunakan *use case* untuk memodelkan interaksi antara aktor dan sistem, pengembang dapat memahami lebih baik kebutuhan dan harapan pengguna terhadap sistem yang akan dibangun. Hal ini sangat membantu dalam tahap perencanaan, desain, dan implementasi sistem informasi agar sesuai dengan kebutuhan pengguna akhir secara lebih efektif.

**Tabel 2. 1 Simbol - Simbol *Use Case Diagram***

No	Gambar	Deskripsi
1.	<p>Actor</p> 	Menyebutkan secara rinci peran - peran yang dilakukan oleh pengguna saat sedang berinteraksi dengan kasus penggunaan.
2.	<p><i>Use Case</i></p> 	Hubungan dan interaksi antara sistem dan pihak yang terlibat.
3.	<p><i>Associantion</i></p> 	Representasi dari koneksi antara pihak yang terlibat dan kasus penggunaan.
4.	<p><i>Generalization</i></p> 	Menyatakan peran khusus dari pihak yang terlibat dalam berinteraksi dengan kasus penggunaan.
5.	<p><i>Include</i></p> 	Menunjukkan bahwa suatu kasus penggunaan sepenuhnya merupakan fungsi dari kasus penggunaan lain.

6.	<p><i>Extend</i></p> 	<p>Menunjukkan bahwa suatu kasus penggunaan dapat menambahkan fungsionalitas dari kasus penggunaan lain jika kondisi tertentu terpenuhi</p>

Sumber: Yakub et al., 2025

## 2.18 Activity Diagram

*Activity Diagram* adalah semacam bagan UML yang digunakan untuk menggambarkan proses kerja suatu kerangka kerja atau proses bisnis. Diagram ini menggambarkan aktivitas-aktivitas yang dilakukan dalam suatu sistem atau proses secara berurutan dan ketergantungannya. Penting untuk diingat bahwa Diagram Aktivitas fokus terutama pada aktivitas sistem itu sendiri karena diagram tersebut tidak mendeskripsikan aktivitas aktor.



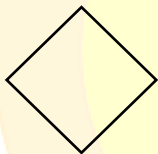


Salah satu jenis diagram UML yang sering digunakan adalah *Activity Diagram*. Diagram ini bertujuan untuk menggambarkan aliran kerja atau aktivitas dalam sebuah sistem atau proses bisnis. Aktivitas-aktivitas ini direpresentasikan dalam bentuk simbol-simbol seperti *elips* untuk awal dan akhir aliran kerja, persegi panjang untuk aktivitas, dan panah untuk menggambarkan arah aliran kerja antar aktivitas.

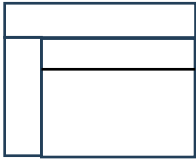
Penting untuk dicatat bahwa *Activity Diagram* fokus pada aktivitas yang dilakukan oleh sistem itu sendiri, bukan oleh aktor yang terlibat dalam proses tersebut. Ini memungkinkan para pengembang untuk memahami dengan jelas urutan aktivitas dalam sistem dan mengidentifikasi area yang mungkin memerlukan perbaikan atau peningkatan dalam aliran kerja.

Dengan menggunakan UML dan diagram seperti *Activity Diagram*, para pengembang dapat merancang sistem perangkat lunak secara lebih Efektif, memahami kebutuhan

pengguna dengan lebih baik, dan mengelola kompleksitas sistem dengan lebih baik. Ini membantu meningkatkan kualitas dan keberhasilan proyek pengembangan perangkat lunak secara keseluruhan.

**Tabel 2. 2 Simbol - Simbol *Activity Diagram***

No	Simbol	Deskripsi
1.	Status awal 	Pada awal aktivitas dalam sistem, sebuah diagram aktivitas menunjukkan titik awal dari proses tersebut.
2.	Aktivitas 	Sistem melakukan aktivitas dengan biasanya memulai dengan kata kerja.
3.	Percabangan 	Percabangan asosiatif terjadi ketika ada lebih dari satu pilihan aktivitas yang bisa dipilih.
4.	Penggabungan / Join 	Penggabungan asosiatif terjadi ketika lebih dari satu aktivitas disatukan menjadi satu.
5.	Status Akhir 	Di akhir aktivitas sistem, sebuah diagram aktivitas menunjukkan titik akhir dari proses tersebut

6.	Swimlane 	Memilah-milah organisasi bisnis yang bertanggung jawab atas aktivitas yang terjadi.
----	---	---

Sumber: Yakub et al., 2025

## 2.19 Analisis Sistem





Analisis sistem adalah fase awal pengembangan sistem, berfungsi sebagai dasar untuk menentukan keberhasilan sistem informasi yang dihasilkan (Muhidin et al., 2017). Ini melibatkan pemecahan sistem informasi lengkap menjadi bagian-bagian komponennya untuk mengidentifikasi dan mengevaluasi masalah yang diharapkan, memungkinkan perbaikan yang diusulkan.


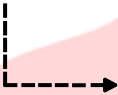
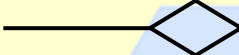
Analisis sistem secara sistematis menilai bagaimana sistem berfungsi dengan mengamati proses *input* dan data *output* untuk meningkatkan proses organisasi. Ini memainkan tiga peran penting: sebagai konsultan, ahli dukungan, dan ahli perubahan (Muhidin et al., 2017).

## 2.20 Class Diagram (Kelas Diagram)

Diagram kelas, atau *class diagram*, mengilustrasikan struktur sistem dengan cara mendefinisikan kelas-kelas yang diperlukan untuk membangun sistem, seperti yang dijelaskan oleh Rosa dan Shalahuddin (2015:141). Untuk menggambarkan arsitektur perangkat lunak dari sistem yang akan dibuat, berikut adalah simbol-simbol yang digunakan dalam *class diagram*:

**Tabel 2. 3 Simbol - Simbol *Class Diagram***

No	Simbol	Deskripsi
1.		Status dalam struktur sistem.
2.	<p>Antarmuka / <i>Interface</i></p> 	Serupa dengan konsep interface dalam pemrograman berbasis objek.
3.	<p><i>Associantion</i></p> 	Percabangan asosiatif terjadi ketika ada lebih dari satu pilihan aktivitas.
4.	<p>Asosiasi berarah / <i>directed association</i></p> 	Hubungan antara kelas-kelas yang menjelaskan bagaimana satu kelas digunakan oleh kelas lain, sering kali disertai dengan multiplicity.

5.	<p>Generalisasi</p> 	<p>Hubungan antara kelas-kelas yang menunjukkan hubungan umum dan spesifik antara kelas.</p>
6.	 <p>Ketergantungan/<i>Depedency</i></p>	<p>Hubungan antara kelas-kelas yang menunjukkan ketergantungan satu sama lain.</p>
7.	<p>Agregasi / <i>aggregation</i></p> 	<p>Hubungan antara kelas-kelas yang menunjukkan bahwa satu kelas merupakan bagian dari keseluruhan yang lebih besar.</p>


Sumber: Yakub et al., 2025





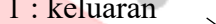


## 2.21 Sequence Diagram (Diagram Sekuen)

Booch (2023) mencatat bahwa diagram sekuen digunakan untuk menggambarkan perilaku objek dalam suatu kasus penggunaan dengan merekam siklus hidup objek serta pesan yang dikirim dan diterima antara objek tersebut. Oleh karena itu, untuk membuat diagram sekuen, penting untuk memahami objek-objek yang terlibat dalam kasus penggunaan dan metode-metode yang dimiliki oleh kelas yang diinstansiasi menjadi objek.

Jumlah diagram sekuen yang diperlukan setidaknya sebanding dengan jumlah kasus penggunaan yang telah didefinisikan, khususnya untuk kasus penggunaan yang memiliki prosesnya sendiri. Semakin banyak interaksi pesan yang tercakup dalam kasus penggunaan, semakin banyak pula diagram sekuen yang harus dibuat. Berikut adalah simbol-simbol yang digunakan dalam diagram sekuen.

**Tabel 2. 4 Simbol - Simbol *Sequence Diagram***

No	Simbol	Deskripsi
1.	 <p>Aktor</p> <div style="border: 1px solid black; padding: 5px; width: fit-content; margin: 10px auto;"> <p>Nama Aktor</p> <hr style="width: 80%; margin: 0 auto;"/> </div> <p>atau</p> <p>Tanpa waktu aktif</p>	<p>Entitas, proses, atau sistem lain yang berinteraksi dengan sistem informasi yang sedang dibuat berada di luar lingkup sistem informasi itu sendiri. Meskipun simbol aktor sering kali digambarkan sebagai manusia, sebenarnya aktor tidak selalu berupa individu manusia dan sering kali direpresentasikan dengan menggunakan kata benda di awal frasa untuk menunjukkan peran aktor.</p>

2.	<p style="text-align: center;">Garis Hidup</p>  <p style="text-align: center;"><i>//lifeline</i></p>	Mengindikasikan siklus hidup suatu objek.
3.	<p style="text-align: center;">Objek</p> <div style="border: 1px solid black; padding: 5px; width: fit-content; margin: auto;"> <p style="text-align: center;">nama objek : nama</p> <p style="text-align: center;">kelas</p> </div>	Menunjukkan objek yang bertukar pesan saat berinteraksi.
4.	<p style="text-align: center;">Waktu aktif</p> 	Menunjukkan objek yang sedang aktif dan berinteraksi; semua yang diletakkan di depan waktu aktif ini merupakan tahapan yang dilakukan di dalamnya.
5.	<p style="text-align: center;">Pesan tipe <i>Create</i></p>  <p style="text-align: center;">Pesan tipe <i>call</i></p>	Menunjukkan bahwa suatu objek membuat objek lain, dengan panah menunjuk ke objek yang dibuat.
6.	<p style="text-align: center;">1 : masukan</p> 	Menunjukkan bahwa suatu objek memanggil operasi atau metode dari objek lain atau dari dirinya sendiri.
7.	<p style="text-align: center;">Pesan tipe <i>return</i></p> <p style="text-align: center;">1 : <u>keluaran</u></p> 	Menunjukkan bahwa suatu objek yang telah menjalankan operasi atau metode menghasilkan keluaran yang diterima oleh objek tertentu, dengan panah menunjuk ke objek penerima.
8.	 	Menunjukkan bahwa suatu objek mengakhiri siklus hidup objek lain, dengan panah menunjuk ke objek yang diakhiri; idealnya,

		jika ada pembuatan objek (create), harus ada penghancuran objek (destroy).
--	--	--

Sumber: Yakub et al., 2025

## 2.21 Tinjauan Jurnal

Penelitian yang dilakukan oleh (Daniel Ivan Fraderic, 2022) di Universitas Buddhi Dharma membahas penerapan *Laravel* sebagai *framework* dalam pengembangan sistem penjualan *e-commerce* berbasis *web*. Dengan menggunakan metodologi Waterfall, peneliti berhasil menciptakan sistem penjualan yang memudahkan perusahaan dalam melakukan transaksi secara *online*, mempermudah pencatatan transaksi, serta mengoptimalkan pengolahan informasi penjualan menggunakan teknologi berbasis *web* yang dapat diakses dengan mudah oleh pengguna. Dalam penelitian serupa, (Goutama & Basri, 2022) juga membahas perancangan sistem informasi penjualan berbasis *web* pada PT Buana Merdeka Jaya, dengan fokus pada penyederhanaan proses penjualan dan pencatatan transaksi melalui aplikasi berbasis *Laravel*. Penelitian ini menunjukkan bahwa *Laravel* dan pengujian Black Box membantu memastikan kelancaran fungsi sistem. (Wijaya et al., 2022) mengembangkan sistem informasi pengaduan masyarakat berbasis *web* menggunakan *Laravel*, yang memungkinkan warga untuk melaporkan keluhan dengan lebih Efektif. *Framework Laravel* digunakan karena fleksibilitasnya dalam mengelola data dan mendukung kecepatan serta akurasi pengiriman informasi.

Penelitian oleh (Dan Perancangan et al., n.d. 2020) membahas penerapan Unified Modelling Language (UML) untuk merancang sistem informasi disposisi surat berbasis *web*. Sistem ini dirancang untuk meningkatkan Efektivitas dalam pengelolaan surat yang sebelumnya dilakukan secara manual dan rentan terhadap human error. Penggunaan sistem berbasis *web* membantu mempercepat proses disposisi, mengurangi kesalahan, dan

mempermudah pemantauan surat. Penerapan konsep ini sangat relevan untuk perancangan sistem layanan hukum berbasis *web*, di mana penggunaan *Laravel* dan metode serupa dapat membantu meningkatkan produktivitas serta keakuratan dalam pengelolaan dokumen dan data hukum.

Penelitian (*Laravel et al.*, n.d 2025.) tentang pemanfaatan *framework Laravel* dalam aplikasi penjualan hijab berbasis *web* menunjukkan bahwa *Laravel* merupakan *framework* yang Efektif untuk pengembangan aplikasi *web*, khususnya dalam hal pengelolaan *database* dan antarmuka pengguna. Penelitian ini menggunakan *Laravel* dan Bootstrap untuk membuat tampilan yang lebih responsif dan mudah diakses, menonjolkan keunggulan *Laravel* dalam struktur file dan koding yang sederhana namun powerful. Untuk aplikasi firma hukum berbasis *web*, *framework* ini menawarkan fleksibilitas dalam pembuatan fitur yang dapat menyesuaikan kebutuhan pengguna dengan fokus pada user experience yang optimal.

Sementara itu, penelitian (Anggara et al., 2024) tentang layanan jasa reparasi elektronik berbasis *web* dengan menggunakan Object-Oriented Analysis and Design (OOAD) memberikan wawasan mengenai pentingnya modularitas dan fleksibilitas dalam pengembangan aplikasi *web*. Dengan pendekatan ini, aplikasi dapat diperluas dan diadaptasi sesuai kebutuhan pengguna, yang relevan untuk pengembangan sistem layanan hukum berbasis *web*, di mana fitur seperti pemesanan layanan atau pemantauan status kasus bisa diimplementasikan dengan lebih baik.

Penelitian terakhir oleh (Pasaribu et al., 2023) di Law Firm Duma & Co mengembangkan aplikasi pelayanan hukum berbasis *web* untuk memudahkan pelanggan mengakses informasi dan layanan hukum secara *online*. Aplikasi ini didesain menggunakan *framework Laravel* dan metode UML untuk memudahkan dokumentasi dan pengelolaan konsultasi hukum. Implementasi sistem ini berhasil mengatasi keterbatasan layanan manual

yang memakan waktu dan mengurangi Efektivitas pelayanan. Studi ini memberikan contoh konkret bagaimana sistem layanan hukum berbasis *Laravel* dapat meningkatkan Efektivitas dan aksesibilitas layanan firma hukum, serta mendukung alur kerja yang lebih efektif.

Penelitian ini membahas penerapan berbagai metode dan *framework* untuk merancang sistem informasi berbasis *web* dalam konteks layanan hukum. (Aziz, 2021) menggunakan metode Waterfall dan *framework* CodeIgniter untuk membangun sistem informasi bagi Kantor Hukum NurhadiSigit, bertujuan untuk mempermudah konsultasi hukum secara *online*, meningkatkan Efektivitas, serta meminimalkan risiko kehilangan data.

Penelitian oleh (Dewi et al., 2020) pada Kantor Konsultan Hukum Samsul-Fatkhul-Serangkai (SFS) menggunakan *framework Laravel* dan metode Rapid Application Development (RAD) untuk membuat sistem manajemen dokumen dan jadwal konsultasi, dengan hasil pengujian yang menunjukkan seluruh fitur berfungsi dengan baik. (Setyawan et al., 2023) juga merancang sistem informasi bimbingan konseling di SMKN 1 Purwakarta menggunakan *framework Laravel*, yang meningkatkan Efektivitas pengelolaan laporan dengan pendekatan Extreme Programming (XP), memberikan landasan kuat bagi pengembangan sistem berbasis *web* di sektor lain, termasuk firma hukum. Selain itu, studi oleh (Laaziri et al., 2019) membandingkan *framework Laravel* dan Symfony, dengan kesimpulan bahwa *Laravel* lebih unggul dalam pengembangan cepat aplikasi skala besar, terutama dalam aspek dokumentasi yang mudah dipahami serta dukungan komunitas yang luas. Terakhir, (Kausar Bagwan & Swati Ghule, 2019) menjelaskan keuntungan menggunakan *Laravel* dalam pengembangan aplikasi *web*, terutama dalam aspek keamanan, seperti penggunaan algoritma hashing Bcrypt dan proteksi CSRF, yang sangat relevan untuk layanan hukum berbasis *web* yang membutuhkan keamanan tingkat tinggi.

Pengembangan sistem layanan hukum berbasis *web* memerlukan pendekatan yang komprehensif dalam pemilihan *framework* dan metodologi pengembangan. Penelitian yang

dilakukan oleh (Saputra et al., 2023) tentang implementasi *framework Laravel* dalam perancangan *website* instansi DCKTRP (Dinas Cipta Karya, Tata Ruang dan Pertanahan) dengan modul PPID menunjukkan bahwa *Laravel* memberikan kemudahan dalam pengembangan sistem informasi instansi pemerintah dengan menghasilkan *website* yang responsif, menarik dan mudah dipahami oleh pengguna. Untuk mendukung perancangan yang terstruktur, (Sembiring & Sinuraya, 2019) dalam penelitiannya tentang sistem informasi berbasis *web* menggunakan metode UML menjelaskan bahwa penggunaan Unified Modeling Language sangat efektif untuk merancang sistem yang kompleks melalui use case diagram, activity diagram, dan sequence diagram yang dapat meningkatkan efektivitas pengelolaan data dan mengurangi human error. Sementara itu, (Bacsafra et al., 2022) memperkuat pendekatan pengembangan dengan menggunakan metode prototype yang memungkinkan pengembangan sistem dalam siklus-siklus kecil dengan perbaikan berkelanjutan berdasarkan *feedback* pengguna, dimana pengujian menggunakan black box testing menunjukkan bahwa pendekatan ini mampu memastikan seluruh fitur aplikasi berfungsi sesuai spesifikasi yang diharapkan.

## BAB III

### METODOLOGI PENELITIAN

#### 3.1 Teknik Pengumpulan Data Penelitian

Dalam penelitian ini, data dikumpulkan melalui dua metode utama, yaitu observasi langsung. Metode ini digunakan untuk memperoleh data yang akurat serta mendukung analisis dan perancangan sistem layanan hukum berbasis *web* menggunakan *framework Laravel*.

##### 3.1.1 Observasi Langsung

Fokus observasi meliputi:

1. Pengelolaan Dokumen Hukum

Observasi dilakukan terhadap aktivitas penyimpanan, pengarsipan, dan pencarian dokumen hukum, termasuk tantangan seperti waktu yang dibutuhkan untuk pencarian dokumen dan risiko kehilangan dokumen.

2. Proses Konsultasi dengan Klien

Proses komunikasi dengan klien, baik secara langsung maupun melalui media digital, diamati untuk mencatat hambatan yang sering muncul, seperti keterbatasan waktu atau media komunikasi yang kurang memadai.

3. Pengelolaan Data Klien

Aktivitas pengelolaan data klien, termasuk pengumpulan, penyimpanan, dan akses data, diamati untuk mengidentifikasi aspek yang dapat ditingkatkan dalam sistem baru.

## 3.2 Metodologi Penelitian

### 3.2.1 Metodologi *Agile*

Penerapan metodologi *Agile* pada pengembangan sistem ini dilakukan dengan mempertimbangkan beberapa elemen penting, yang mencakup kolaborasi dengan *stakeholder*, pengujian berkelanjutan, dan iterasi yang cepat untuk memastikan bahwa sistem yang dikembangkan dapat beradaptasi dengan kebutuhan pengguna dan memaksimalkan efektivitas operasional firma hukum.

### 3.2.2 Langkah-langkah Implementasi *Agile* dalam Penelitian Ini

#### 1. Perencanaan Sprint (*Sprint Planning*)

Pada tahap awal, perencanaan sprint dilakukan untuk menentukan fitur dan fungsi yang harus dikembangkan dalam setiap iterasi. Setiap fitur sistem yang akan dibangun, seperti manajemen kasus, sistem konsultasi hukum *online*, dan pengelolaan dokumen hukum, dipecah menjadi komponen-komponen kecil yang dapat diselesaikan dalam waktu singkat. Sprint pertama difokuskan pada pengembangan fitur inti sistem, seperti fitur login, dashboard utama, dan manajemen data klien.

#### 2. Pengembangan dan Implementasi *Sprint (Sprint Development)*

Dalam setiap sprint, tim pengembang mulai bekerja pada fitur-fitur yang telah ditetapkan dalam perencanaan sprint. Penerapan *framework Laravel* memungkinkan pengembangan cepat dan efektif, dengan pengelolaan *routing*, autentikasi, dan integrasi *database* yang mudah. Setiap bagian dari sistem yang selesai dikembangkan diuji secara langsung dan integrasi antar fitur dilakukan dengan hati-hati untuk memastikan sistem berfungsi dengan baik.

### 3. Pengujian dan Umpan Balik (*Testing and Feedback*)

Setelah setiap sprint, dilakukan pengujian sistem yang melibatkan *stakeholder* dan pengguna utama sistem (seperti pengacara, *admin* firma hukum, dan klien). Pengujian dilakukan untuk mengevaluasi apakah sistem memenuhi harapan pengguna dan apakah fitur-fitur yang dikembangkan dapat dioperasikan dengan lancar.

### 4. Evaluasi Sprint (*Sprint Review*)

Setelah setiap sprint, dilakukan evaluasi sprint untuk menilai apakah tujuan yang telah ditetapkan tercapai. Evaluasi ini mencakup dua aspek utama:

1. Evaluasi fungsionalitas, Apakah fitur yang dibangun berfungsi sesuai dengan kebutuhan pengguna? Misalnya, apakah sistem konsultasi hukum *online* bekerja dengan baik? Apakah manajemen dokumen dapat dilakukan secara digital tanpa hambatan?
2. Evaluasi performa, Apakah sistem cukup cepat dalam merespons permintaan pengguna, dan apakah server dapat menangani banyak pengguna secara bersamaan tanpa gangguan?

### 5. Retrospektif Sprint (*Sprint Retrospective*)

Setelah evaluasi sprint, dilakukan retrospektif sprint, yaitu sesi refleksi oleh tim pengembang mengenai apa yang berjalan baik dan apa yang perlu diperbaiki dalam proses pengembangan. Hal ini membantu meningkatkan efektivitas tim dalam sprint-sprint berikutnya.

Dalam retrospektif ini, berbagai isu yang muncul selama proses pengembangan dibahas, seperti tantangan dalam mengimplementasikan fitur tertentu atau masalah terkait koordinasi antar tim. Tujuan dari retrospektif adalah untuk meningkatkan kolaborasi tim dan efektif pengembangan di sprint selanjutnya.

### 3.2.3 Manfaat Metodologi *Agile*

#### 1. Fleksibilitas dan Responsif Terhadap Perubahan

Dengan menerapkan metodologi *Agile*, pengembangan sistem ini dapat dengan mudah menyesuaikan diri terhadap perubahan kebutuhan yang mungkin terjadi selama proyek berjalan. Jika ada fitur baru yang diperlukan atau perubahan pada fitur yang ada, perubahan tersebut dapat diimplementasikan pada sprint berikutnya tanpa mengganggu keseluruhan proses pengembangan.

#### 2. Kolaborasi yang Lebih Baik dengan *Stakeholder*

*Agile* memfokuskan pada kolaborasi yang erat antara pengembang dan *stakeholder*. Setiap sprint memungkinkan adanya komunikasi langsung dengan pengguna untuk memastikan bahwa sistem yang dibangun sesuai dengan ekspektasi mereka. Dalam konteks firma hukum, kolaborasi ini penting untuk memastikan bahwa sistem memenuhi kebutuhan praktis dari pengacara, staf *administratif*, dan klien.

#### 3. Peningkatan Kualitas Sistem Secara Bertahap

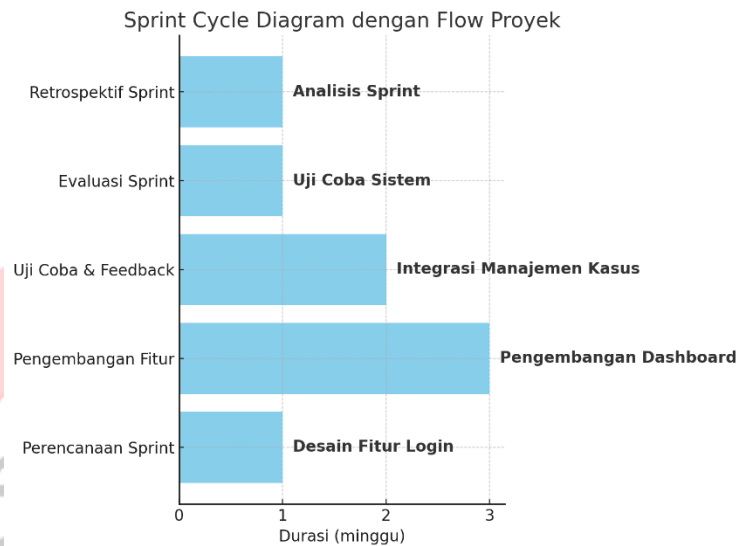
Setiap sprint menghasilkan peningkatan sistem yang dapat diuji dan diterima secara langsung oleh pengguna. Dengan pengujian yang sering dilakukan, kualitas sistem meningkat secara bertahap, meminimalkan risiko adanya masalah besar di akhir proyek.

#### 4. Mempercepat Proses Pengembangan dan Implementasi

*Agile* memungkinkan pengembangan yang lebih cepat dibandingkan dengan metodologi tradisional. Fitur-fitur dasar dapat segera diuji dan diperbaiki setelah masing-masing sprint, sehingga sistem dapat segera digunakan meskipun tidak semua fitur telah selesai. Hal ini juga memungkinkan pengguna untuk segera

menikmati manfaat dari sistem yang sudah ada, meskipun fitur tambahan terus ditambahkan.

### 3.2.4 Sprint Cycle Diagram

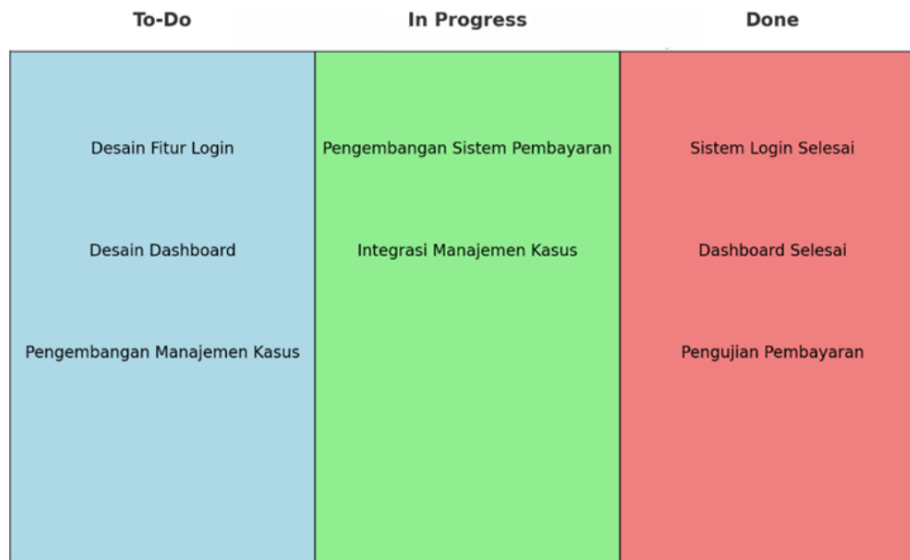


Gambar 3. 1 *Sprint Cycle Diagram*

Gambar 3.1 adalah *Sprint Cycle Diagram* menggambarkan tahapan-tahapan dalam pengembangan sistem selama satu siklus *sprint*. Setiap tahapan sprint terhubung dengan tugas-tugas spesifik yang dilakukan selama proyek ini:

1. Perencanaan Sprint, Desain fitur login yang menjadi bagian penting dalam awal pengembangan.
2. Pengembangan Fitur, Pengembangan dashboard dan integrasi manajemen kasus.
3. Uji Coba & *Feedback*, Pengujian sistem dan pengumpulan umpan balik untuk perbaikan.
4. Evaluasi Sprint, Analisis dan evaluasi hasil sprint untuk merencanakan sprint berikutnya.
5. Retrospektif Sprint, Merefleksikan apa yang berjalan baik dan apa yang perlu diperbaiki.

### 3.2.5 Kanban Board Diagram

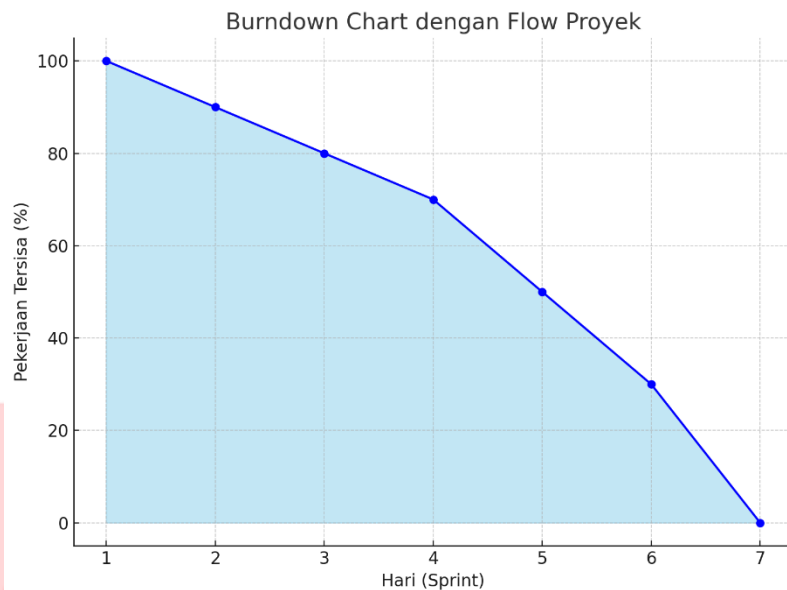


Gambar 3. 2 *Kanban Board Diagram*

Gambar 3.2 adalah *Kanban Board Diagram*. digunakan untuk memvisualisasikan alur pekerjaan dalam proyek ini. Diagram ini dibagi menjadi tiga kolom yang menunjukkan status tugas-tugas yang sedang berjalan:

1. *To-Do*, Menampilkan tugas-tugas yang perlu dikerjakan, seperti desain login *system*, desain dashboard, dan pengembangan manajemen kasus hukum.
2. *In Progress*, Tugas-tugas yang sedang aktif dikerjakan, seperti pengembangan sistem pembayaran dan integrasi sistem manajemen kasus.
3. *Done*, Tugas yang telah selesai, termasuk sistem login yang sudah diuji, dashboard yang sudah fungsional, dan pengujian sistem pembayaran yang berhasil.

### 3.2.6 Burndown Chart



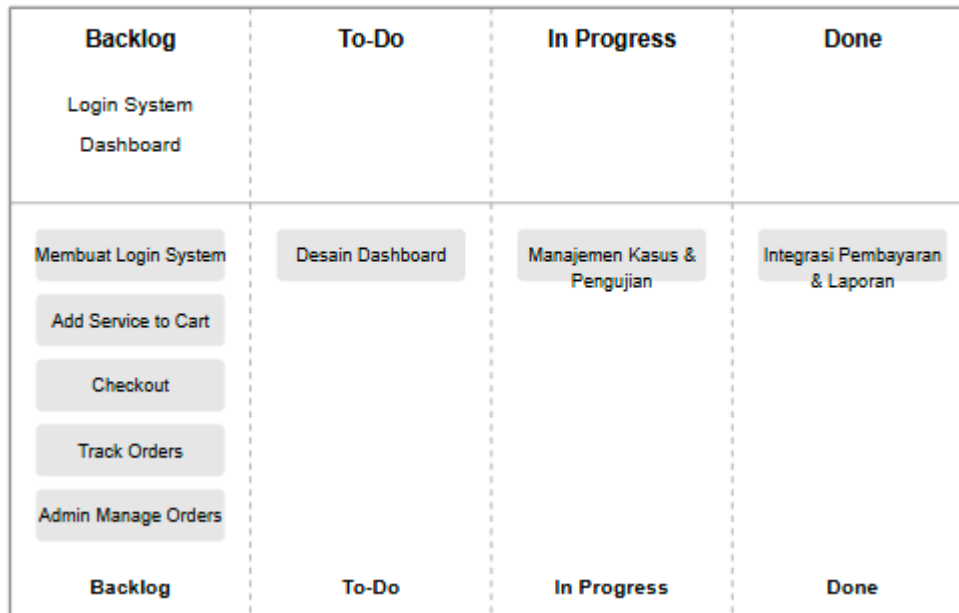
Gambar 3. 3 *Burndown Chart*

Pada Gambar 3.3 *Burndown Chart* memvisualisasikan progres pengembangan selama sprint, menunjukkan berapa banyak pekerjaan yang tersisa setiap hari:

1. Hari 1-2: Fokus pada menyelesaikan desain dan pengembangan login *system* serta dashboard.
2. Hari 3-4: Pengembangan fitur manajemen kasus dan integrasi sistem pembayaran dimulai.
3. Hari 5-6: Pengujian sistem dan integrasi pembayaran dilakukan untuk memastikan fungsionalitas yang baik.
4. Hari 7: Semua tugas telah selesai, dan sprint mencapai penyelesaian penuh.

### 3.2.7 User Story Mapping

#### User Story Mapping Diagram dengan Flow Proyek



Gambar 3. 4 *User Story Mapping Diagram*

Gambar 3.4 Menjelaskan bahwa *User Story Mapping Diagram* digunakan untuk memprioritaskan dan merencanakan *user stories* yang akan dikembangkan dalam proyek ini. Alur kerja pengembangan sistem diatur dengan memetakan *user stories* ke dalam berbagai tahap berikut:

1. *Backlog*, Berisi *user stories* utama seperti login system, dashboard, dan manajemen kasus hukum yang harus segera diprioritaskan.
2. *To-Do*, Fitur pengelolaan dokumen hukum dan konsultasi *online* yang direncanakan untuk dikerjakan setelah tugas dasar selesai.
3. *In Progress*, Fitur seperti integrasi pembayaran dan pengelolaan data klien sedang dalam tahap pengembangan.
4. *Done*, Semua fitur yang telah selesai dikerjakan dan diuji, siap digunakan oleh pengguna, seperti sistem login dan dashboard yang sudah berfungsi.

### 3.2.8 Product Backlog Diagram

#### Product Backlog Diagram dengan Flow Proyek



Gambar 3. 5 *Product Backlog Diagram*

Pada Gambar 3.5 Menjelaskan Bahwa *Product Backlog Diagram* menggambarkan daftar prioritas tugas yang perlu dikerjakan dalam pengembangan sistem. Setiap item dalam *backlog* diurutkan sesuai dengan tingkat kepentingannya dalam pengembangan:

1. *Login System* dan *Dashboard* menjadi prioritas utama yang harus diselesaikan terlebih dahulu.
2. Manajemen Kasus dan Konsultasi *Online* menjadi tugas berikutnya setelah fitur dasar selesai dikembangkan.
3. Integrasi Pembayaran dan *User Feedback* merupakan tugas yang akan dikerjakan di tahap selanjutnya setelah semua fitur utama berfungsi dengan baik.

### 3.3 Analisis Masalah

#### 3.3.1 PIECES Framework

Dengan 20 responden yang terdiri dari 15 staff firma hukum, dan 5 klien, hukum eksternal, analisis masalah dalam pengembangan layanan firma hukum berbasis *web* menggunakan *PIECES Framework* bertujuan untuk mengidentifikasi berbagai aspek yang memengaruhi kinerja, efektivitas, dan kualitas layanan sistem yang akan dirancang. Pendekatan ini melibatkan analisis terhadap enam dimensi utama: *Performance, Information and Data, Economics, Control and Security, Efficiency*, dan *Service*.

Berikut adalah analisis lengkap berdasarkan *PIECES Framework*:

##### 1. *Performance*

Berdasarkan hasil kuesioner dari 20 responden, pada *domain performance* terdapat empat poin pernyataan terkait layanan jasa firma hukum berbasis *web*:

- a. Mudah saat mengakses sistem informasi layanan firma hukum
- b. Sistem informasi merespon dengan cepat setiap perintah pengguna
- c. Sistem dapat digunakan oleh banyak pengguna secara bersamaan dengan stabil
- d. Total waktu yang dibutuhkan dalam melakukan pencarian data hingga menghasilkan informasi dapat dilakukan dengan cepat

Tabel 3. 1 Jawaban Responden Terhadap Variabel *Performance*

Pilihan Jawaban	P1	P2	P3	P4	Total
Sangat Puas	45	35	25	30	135
Puas	35	40	45	35	155
Ragu-ragu	15	20	25	20	85

Tidak Puas	5	5	5	8	23
Sangat Tidak Puas	0	0	0	2	2

$$RK = (5 \times 135 + 4 \times 155 + 3 \times 85 + 2 \times 23 + 1 \times 2) / 400$$

$$RK = 3.87$$

Hasil perhitungan nilai rata-rata dari 20 responden pada *domain performance*, didapatkan nilai 3.87 dan berdasarkan tabel penilaian berada dikategori PUAS. Maka hal ini menandakan indikasi positif bahwa performa sistem layanan firma hukum dapat diterima dan telah berjalan dengan baik.

## 2. *Information and Data*

Pada domain *information and data* terdapat empat poin pernyataan terkait layanan jasa firma hukum:

- a. Data yang disimpan oleh sistem informasi sesuai dengan yang dimasukkan
- b. Informasi yang dihasilkan akurat dan sesuai kebutuhan
- c. Format informasi yang dihasilkan mudah dibaca dan dipahami
- d. Informasi yang diberikan mudah untuk diolah dan digunakan kembali

Tabel 3. 2 Jawaban Responden terhadap *Information and Data*

Pilihan Jawaban	P1	P2	P3	P4	Total
Sangat Puas	50	45	40	35	170
Puas	35	40	45	45	165
Ragu-ragu	12	13	12	15	52
Tidak Puas	3	2	3	5	13
Sangat Tidak Puas	0	0	0	0	0

$$RK = (5 \times 170 + 4 \times 165 + 3 \times 52 + 2 \times 13 + 1 \times 0) / 400$$

$$RK = 3.93$$

Hasil perhitungan nilai rata-rata pada domain *Information and Data*, didapatkan nilai 3.93 dan berdasarkan tabel penilaian berada dikategori PUAS. Maka hal ini menandakan indikasi positif bahwa kualitas informasi sistem telah berjalan dengan baik.

### 3. *Economics*

Pada domain *economics* terdapat dua poin pernyataan terkait layanan jasa firma hukum:

- a. Terdapat Efektivitas biaya dalam penggunaan sistem dibanding manual
- b. Sistem dapat mengurangi kesalahan yang berakibat pada kerugian finansial

Tabel 3. 3 Jawaban Responden terhadap Variabel *Economics*

Pilihan Jawaban	P1	P2	Total
Sangat Puas	25	28	53
Puas	45	42	87
Ragu-ragu	25	25	50
Tidak Puas	5	5	10
Sangat Tidak Puas	0	0	0

$$RK = (5 \times 53 + 4 \times 87 + 3 \times 50 + 2 \times 10 + 1 \times 0) / 200$$

$$RK = 3.92$$

Hasil perhitungan nilai rata-rata pada domain *Economics*, didapatkan nilai 3.92 dan berdasarkan tabel penilaian berada dikategori PUAS. Maka hal ini menandakan sistem telah memberikan manfaat ekonomis.

#### 4. *Control and Security*

Pada domain *control and security* terdapat tiga poin pernyataan:

- a. Terdapat pengontrolan akses pengguna yang baik
- b. Sistem memiliki keamanan data yang terjamin
- c. Sistem dapat mencegah akses tidak sah

Tabel 3. 4 Jawaban Responden terhadap variabel *Control and Security*

Pilihan Jawaban	P1	P2	P3	Total
Sangat Puas	35	30	35	100
Puas	40	45	40	125
Ragu-ragu	20	20	20	60
Tidak Puas	5	5	5	15
Sangat Tidak Puas	0	0	0	0

$$RK = (5 \times 100 + 4 \times 125 + 3 \times 60 + 2 \times 15 + 1 \times 0) / 300$$

$$RK = 3.90$$

Hasil perhitungan nilai rata-rata pada domain *Control and Security*, didapatkan nilai 3.90 dan berdasarkan tabel penilaian berada dikategori PUAS. Maka hal ini menandakan sistem memiliki kontrol dan keamanan yang baik.

#### 5. *Efficiency*

Pada domain *efficiency* terdapat dua poin pernyataan:

- a. Penggunaan sistem mempercepat proses layanan hukum
- b. Sistem memudahkan proses dokumentasi dan pelaporan

Tabel 3. 5 Jawaban Responden terhadap Variabel *Efficiency*

Pilihan Jawaban	P1	P2	Total
Sangat Puas	30	29	59
Puas	48	50	98
Ragu-ragu	17	16	33
Tidak Puas	5	5	10
Sangat Tidak Puas	0	0	0

$$RK = (5 \times 59 + 4 \times 98 + 3 \times 33 + 2 \times 10 + 1 \times 0) / 200$$

$$RK = 4.03$$

Hasil perhitungan nilai rata-rata pada domain *Efficiency*, didapatkan nilai 4.03 dan berdasarkan tabel penilaian berada dikategori PUAS. Maka hal ini menandakan sistem telah berjalan efektif.

#### 6. *Service*

Pada domain *service* terdapat tiga poin pernyataan:

- a. Sistem memberikan layanan yang mudah dipahami
- b. Sistem terintegrasi dengan baik antar modul
- c. Sistem memberikan kepuasan dalam proses layanan hukum

Tabel 3. 6 Jawaban Responden terhadap variabel *Service*

Pilihan Jawaban	P1	P2	P3	Total
Sangat Puas	40	35	38	113
Puas	45	48	45	138
Ragu-ragu	12	13	14	39
Tidak Puas	3	4	3	10
Sangat Tidak Puas	0	0	0	0

$$RK = (5 \times 113 + 4 \times 138 + 3 \times 39 + 2 \times 10 + 1 \times 0) / 300$$

$$RK = 3.95$$

Hasil perhitungan nilai rata-rata pada domain *Service*, didapatkan nilai 3.95 dan berdasarkan tabel penilaian berada dikategori PUAS.

Tabel 3. 7 Nilai Skor Rekap Variabel PIECES

Domain	Rata-rata	Kategori
<i>Performance</i>	3.87	Puas
<i>Information and Data</i>	3.93	Puas
<i>Economics</i>	3.92	Puas
<i>Control and Security</i>	3.90	Puas
<i>Efficiency</i>	4.03	Puas
<i>Service</i>	3.95	Puas

Nilai RK ini kemudian digunakan untuk menentukan kategori kepuasan, dimana:

- a.  $RK > 4.2 =$  Sangat Puas
- b.  $3.4 < RK \leq 4.2 =$  Puas
- c.  $2.6 < RK \leq 3.4 =$  Cukup Puas
- d.  $1.8 < RK \leq 2.6 =$  Tidak Puas
- e.  $RK \leq 1.8 =$  Sangat Tidak Puas

### 3.4 Requirement Elicitation

Dengan melibatkan 4 responden yang terdiri dari 4 staff firma hukum senior, proses pengumpulan persyaratan (*requirement elicitation*) dilakukan untuk mengklarifikasi dan

memahami secara mendalam kebutuhan sistem yang diinginkan oleh pengguna atau pelanggan.

Proses elisitasi kebutuhan ini dilakukan melalui wawancara mendalam dengan 4 responden kunci untuk memastikan sistem yang dikembangkan benar-benar sesuai dengan kebutuhan operasional firma hukum dan ekspektasi klien.

### 3.4.1 Tahap 1

Pada tahap awal, berdasarkan *input* dari 4 responden, permintaan dari pengguna akan dibagi menjadi tiga bagian yaitu *Mandatory* (penting), *Desirable* (perlu dipertimbangkan), dan *Inessential* (tidak relevan dengan rancangan sistem).

Tabel 3. 8 *Elicitation* Tahap 1

No	User Ingin Sistem Dapat:
1	Menampilkan Menu <i>Login</i>
2	Menampilkan Menu <i>Dashboard</i>
3	Menampilkan Total <i>Sales, Total Orders, Customers</i>
4	Menampilkan Grafik Penjualan
5	Menampilkan Status Stok Layanan
6	Memiliki fitur <i>Add, Edit, Delete</i> Layanan
7	Menambahkan Kategori Pada Layanan
8	Memberikan fitur <i>Search</i> Layanan
9	Menampilkan Daftar Layanan dan Harga
10	Memiliki Fitur <i>Export</i> Layanan
11	Membuat Laporan Penjualan dalam format Excel/PDF
12	Menampilkan Grafik Pengunjung
13	Menampilkan Harga Layanan

14	Menyediakan Fitur <i>Logout</i>
15	Menyediakan Fitur Pilihan Bahasa

### 3.4.2 Tahap 2

Tahap 2 ini adalah hasil dari pengelompokan dari Elisitasi Tahap 1 berdasarkan evaluasi bersama 4 responden dengan memakai metode *MDI* yang terdiri dari *Mandatory* (*Requirement* penting dan tidak boleh dihilangkan dari perancangan sistem), *Desirable* (*Requirement* ini tidak terlalu penting, tapi jangan dihilangkan. Namun dapat menggunakannya agar membuat sistem lebih sempurna), dan *Inessential* (*Requirement* ini berada di luar sistem yang sedang dikembangkan dan harus dihapus)

Tabel 3. 9 *Elicitation* Tahap 2

No	User Ingin Sistem Dapat:	Mandatory	Desirable	Inessential
1	Menampilkan Menu <i>Login</i>	✓		
2	Menampilkan Menu <i>Dashboard</i>	✓		
3	Menampilkan Total <i>Sales</i> , Total <i>Orders</i> , <i>Customers</i>	✓		
4	Menampilkan Grafik Penjualan		✓	
5	Menampilkan Status Layanan	✓		
6	Memiliki fitur <i>Add</i> , <i>Edit</i> , <i>Delete</i> Layanan	✓		
7	Menambahkan Kategori Pada Layanan		✓	
8	Memberikan fitur <i>Search</i> Layanan		✓	
9	Menampilkan Daftar Layanan dan Harga	✓		
10	Memiliki Fitur <i>Export</i> Layanan		✓	

11	Membuat Laporan Penjualan dalam format Excel/PDF	✓		
12	Menampilkan Grafik Pengunjung		✓	
13	Menampilkan Harga Layanan	✓		
14	Menyediakan Fitur <i>Logout</i>	✓		
15	Menyediakan Fitur Pilihan Bahasa			✓

### 3.4.3 Tahap Final

Pada tahap akhir, berdasarkan konsensus dari 4 responden, kebutuhan yang *Mandatory* dan beberapa *Desirable* yang dianggap penting akan masuk dalam final draft RE.

Tabel 3. 10 *Elicitation* Tahap 3

No	User Ingin Sistem Dapat:
1	Menampilkan Menu <i>Login</i>
2	Menampilkan Menu <i>Dashboard</i>
3	Menampilkan Total <i>Sales, Total Orders, Customers</i>
4	Menampilkan Grafik Penjualan
5	Menampilkan Status Layanan
6	Memiliki fitur <i>Add, Edit, Delete</i> Layanan
7	Menambahkan Kategori Pada Layanan
8	Memberikan fitur <i>Search</i> Layanan
9	Menampilkan Daftar dan Harga
10	Memiliki Fitur <i>Export</i> Layanan
11	Membuat Laporan Penjualan dalam format Excel/PDF

12	Menampilkan Harga Layanan
13	Menyediakan Fitur <i>Logout</i>

### 3.5 Jadwal Penelitian

Rencana jadwal penelitian ini mencakup kegiatan utama beserta rentang waktu pelaksanaannya, yang disusun dalam periode Agustus hingga Desember. Berikut adalah rincian aktivitas berdasarkan fase dan bulan:

#### 1. MARET 2025 - Analisis Kebutuhan

Minggu 1-2: Tinjauan pustaka dan pencarian referensi tentang sistem hukum berbasis *web* serta *framework Laravel*. Minggu 3-4: Penyebaran kuesioner *PIECES Framework* kepada 20 responden (10 staff firma hukum, 5 klien, 5 praktisi eksternal) dan wawancara *stakeholder* untuk analisis kebutuhan sistem.

#### 2. APRIL 2025 - Perancangan Sistem

Minggu 1-2: Perancangan arsitektur sistem menggunakan pola MVC dan rancangan *database* lengkap dengan ERD. Minggu 3-4: Pembuatan diagram UML (*Use Case, Activity, Sequence, Class*) dan rancangan UI/UX (*wireframe*) untuk semua halaman sistem.

#### 3. MEI 2025 - Implementasi *Sprint 1-2*

Minggu 1-2 (*Sprint 1*): Implementasi sistem autentikasi (*login/registrasi*) untuk *admin* dan *customer*, setup *framework Laravel*, dan pembuatan *dashboard* dasar.

Minggu 3-4 (*Sprint 2*): Implementasi katalog layanan hukum, sistem kategori, keranjang belanja, dan halaman detail layanan.

#### 4. JUNI 2025 - Implementasi *Sprint 3-4*

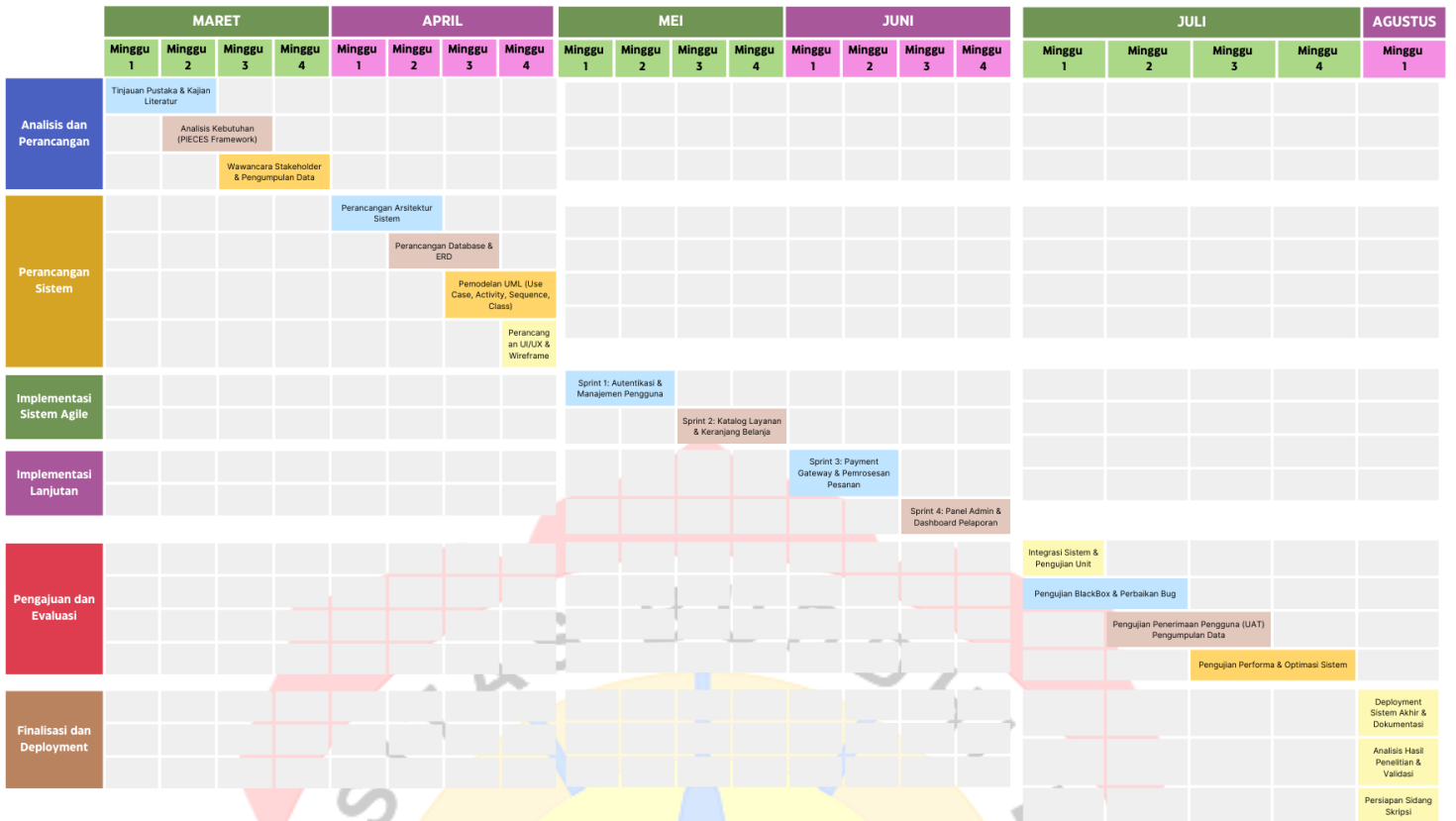
Minggu 1-2 (*Sprint 3*): Implementasi sistem pembayaran dengan integrasi *payment gateway*, manajemen pesanan, dan fitur *tracking* pesanan. Minggu 3-4 (*Sprint 4*): Implementasi panel *admin* lengkap dengan fitur manajemen *customer*, layanan, dan sistem laporan penjualan.

#### 5. JULI 2025 - Pengujian & Evaluasi

Minggu 1-2: Pengujian sistem (*unit testing* dan *BlackBox testing*), perbaikan *bug*, dan optimasi performa. Minggu 3-4: Pelaksanaan UAT dengan 10 responden, analisis *feedback* pengguna, dan perbaikan sistem berdasar hasil UAT.

#### 6. AGUSTUS 2025 - Finalisasi

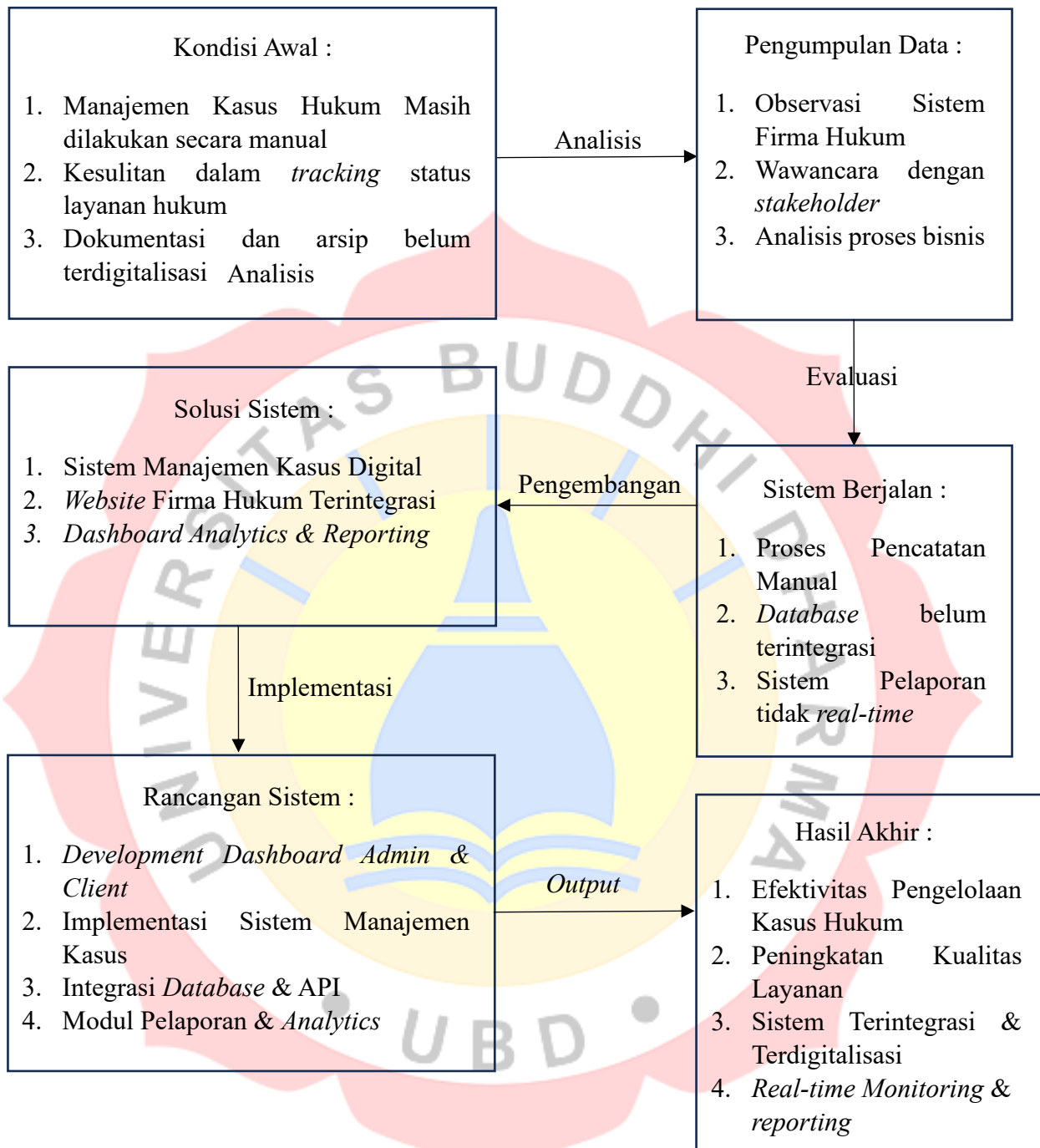
Minggu 1: *Deployment* sistem ke server produksi, testing sistem *online*, dan finalisasi dokumentasi. Minggu 2: Persiapan sidang skripsi dan *review* akhir. Sidang: 13 Agustus 2025.



Gambar 3. 6 Gantt Chart



### 3.6 Kerangka Pemikiran



Gambar 3. 7 Kerangka Pemikiran

### 3.7 Prosedur Sistem Usulan

Prosedur sistem usulan merupakan rancangan alur kerja sistem yang diusulkan untuk mengoptimalkan proses bisnis layanan jasa hukum berbasis *e-commerce*. Sistem ini dirancang untuk memudahkan interaksi antara dua aktor utama yaitu *customer*, dan *admin* dalam satu *platform* terintegrasi.

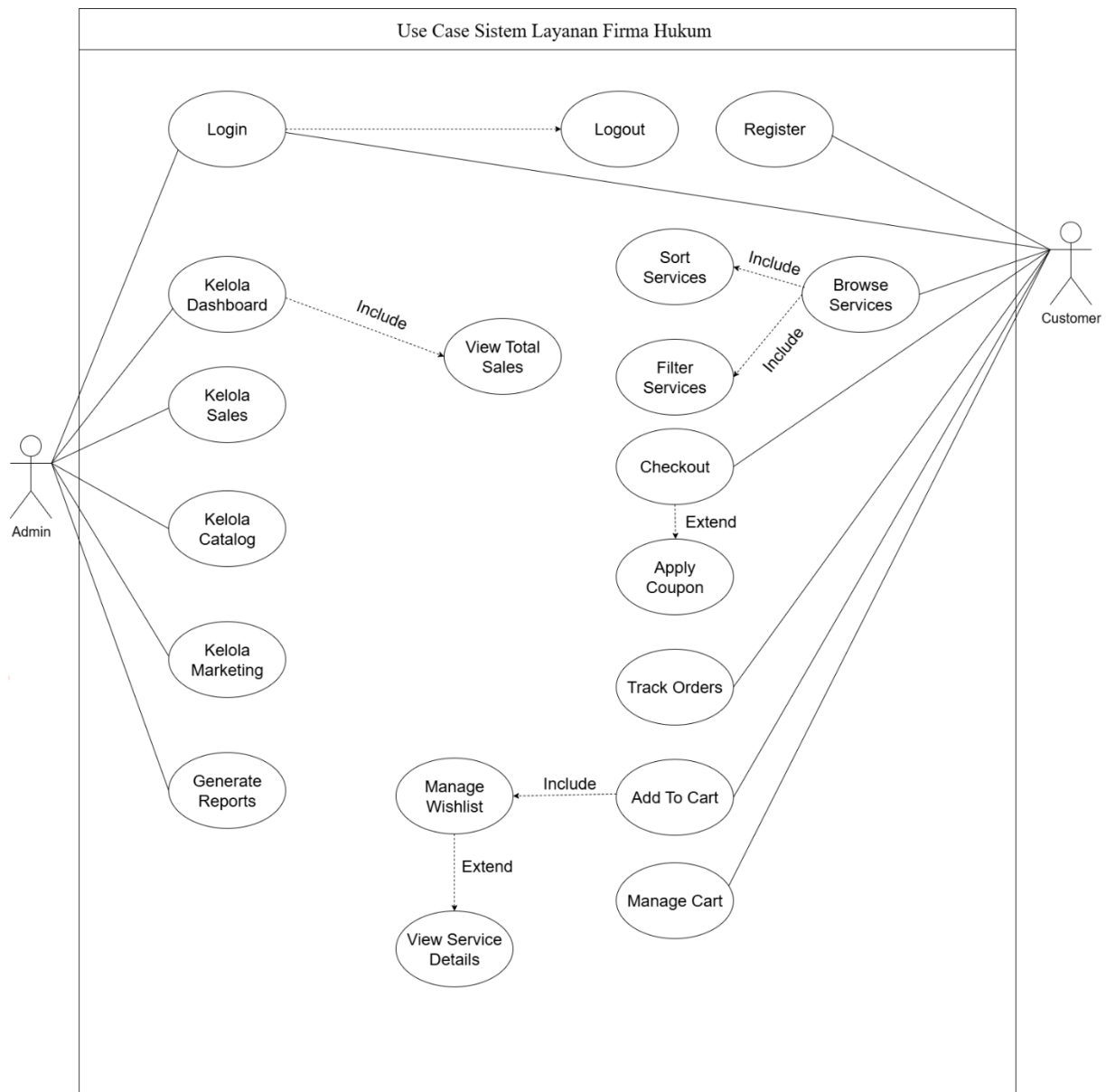
Dalam prosedur sistem usulan ini, setiap aktor memiliki peran dan tanggung jawab yang berbeda namun saling terkait untuk menciptakan layanan yang efektif :

Tabel 3. 11 Prosedur Sistem Usulan

No	Aktor	Peran
1	<i>Customer</i>	<ol style="list-style-type: none"><li>a. Melakukan Pendaftaran akun kedalam <i>website</i></li><li>b. Melakukan <i>Login</i> sistem</li><li>c. Melihat Katalog atau Halaman utama</li><li>d. Memilih kategori layanan hukum yang dibutuhkan</li><li>e. Melakukan konsultasi awal secara <i>online</i></li><li>f. Melakukan pemesanan layanan hukum</li><li>g. Melakukan pembayaran</li><li>h. Menerima notifikasi status pesanan</li><li>i. Memberikan <i>review</i> dan <i>rating</i></li></ol>
2	<i>Admin</i>	<ol style="list-style-type: none"><li>a. Mengelola data <i>user/customer</i></li><li>b. Memverifikasi pendaftaran <i>customer</i></li><li>c. Mengelola katalog layanan hukum</li><li>d. Menerima dan memproses pesanan</li><li>e. Menghubungkan <i>customer</i> dengan pengacara yang sesuai</li><li>f. Memverifikasi pembayaran</li><li>g. Monitoring progress layanan</li><li>h. Mengelola <i>review</i> dan <i>rating</i></li><li>i. Membuat laporan transaksi</li></ol>

### 3.8 Rancangan Sistem Usulan

#### 3.8.1 Use Case Diagram



Gambar 3. 8 Use Case Diagram

Pada Gambar 3. 7 Use case diagram di atas menggambarkan interaksi antara pengguna (*actors*) dengan Sistem Layanan Firma Hukum. Diagram ini menunjukkan dua aktor utama yaitu *Admin* dan *Customer*, dimana masing-masing memiliki akses ke fungsi-fungsi yang berbeda dalam sistem.

1. Aktor dalam Sistem:

- a. *Admin*: Berperan sebagai pengelola sistem dengan akses ke fungsi-fungsi manajemen
- b. *Customer*: Pengguna akhir yang dapat mengakses layanan dan melakukan transaksi

2. Fungsi Utama untuk *Admin*:

- a. Kelola *Dashboard*: Mencakup monitoring stok dan melihat total penjualan
- b. Kelola *Sales*: Manajemen penjualan
- c. Kelola *Catalog*: Pengelolaan katalog layanan
- d. Kelola *Marketing*: Manajemen pemasaran
- e. *Generate Reports*: Pembuatan laporan

3. Fungsi Utama untuk *Customer*:

- a. *Login dan Register*: Akses masuk sistem dan pendaftaran akun baru
- b. *Browse Services*: Menjelajahi layanan dengan fitur *sort* dan *filter services*
- c. *Checkout*: Proses pembayaran dengan opsi *apply coupon*
- d. *Manage Cart*: Pengelolaan keranjang belanja
- e. *Track Orders*: Pelacakan pesanan
- f. *Manage Wishlist*: Pengelolaan daftar keinginan

4. Relasi dalam Diagram:

- a. *Include*: Menunjukkan fitur yang wajib dijalankan, seperti:

- I. *Browse Services include Sort Services dan Filter Services*
  - II. *Kelola Dashboard include View Total Sales*
  - III. *Manage Wishlist include Add to Cart*
- a. *Extend: Menunjukkan fitur opsional, seperti:*
- I. *Checkout extend Apply Coupon*
  - II. *Manage Wishlist extend View Service Details*

5. Alur Proses:

- a. *Customer* dapat melakukan registrasi dan *login* ke sistem
- b. Setelah *login*, *customer* dapat menjelajahi layanan, mengelola keranjang, dan melakukan *checkout*
- c. *Admin* mengelola berbagai aspek sistem mulai dari *dashboard*, penjualan, katalog, hingga pemasaran
- d. Sistem menyediakan fitur pelacakan pesanan bagi *customer* dan pembuatan laporan bagi *admin*

*Use case diagram* ini memberikan gambaran komprehensif tentang fungsionalitas sistem dan bagaimana pengguna berinteraksi dengannya, memudahkan pemahaman tentang cakupan dan batasan sistem yang dikembangkan.

### 3.8.2 Use Case Diagram Scenario *Admin, Customer*

#### 1. *Scenario Login*

Tabel 3. 12 *Scenario Login*

<b>Identifikasi</b>	
Nama <i>Use Case</i>	<i>Login</i>
Aktor	<i>Admin, Customer</i>
Tujuan	Masuk ke dalam sistem menggunakan kredensial yang valid
Kondisi Awal	Aktor berada di halaman <i>login</i>
Skenario Utama	<ol style="list-style-type: none"><li>1. Aktor memasukkan username dan password</li><li>2. Sistem memvalidasi kredensial</li><li>3. Sistem mengautentikasi pengguna</li><li>4. Sistem mengarahkan ke halaman <i>dashboard</i> sesuai role</li></ol>
Skenario Alternatif	Jika kredensial tidak valid: <ol style="list-style-type: none"><li>1. Sistem menampilkan pesan error</li><li>2. Kembali ke langkah 1</li></ol>
Kondisi Akhir	Aktor berhasil masuk ke sistem

#### 2. *Scenario Logout*

Tabel 3. 13 *Scenario Logout*

<b>Identifikasi</b>	
Nama <i>Use Case</i>	<i>Logout</i>
Aktor	<i>Admin, Customer</i>
Tujuan	Keluar dari sistem
Kondisi Awal	Aktor sudah masuk dalam sistem
Skenario Utama	<ol style="list-style-type: none"><li>1. Aktor memilih opsi <i>Logout</i></li><li>2. Sistem menutup sesi pengguna</li><li>3. Sistem mengarahkan aktor kembali ke halaman <i>login</i></li></ol>

Skenario Alternatif	Tidak ada
Kondisi Akhir	Aktor berhasil keluar dari sistem

### 3. *Scenario Kelola Dashboard*

Tabel 3. 14 *Scenario Kelola Dashboard*

<b>Identifikasi</b>	
Nama <i>Use Case</i>	<i>Kelola Dashboard</i>
Aktor	<i>Admin</i>
Tujuan	Mengelola ringkasan data stok dan penjualan
Kondisi Awal	<i>Admin</i> berada di dalam halaman utama sistem
Skenario Utama	<ol style="list-style-type: none"> <li>1. <i>Admin</i> memilih menu “<i>Kelola dashboard</i>”</li> <li>2. Sistem menampilkan ringkasan stok layanan</li> <li>3. Sistem menampilkan total penjualan</li> </ol>
Skenario Alternatif	Tidak ada data yang tersedia : <ol style="list-style-type: none"> <li>1. Sistem menampilkan pesan “Data tidak tersedia”</li> </ol>
Kondisi Akhir	<i>Admin</i> berhasil mengakses data <i>dashboard</i>

### 4. *Scenario Browse Service*

Tabel 3. 15 *Scenario Browse Service*

<b>Identifikasi</b>	
Nama <i>Use Case</i>	<i>Browse Service</i>
Aktor	<i>Customer</i>
Tujuan	Menelusuri layanan yang tersedia dalam sistem
Kondisi Awal	Aktor berada di halaman utama layanan
Skenario Utama	<ol style="list-style-type: none"> <li>1. Aktor membuka menu “<i>Browse Service</i>”</li> <li>2. Sistem menampilkan daftar layanan</li> <li>3. Aktor dapat memilih opsi “<i>Sort Services</i>” atau “<i>Filter Services</i>”</li> </ol>
Skenario Alternatif	Jika tidak ada layanan:

	1. Sistem menampilkan pesan “Layanan tidak tersedia”
Kondisi Akhir	Aktor melihat daftar layanan yang tersedia

### 5. Scenario Sort Services

Tabel 3. 16 Scenario Sort Services

<b>Identifikasi</b>	
Nama Use Case	<i>Sort Services</i>
Aktor	<i>Customer</i>
Tujuan	Menyortir Layanan berdasarkan kriteria tertentu
Kondisi Awal	Aktor berada di halaman daftar layanan
Skenario Utama	<ol style="list-style-type: none"> <li>1. Aktor memilih kriteria sortir</li> <li>2. Sistem mengatur daftar layanan berdasarkan kriteria tersebut</li> </ol>
Skenario Alternatif	Tidak ada layanan yang cocok dengan sortir: <ol style="list-style-type: none"> <li>1. Sistem menampilkan pesan “Layanan tidak ditemukan”</li> </ol>
Kondisi Akhir	Layanan di sortir sesuai kriteria

### 6. Scenario Checkout

Tabel 3. 17 Scenario Checkout

<b>Identifikasi</b>	
Nama Use Case	<i>Checkout</i>
Aktor	<i>Customer</i>
Tujuan	Menyelesaikan transaksi pembelian layanan
Kondisi Awal	Aktor memiliki layanan dalam keranjang
Skenario Utama	<ol style="list-style-type: none"> <li>1. Aktor memilih opsi “<i>Checkout</i>”</li> <li>2. Sistem menampilkan ringkasan pembayaran</li> <li>3. Aktor memasukkan metode pembayaran</li> <li>4. Sistem memproses pembayaran</li> <li>5. Sistem mengonfirmasi transaksi berhasil</li> </ol>

Skenario Alternatif	Jika pembayaran gagal: 1. Sistem menampilkan pesan “Pembayaran gagal” 2. Aktor kembali ke metode pembayaran
Kondisi Akhir	Transaksi selesai dan layanan dikonfirmasi

### 7. Scenario Add to Cart

Tabel 3. 18 Scenario Add to Cart

<b>Identifikasi</b>	
Nama Use Case	<i>Add to Cart</i>
Aktor	<i>Customer</i>
Tujuan	Menambahkan layanan ke dalam keranjang
Kondisi Awal	Aktor berada di halaman detail layanan
Skenario Utama	1. Aktor memilih layanan yang diinginkan 2. Aktor mengklik “Add to Cart” 3. Sistem menambahkan layanan ke keranjang
Skenario Alternatif	Jika layanan sudah ada di keranjang 1. Sistem menampilkan pesan “Layanan sudah ada di keranjang”
Kondisi Akhir	Layanan ditambahkan ke keranjang

### 8. Scenario Track Orders

Tabel 3. 19 Scenario Track Orders

<b>Identifikasi</b>	
Nama Use Case	<i>Track Orders</i>
Aktor	<i>Customer</i>
Tujuan	Melacak status pemesanan layanan
Kondisi Awal	Aktor telah melakukan pemesanan
Skenario Utama	1. Aktor membuka menu “Track Orders” 2. Sistem menampilkan status pemesanan layanan

Skenario Alternatif	Jika data pesanan tidak ditemukan: 1. Sistem menampilkan pesan “Pesanan tidak ditemukan”
Kondisi Akhir	Aktor melihat status pemesanan

### 9. Scenario Manage Wishlist

Tabel 3. 20 Scenario Manage Wishlist

<b>Identifikasi</b>	
Nama Use Case	<i>Manage Wishlist</i>
Aktor	<i>Customer</i>
Tujuan	Mengelola Daftar layanan yang ingin disimpan
Kondisi Awal	Aktor berada di halaman layanan atau wishlist
Skenario Utama	<ol style="list-style-type: none"> <li>1. Aktor memilih layanan untuk ditambahkan ke wishlist</li> <li>2. Sistem menambahkan layanan ke daftar wishlist</li> <li>3. Aktor dapat melihat dan menghapus layanan dari wishlist</li> </ol>
Skenario Alternatif	Jika layanan sudah ada di wishlist:
Kondisi Akhir	Wishlist diperbaharui sesuai tindakan aktor

### 10. Scenario View Service Details

Tabel 3. 21 Scenario View Service Details

<b>Identifikasi</b>	
Nama Use Case	<i>View Service Details</i>
Aktor	<i>Customer</i>
Tujuan	Melihat detail lengkap dari suatu layanan
Kondisi Awal	Aktor berada di halaman daftar layanan
Skenario Utama	<ol style="list-style-type: none"> <li>1. Aktor memilih layanan dari daftar</li> <li>2. Sistem menampilkan detail layanan, termasuk deskripsi, harga, dan informasi tambahan</li> </ol>

Skenario Alternatif	Jika layanan tidak ditemukan 1. Sistem menampilkan pesan “Layanan tidak tersedia”
Kondisi Akhir	Detail layanan ditampilkan pada aktor

### 11. Scenario Apply Coupon

Tabel 3. 22 Scenario Apply Coupon

<b>Identifikasi</b>	
Nama Use Case	Apply Coupon
Aktor	Customer
Tujuan	Menerapkan Kupon diskon saat <i>checkout</i>
Kondisi Awal	Aktor berada di halaman <i>checkout</i>
Skenario Utama	<ol style="list-style-type: none"> <li>1. Aktor memasukkan kode kupon</li> <li>2. Sistem memvalidasi kode kupon</li> <li>3. Jika valid, sistem menerapkan diskon pada total pembayaran</li> </ol>
Skenario Alternatif	Jika kode kupon tidak valid: <ol style="list-style-type: none"> <li>1. Sistem menampilkan pesan “Kode kupon tidak valid”</li> </ol>
Kondisi Akhir	Diskon diterapkan pada total pembayaran

### 12. Scenario View Total Sales

<b>Identifikasi</b>	
Nama Use Case	View Total Sales
Aktor	Admin
Tujuan	Melihat total penjualan layanan
Kondisi Awal	Admin berada di halaman <i>dashboard</i>
Skenario Utama	<ol style="list-style-type: none"> <li>1. Admin memilih opsi “View Total Sales”</li> <li>2. Sistem menghitung total penjualan layanan</li> <li>3. Sistem menampilkan total penjualan berdasarkan periode tertentu</li> </ol>

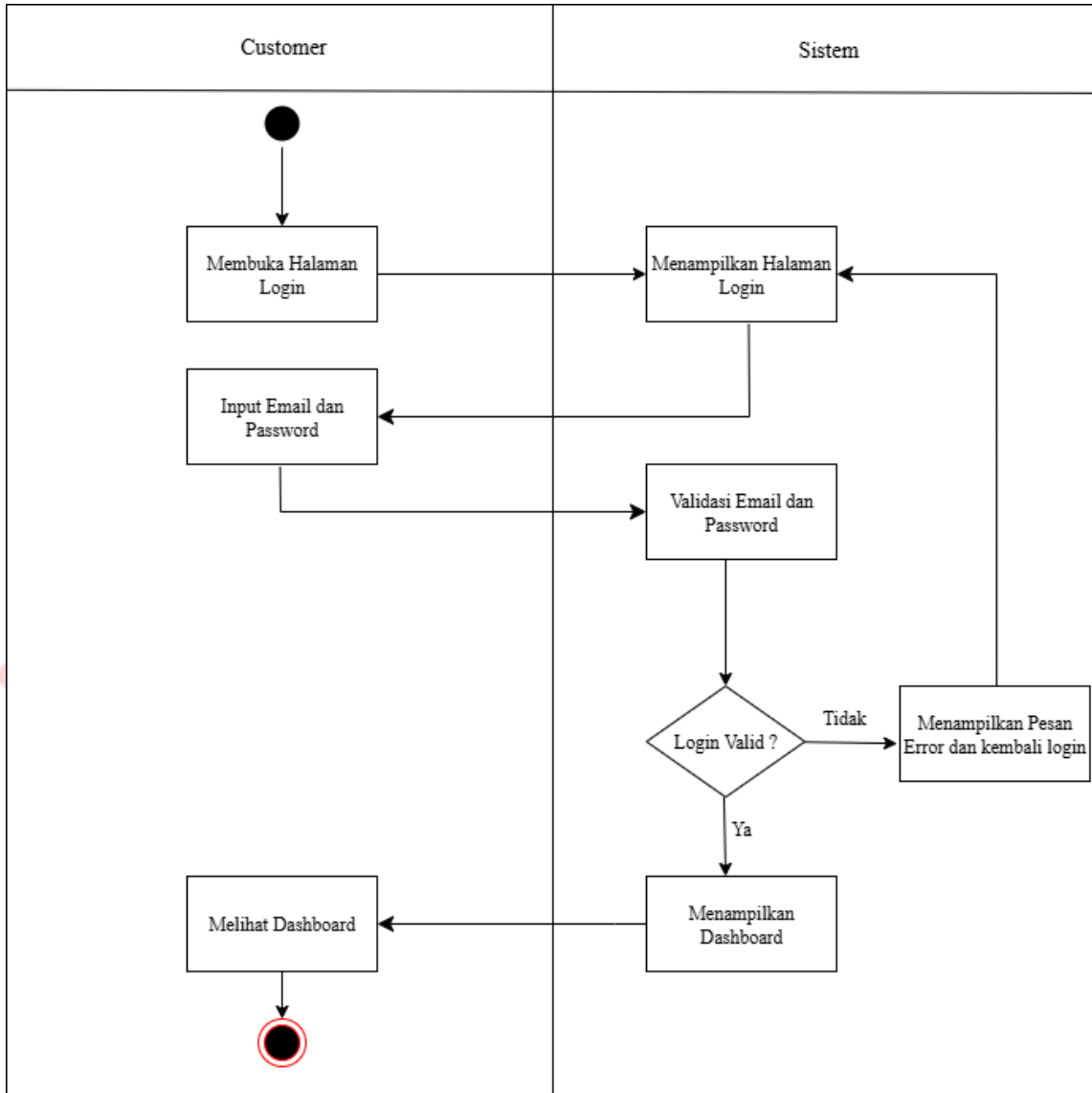
Skenario Alternatif	Jika tidak ada data penjualan: 1. Sistem menampilkan pesan “Data penjualan tidak tersedia”
Kondisi Akhir	<i>Admin</i> melihat laporan total penjualan

### 13. Scenario Generate Reports

<b>Identifikasi</b>	
Nama <i>Use Case</i>	<i>Generate Reports</i>
Aktor	<i>Admin</i>
Tujuan	Membuat laporan berdasarkan data layanan
Kondisi Awal	<i>Admin</i> berada di halaman <i>dashboard</i> atau menu laporan
Skenario Utama	<ol style="list-style-type: none"> <li>1. <i>Admin</i> memilih opsi “<i>Generate reports</i>”</li> <li>2. Sistem memproses data layanan, penjualan, dan stok</li> <li>3. Sistem menghasilkan laporan dalam format yang dipilih (PDF/Excel)</li> </ol>
Skenario Alternatif	Jika data tidak tersedia <ol style="list-style-type: none"> <li>1. Sistem menampilkan pesan “Data tidak tersedia untuk laporan”</li> </ol>
Kondisi Akhir	Laporan berhasil dihasilkan dan siap diunduh.

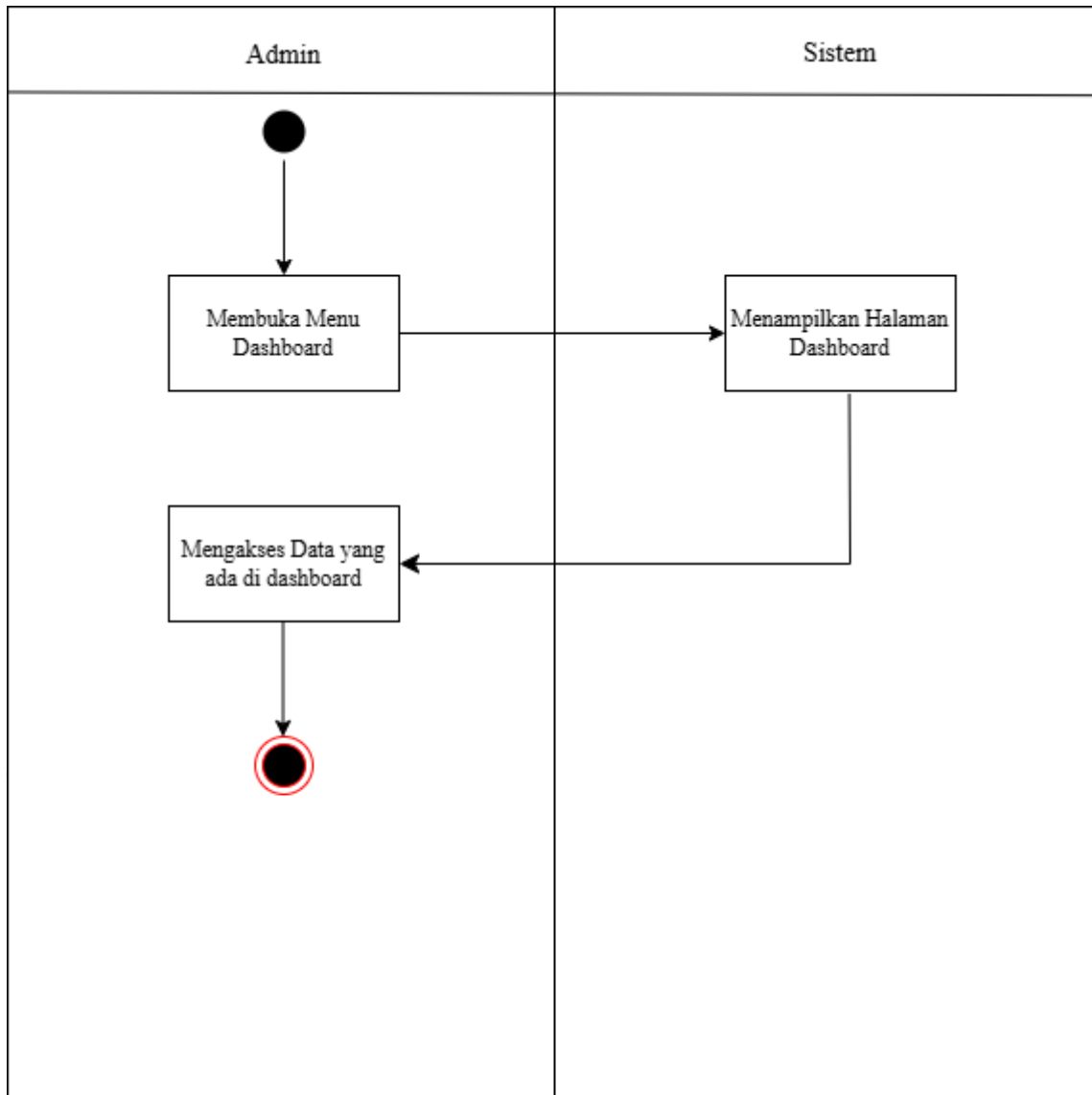
### 3.8.3 Activity Diagram

#### 1. Activity Diagram Admin Login



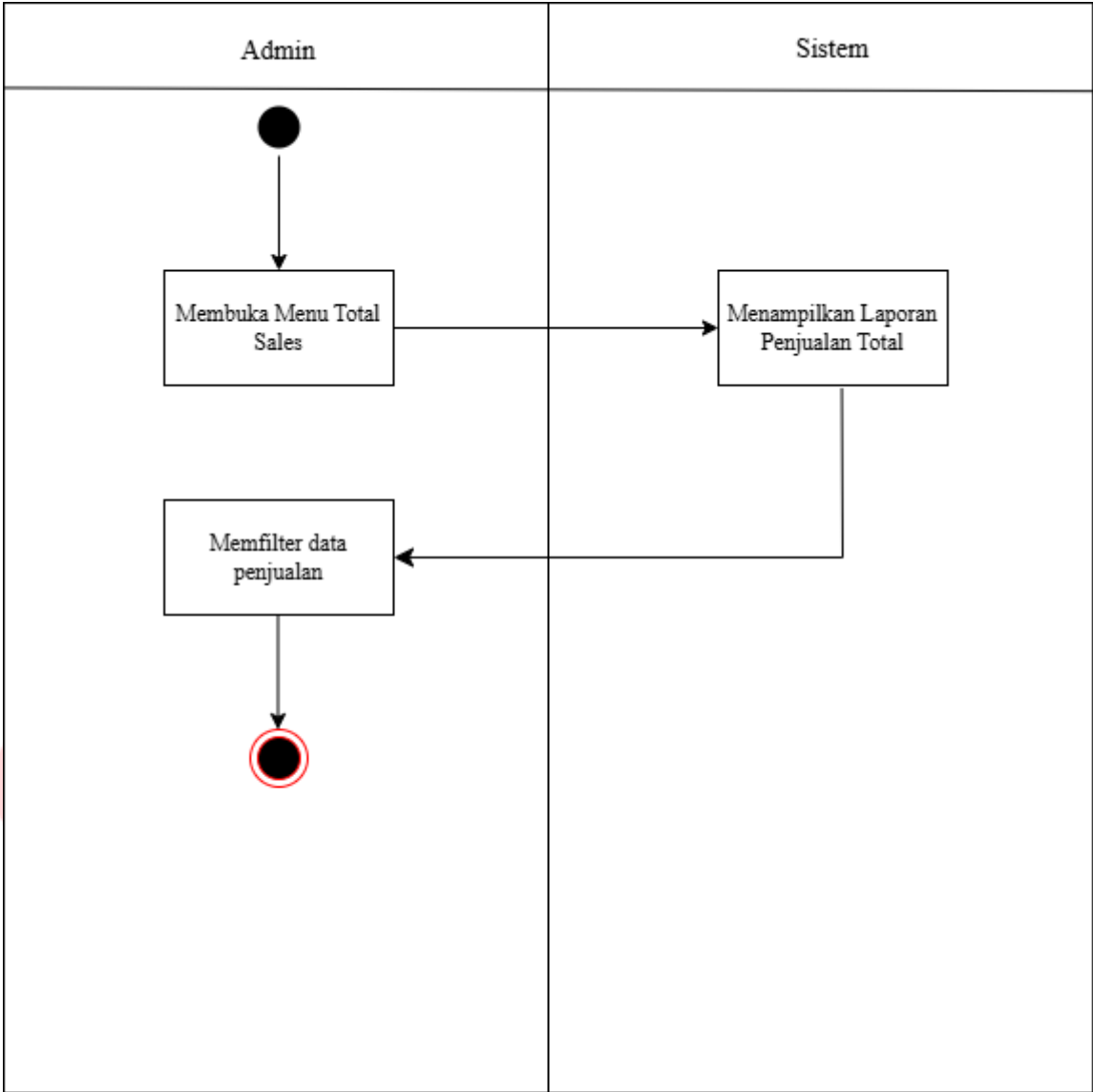
Gambar 3. 9 Activity Diagram Admin Login

## 2. Activity Diagram Kelola Dashboard



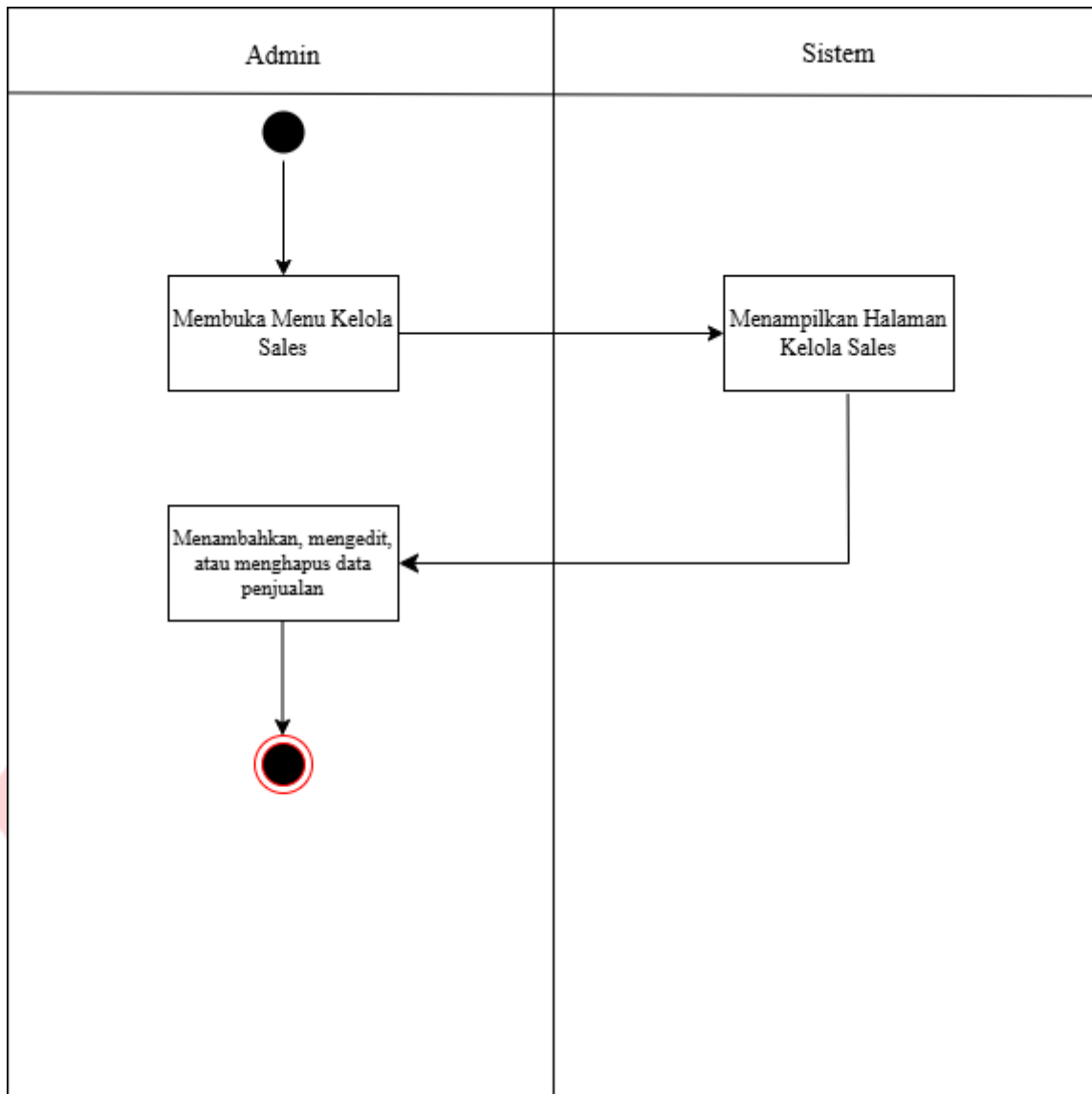
Gambar 3. 10 Activity Diagram Kelola Dashboard

3. Activity Diagram *View Total Sales*



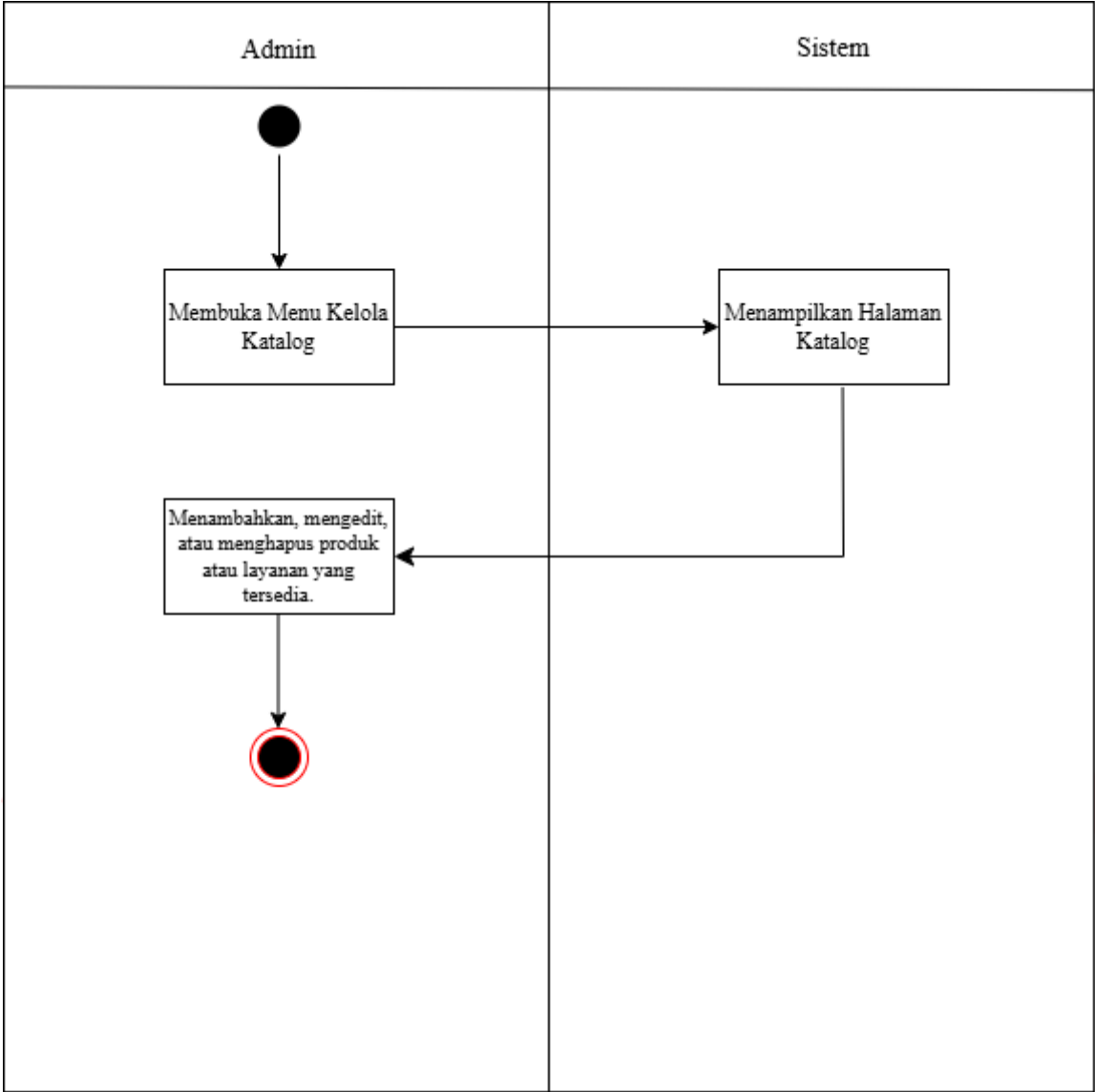
Gambar 3. 11 Activity Diagram *View Total Sales*

#### 4. Activity Diagram Kelola Sales



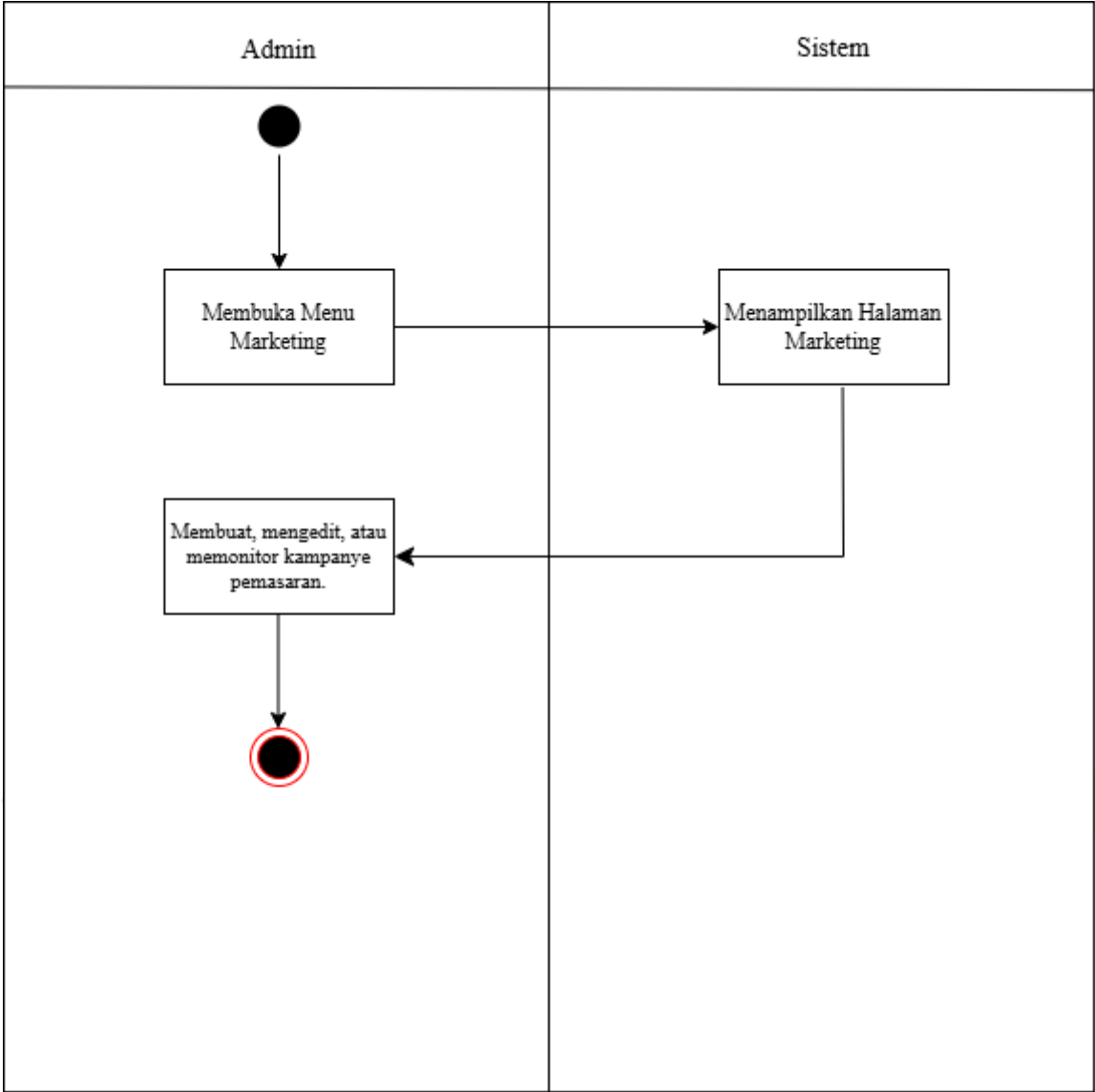
Gambar 3. 12 Activity Diagram Kelola Sales

5. Activity Diagram Kelola Katalog



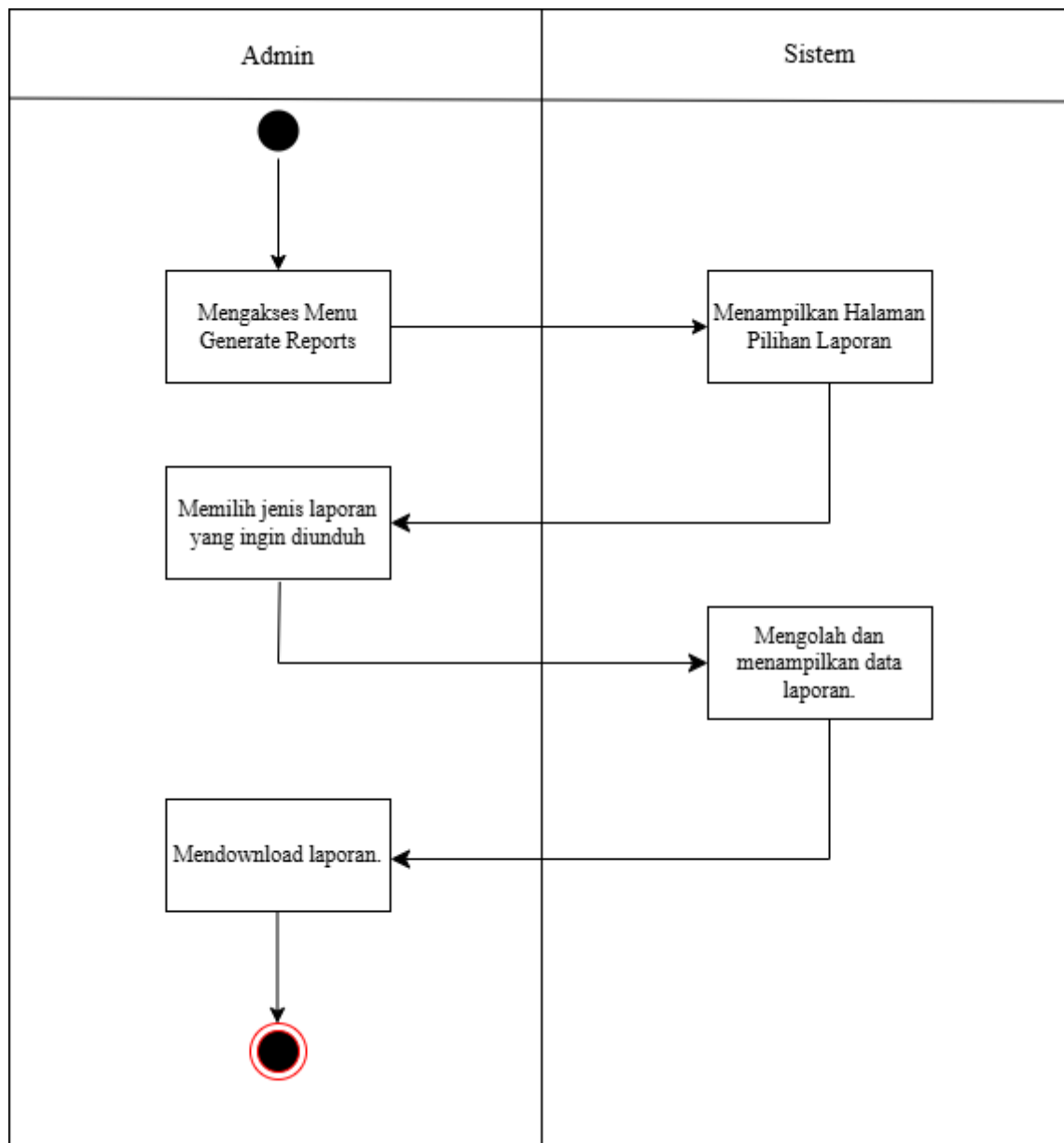
Gambar 3. 13 Activity Diagram Kelola Katalog

6. Activity Diagram Kelola Marketing



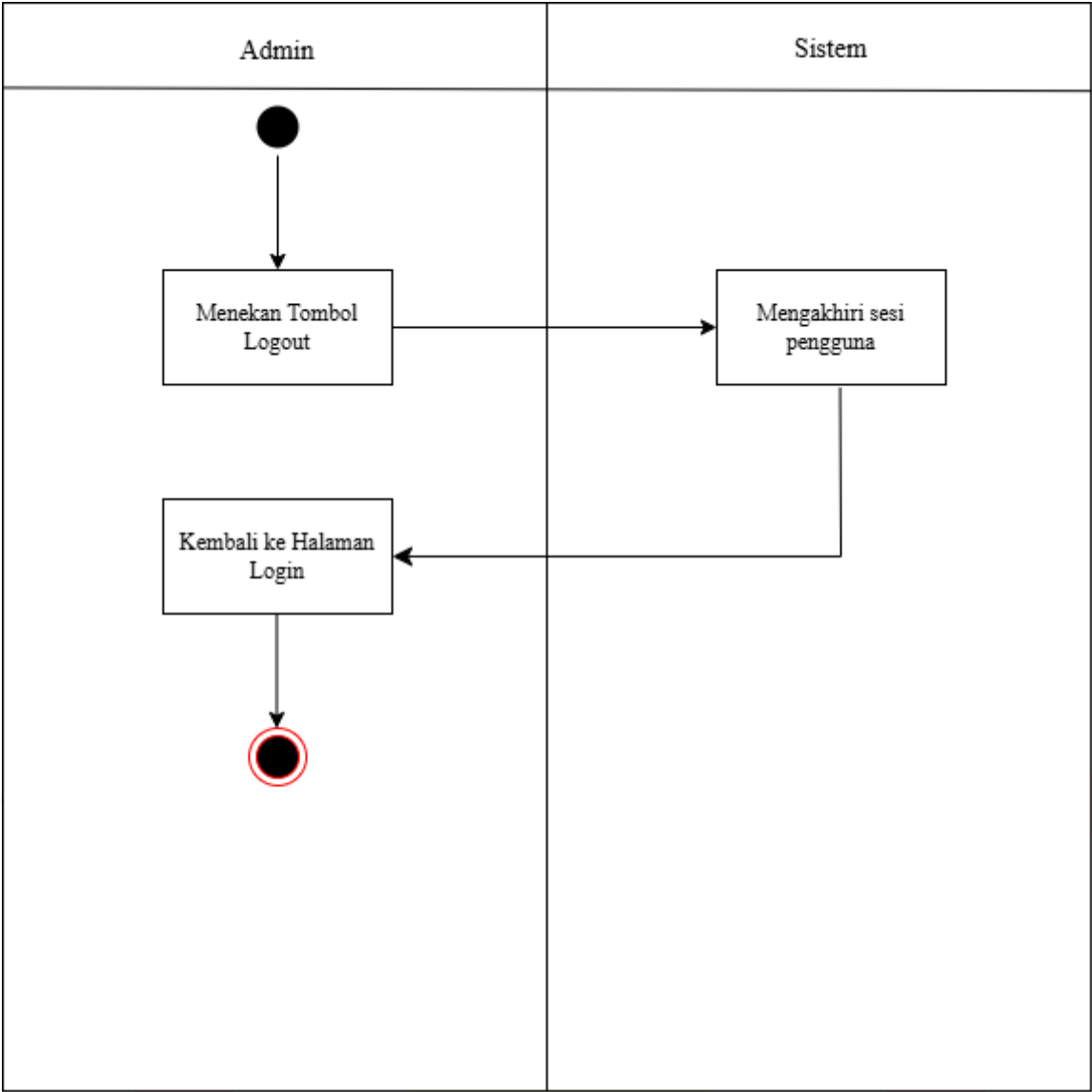
Gambar 3. 14 Activity Diagram Kelola Marketing

7. Activity Diagram Generate Reports



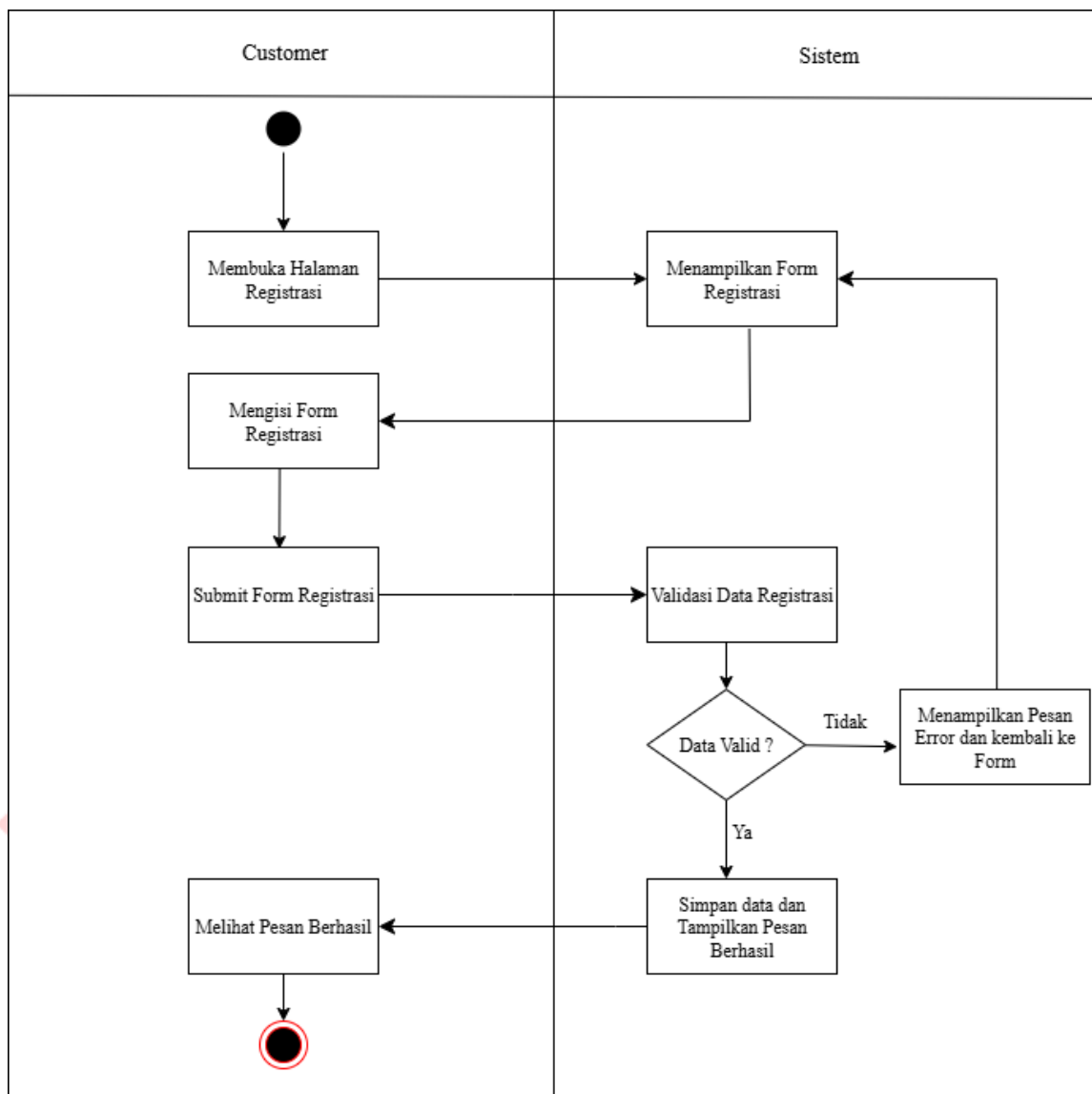
Gambar 3. 15 Activity Diagram Generate Reports

8. Activity Diagram Logout



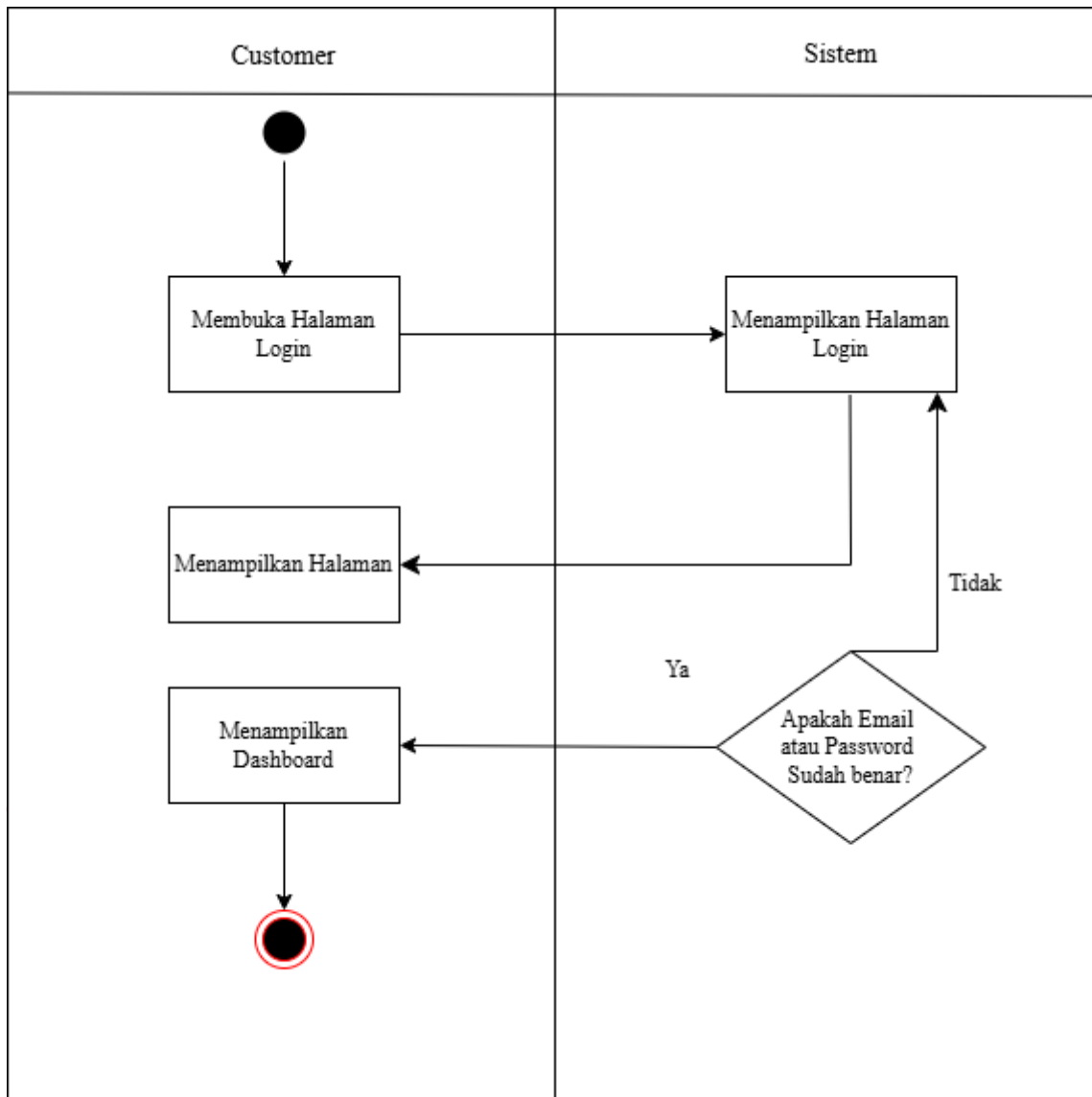
Gambar 3. 16 Activity Diagram Logout

### 9. Activity Diagram Register



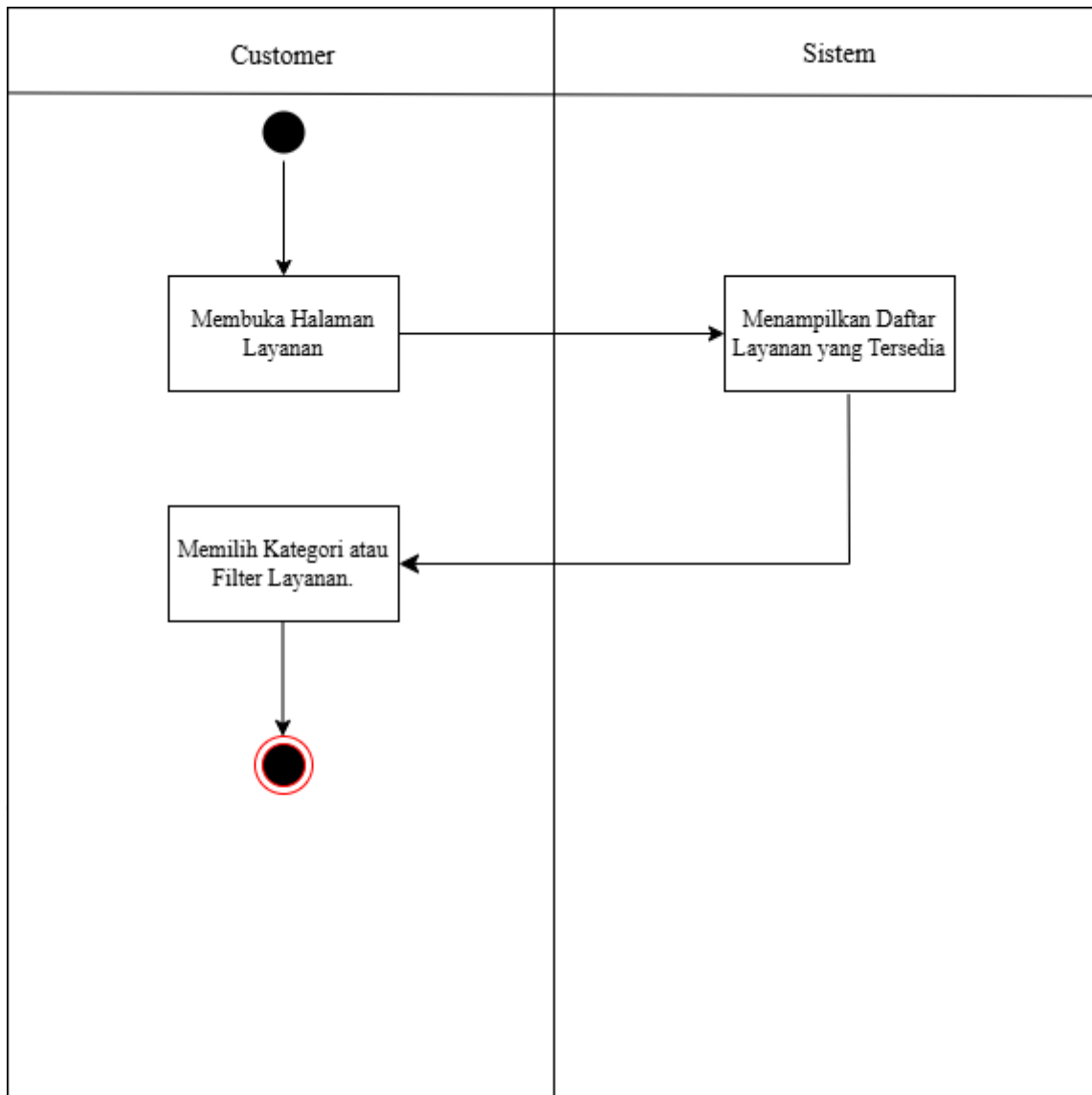
Gambar 3. 17 Activity Diagram Register

10. Activity Diagram Login



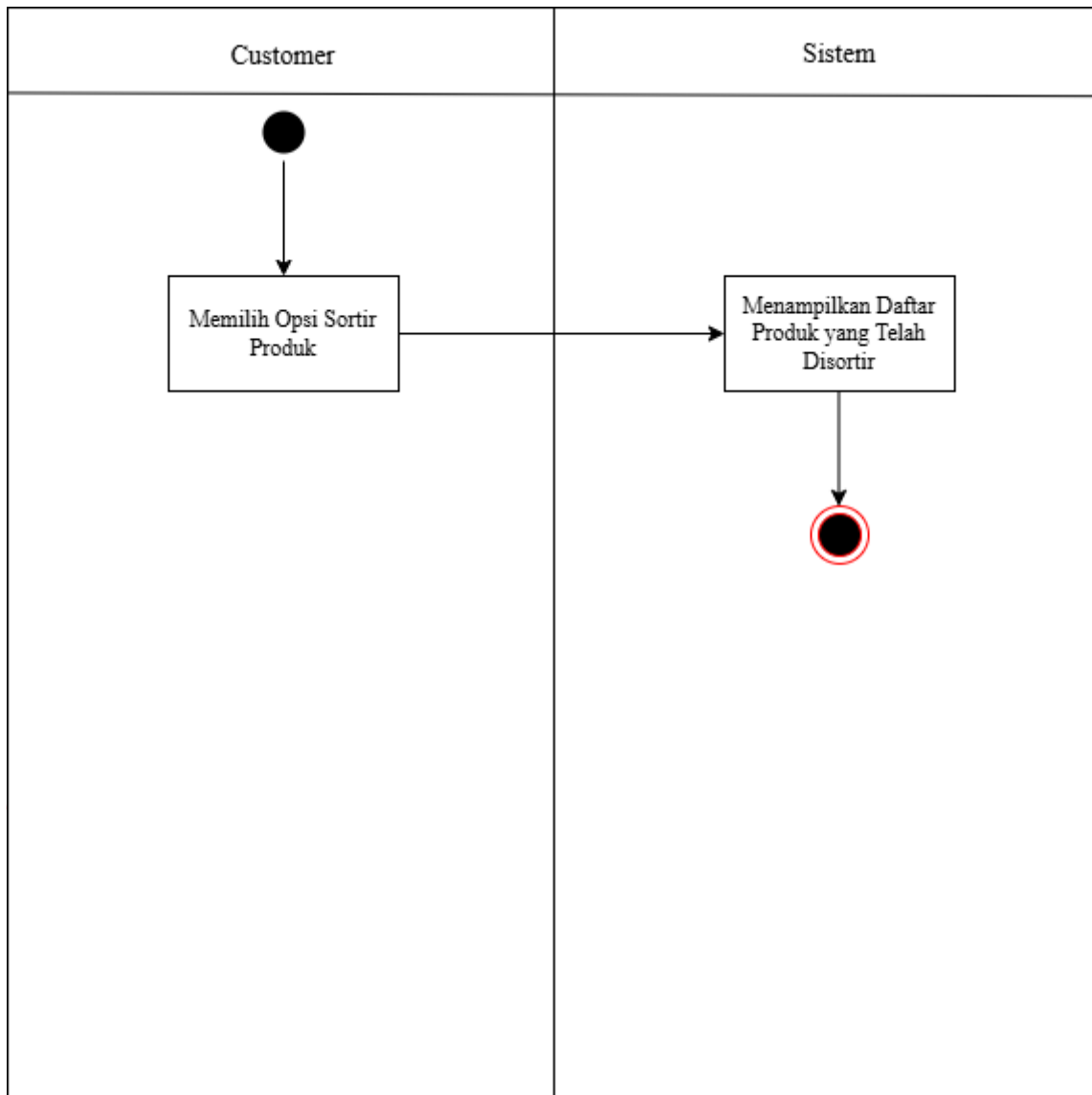
Gambar 3. 18 Activity Diagram Login

11. Activity Diagram Browse Service



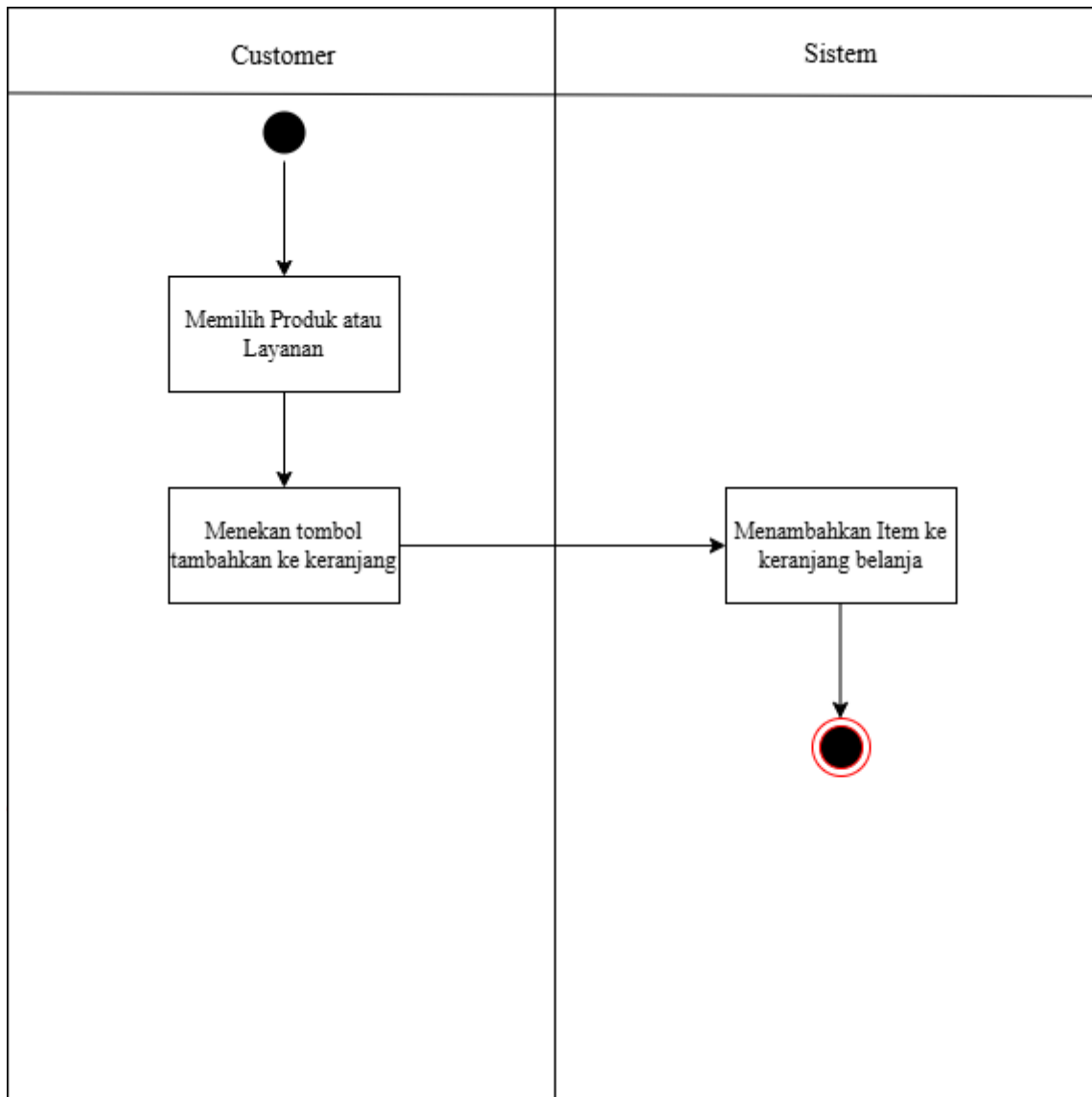
Gambar 3. 19 Activity Diagram Browse Service

12. Activity Diagram Sort Services



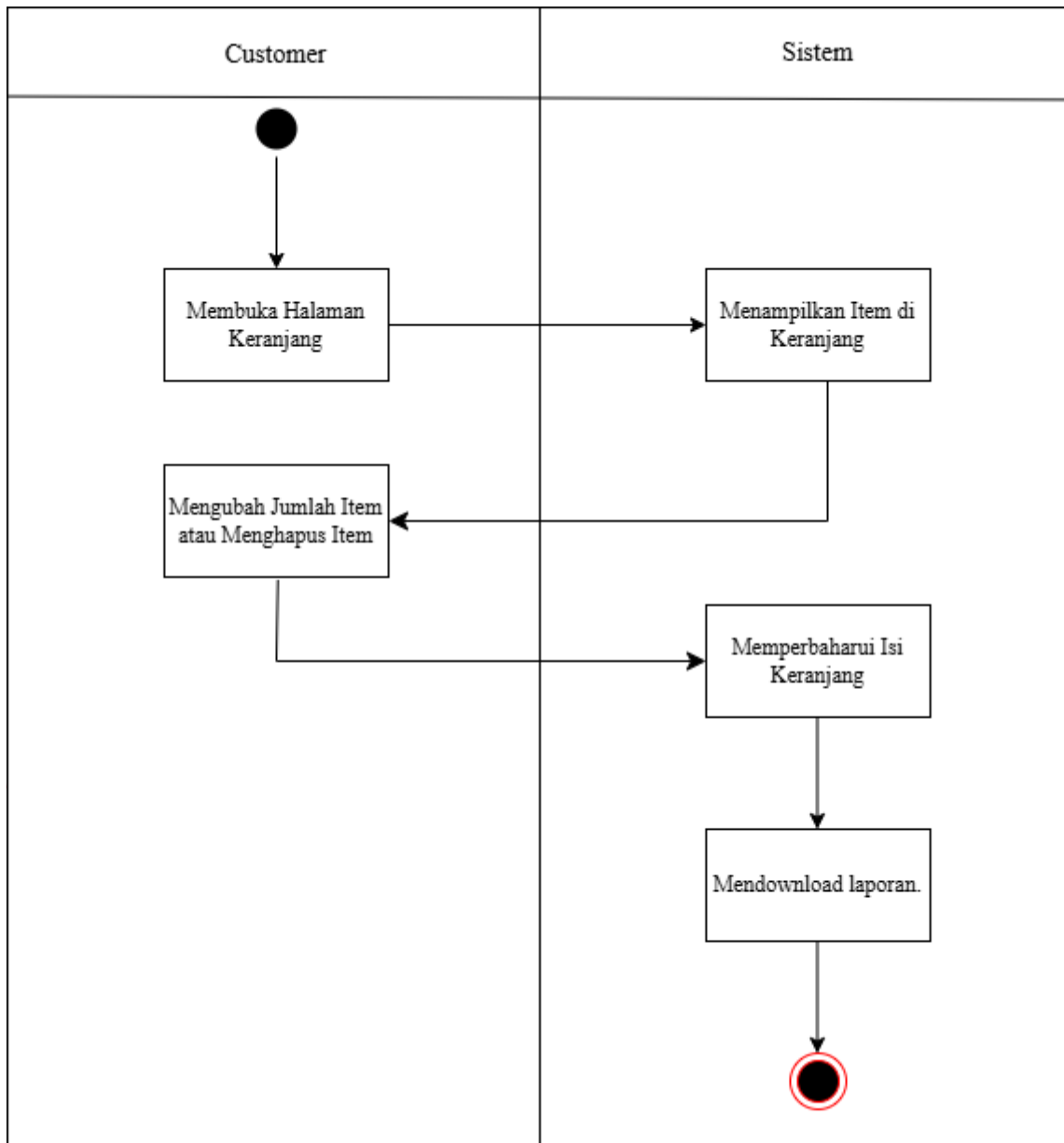
Gambar 3. 20 Activity Diagram Sort Services

13. Activity Diagram Add to Cart



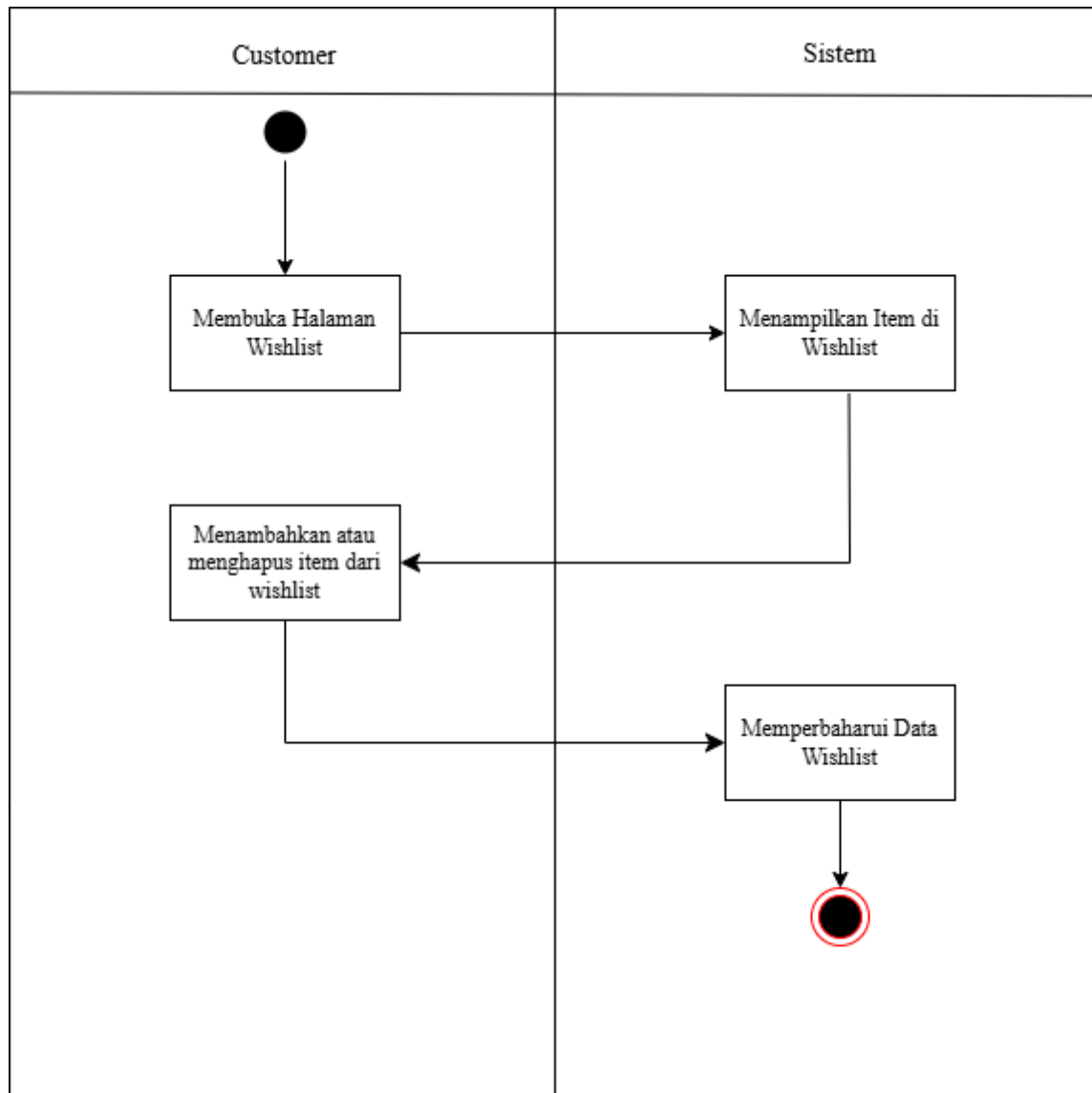
Gambar 3. 21 Activity Diagram Add to Cart

14. Activity Diagram Manage Cart



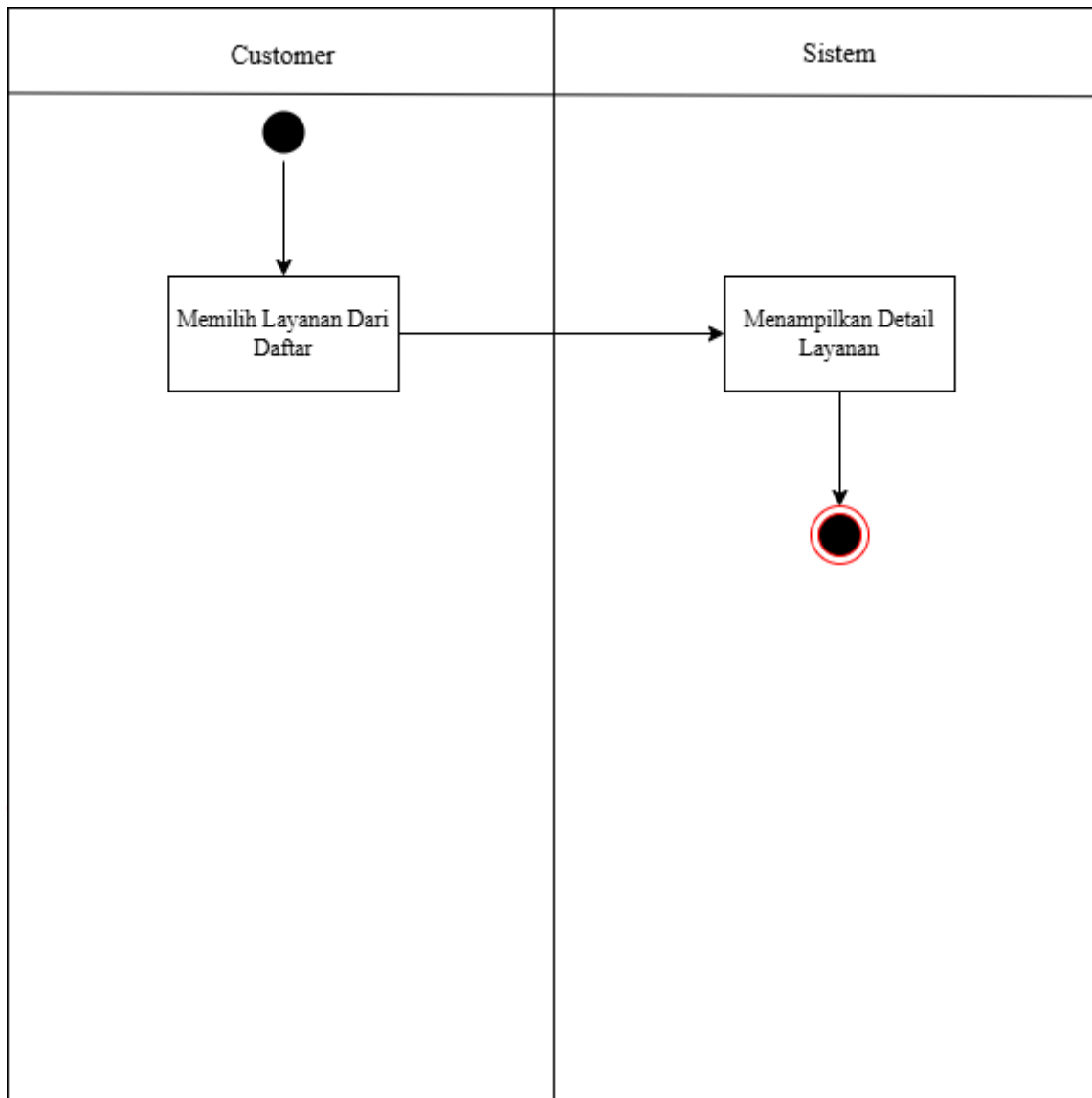
Gambar 3. 22 Activity Diagram Manage Cart

15. Activity Diagram Manage Wishlist



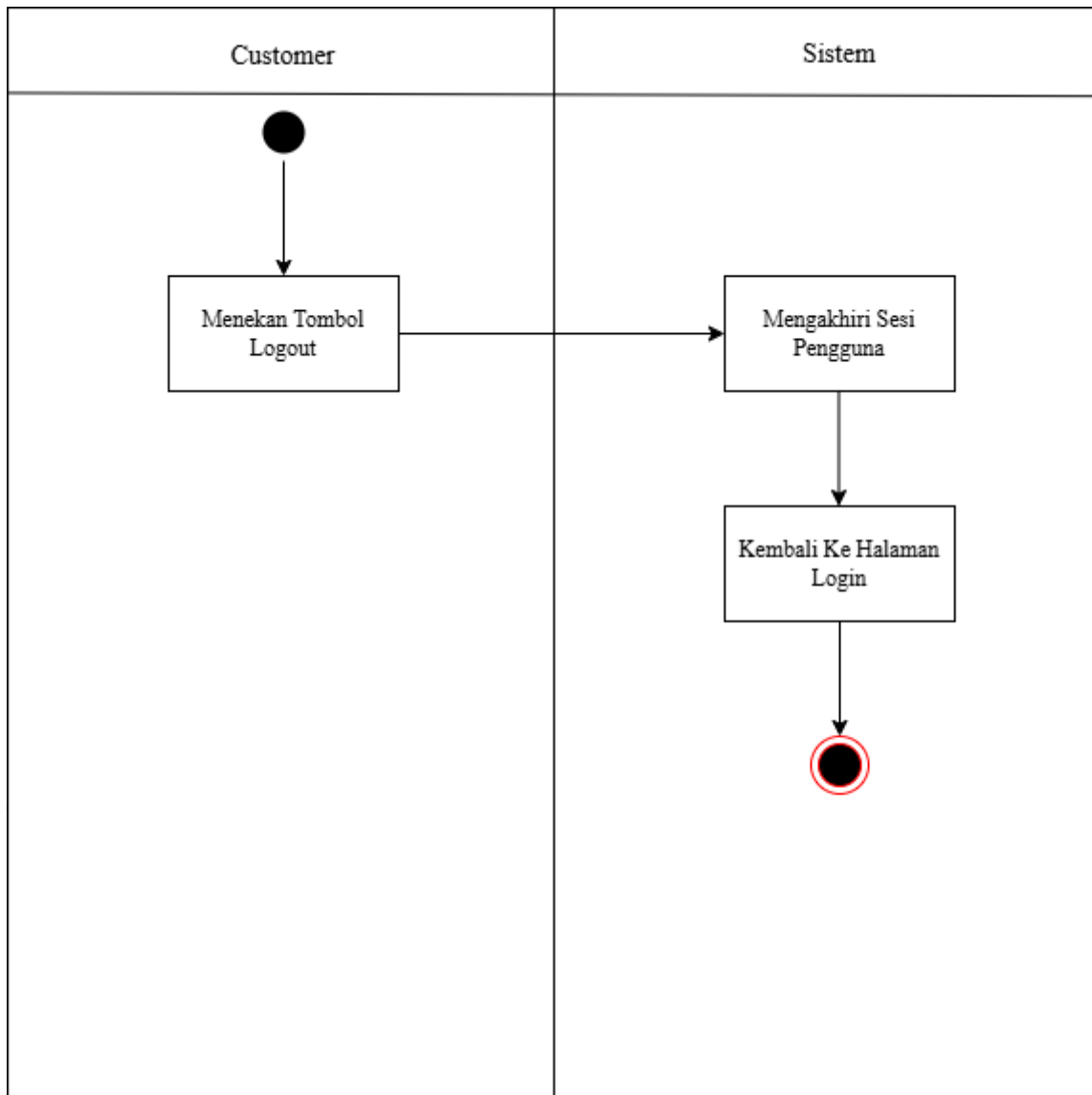
Gambar 3. 23 Activity Diagram Manage Wishlist

16. Activity Diagram *View Service Details*



Gambar 3. 24 *Activity Diagram View Service Details*

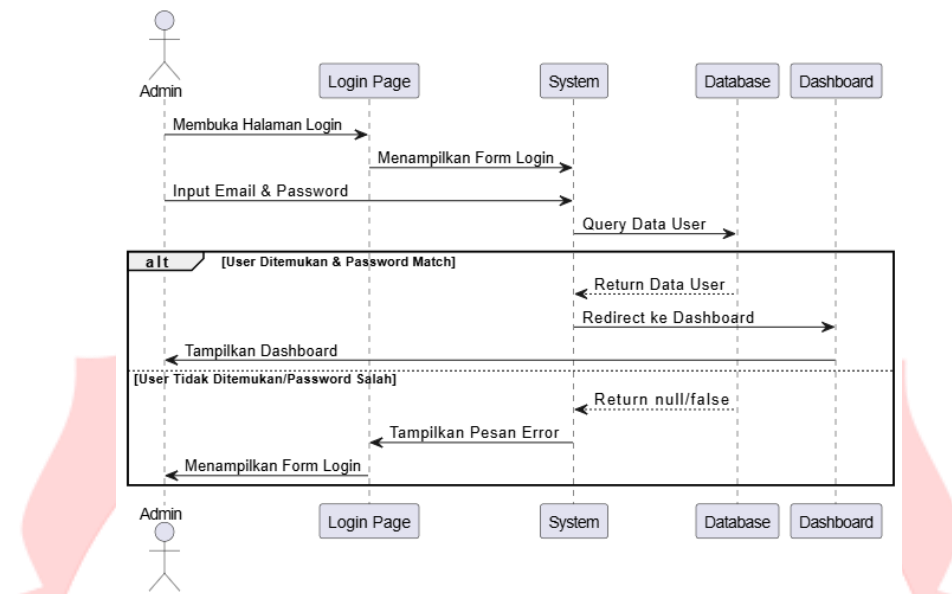
17. Activity Diagram Logout



Gambar 3. 25 Activity Diagram Logout

### 3.8.4 Sequence Diagram

#### 1. Sequence Diagram Login

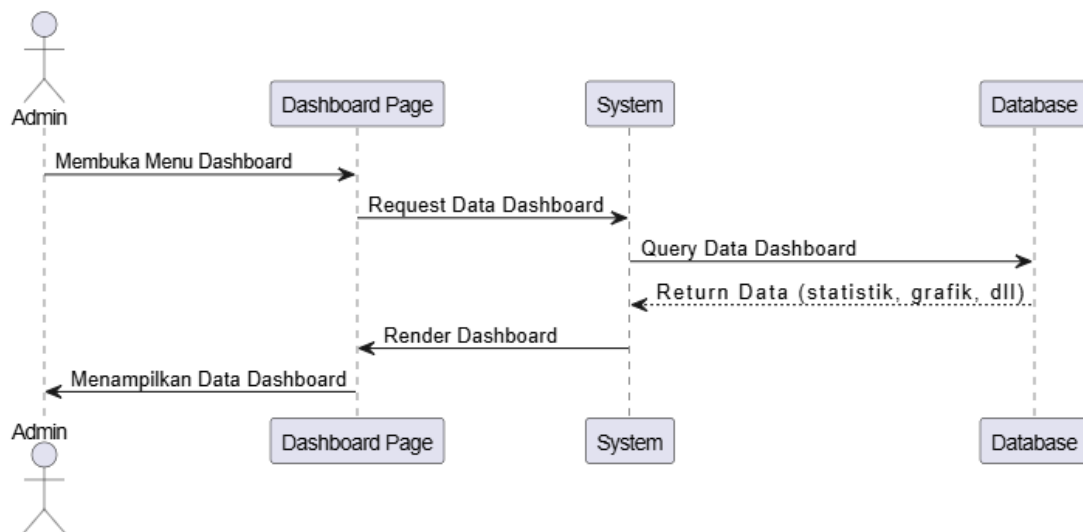


Gambar 3. 26 Sequence Diagram Login

Berikut penjelasan dari Sequence Diagram Login

- 1 (satu) aktor yang melakukan kegiatan yaitu *Admin*.
- 4 (empat) lifeline yang menjelaskan komponen sistem yaitu Login Page (halaman login), *System* (sistem), *Database* (basis data), dan Dashboard (halaman utama).
- 1 (satu) fragment alt yang menjelaskan kondisi alternatif untuk validasi login.
- 8 (delapan) message yang menjelaskan alur sistem yaitu Membuka Halaman Login, Menampilkan Form Login, Input Email & Password, Query Data User, Return Data User, Redirect ke Dashboard, Tampilkan Pesan Error, dan Menampilkan Form Login (kembali).
- 2 (dua) return message yang menjelaskan balasan dari sistem yaitu Return Data User (jika login berhasil) dan Return null/false (jika login gagal).
- 1 (satu) conditional flow dengan dua kemungkinan yaitu [User Ditemukan & Password Match] yang akan redirect ke Dashboard dan [User Tidak Ditemukan/Password Salah] yang akan tampilkan pesan error dan kembali ke form login.

## 2. Sequence Diagram Kelola Dashboard

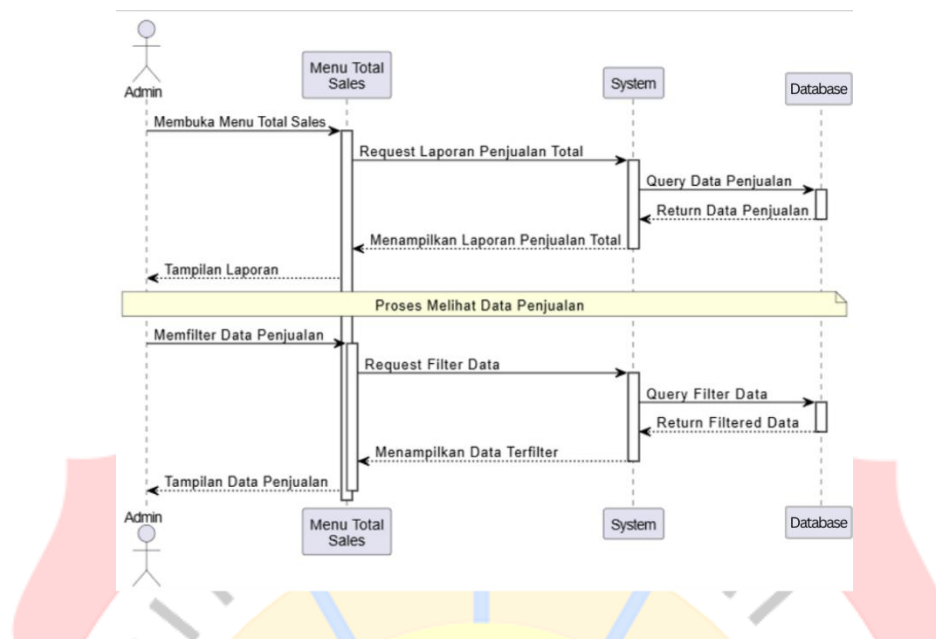


Gambar 3. 27 Sequence Diagram Kelola Dashboard

### Berikut penjelasan dari *Sequence Diagram Kelola Dashboard*:

- 1 (satu) aktor yang melakukan kegiatan yaitu *Admin*.
- 3 (tiga) *lifeline* yang menjelaskan komponen sistem yaitu *Dashboard Page* (halaman dashboard), *System* (sistem), dan *Database* (basis data).
- Tidak ada *fragment* khusus yang digunakan.
- 5 (lima) *message* yang menjelaskan alur sistem yaitu *Membuka Menu Dashboard*, *Request Data Dashboard*, *Query Data Dashboard*, *Render Dashboard*, dan *Menampilkan Data Dashboard*.
- 1 (satu) *return message* yaitu *Return Data (statistik, grafik, dll)* yang mengirimkan data dashboard dari database.
- Tidak ada *conditional flow* dalam diagram ini.

### 3. Sequence Diagram View Total Sales

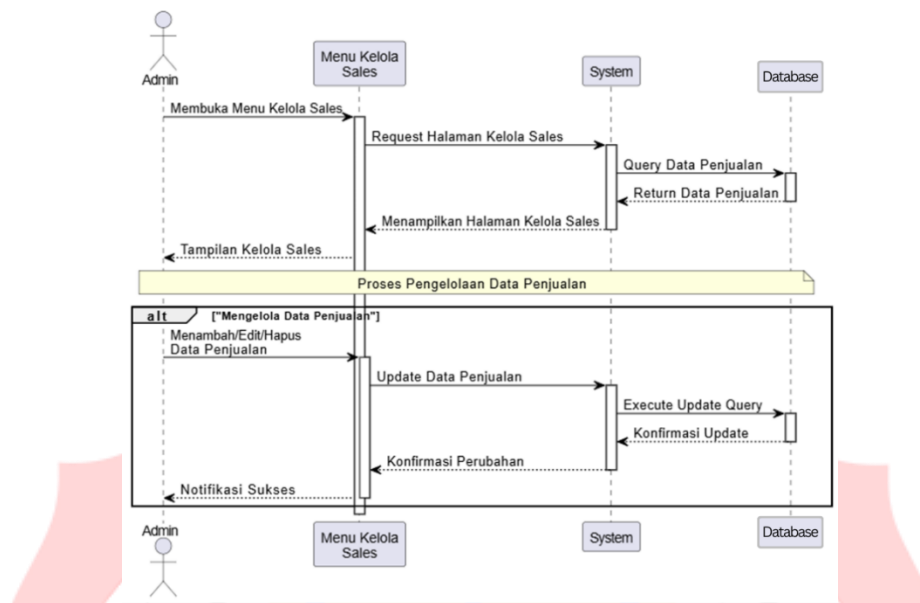


Gambar 3. 28 Sequence Diagram View Total Sales

Berikut penjelasan dari Sequence Diagram View Total Sales:

- 1 (satu) aktor yang melakukan kegiatan yaitu *Admin*.
- 3 (tiga) lifeline yang menjelaskan komponen sistem yaitu Menu Total Sales, *System* (sistem), dan *Database* (basis data).
- 1 (satu) fragment yaitu Proses Melihat Data Penjualan yang menjelaskan alur pemrosesan dan filtering data.
- 8 (delapan) message yang menjelaskan alur sistem yaitu Membuka Menu Total Sales, Request Laporan Penjualan Total, Query Data Penjualan, Menampilkan Laporan Penjualan Total, Memfilter Data Penjualan, Request Filter Data, Query Filter Data, dan Menampilkan Data Terfilter.
- 3 (tiga) return message yaitu Return Data Penjualan, Return Filtered Data, dan Tampilkan Laporan serta Tampilan Data Penjualan.
- Tidak ada conditional flow tetapi ada proses filtering data penjualan berdasarkan kriteria tertentu.

#### 4. Sequence Diagram Kelola Sales

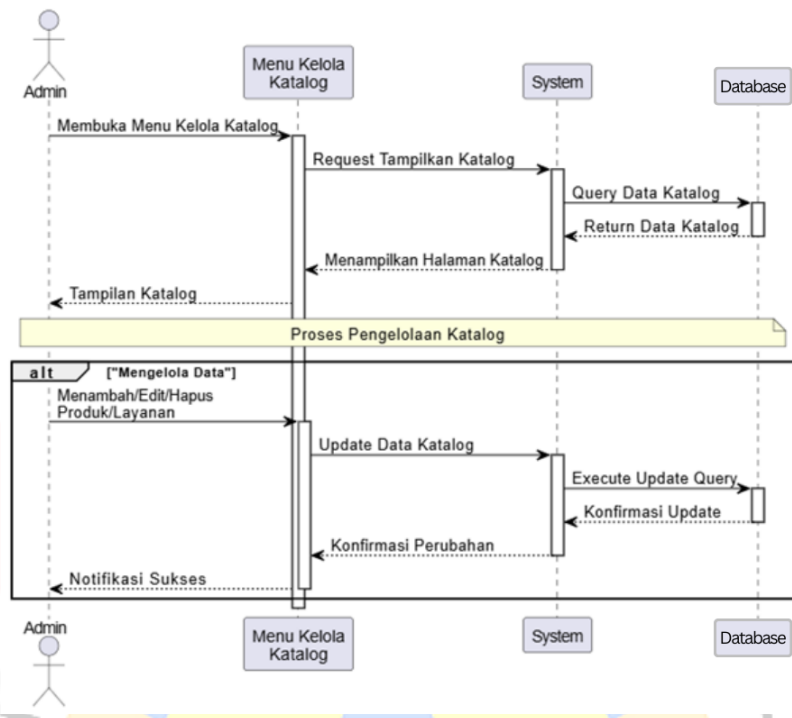


Gambar 3. 29 Sequence Diagram Kelola Sales

Berikut penjelasan dari *Sequence Diagram Kelola Sales*:

- 1 (satu) aktor yang melakukan kegiatan yaitu *Admin*.
- 3 (tiga) *lifeline* yang menjelaskan komponen sistem yaitu *Menu Kelola Sales*, *System* (sistem), dan *Database* (basis data).
- 2 (dua) *fragment* yaitu **Proses Pengelolaan Data Penjualan** dan *alt* dengan kondisi **["Mengelola Data Penjualan"]** untuk operasi *CRUD* data.
- 9 (sembilan) *message* yang menjelaskan alur sistem yaitu *Membuka Menu Kelola Sales*, *Request Halaman Kelola Sales*, *Query Data Penjualan*, *Menampilkan Halaman Kelola Sales*, *Menambah/Edit/Hapus Data Penjualan*, *Update Data Penjualan*, *Execute Update Query*, *Konfirmasi Perubahan*, dan *Notifikasi Sukses*.
- 3 (tiga) *return message* yaitu *Return Data Penjualan*, *Konfirmasi Update*, dan *Tampilan Kelola Sales*.
- 1 (satu) *conditional flow* dengan kondisi **["Mengelola Data Penjualan"]** yang memungkinkan operasi tambah, *edit*, atau hapus data penjualan.

## 5. Sequence Diagram Kelola Katalog



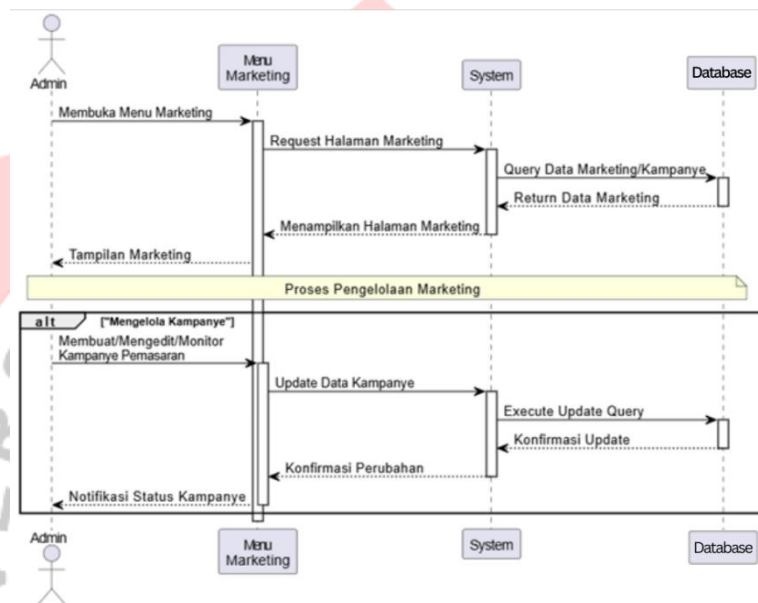
Gambar 3. 30 Sequence Diagram Kelola Katalog

Berikut penjelasan dari *Sequence Diagram Kelola Katalog*:

- 1 (satu) aktor yang melakukan kegiatan yaitu *Admin*.
- 3 (tiga) *lifeline* yang menjelaskan komponen sistem yaitu *Menu Kelola Katalog*, *System* (sistem), dan *Database* (basis data).
- 2 (dua) *fragment* yaitu **Proses Pengelolaan Katalog** dan *alt* dengan kondisi ["**Mengelola Data**"] untuk operasi *CRUD* katalog.
- 10 (sepuluh) *message* yang menjelaskan alur sistem yaitu *Membuka Menu Kelola Katalog*, *Request Tampilkan Katalog*, *Query Data Katalog*, *Return Data Katalog*, *Menampilkan Halaman Katalog*, *Tampilan Katalog*, *Menambah/Edit/Hapus Produk/Layanan*, *Update Data Katalog*, *Execute Update Query*, dan *Konfirmasi Update*.
- 3 (tiga) *return message* yaitu *Return Data Katalog*, *Konfirmasi Update*, dan *Notifikasi Sukses*.

- f. 1 (satu) *conditional flow* dengan kondisi ["**Mengelola Data**"] yang memungkinkan operasi tambah, *edit*, atau hapus produk/layanan dalam katalog.

## 6. Sequence Diagram Kelola Marketing



Gambar 3. 31 Sequence Diagram Kelola Marketing

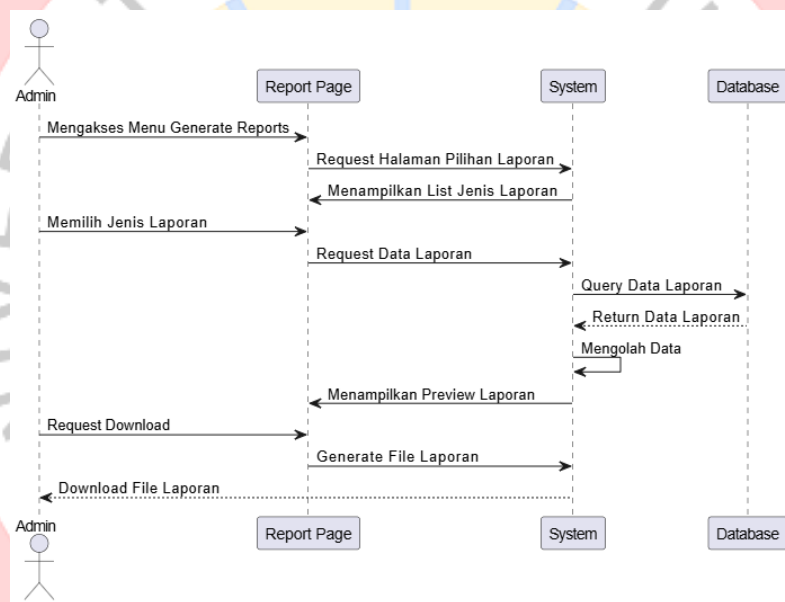
### Berikut penjelasan dari Sequence Diagram Kelola Marketing:

- 1 (satu) aktor yang melakukan kegiatan yaitu *Admin*.
- 3 (tiga) *lifeline* yang menjelaskan komponen sistem yaitu *Menu Marketing*, *System* (sistem), dan *Database* (basis data).
- 2 (dua) *fragment* yaitu **Proses Pengelolaan Marketing** dan *alt* dengan kondisi ["**Mengelola Kampanye**"] untuk operasi manajemen kampanye.
- 9 (sembilan) *message* yang menjelaskan alur sistem yaitu *Membuka Menu Marketing*, *Request Halaman Marketing*, *Query Data Marketing/Kampanye*, *Menampilkan Halaman Marketing*, *Membuat/Mengedit/Monitor Kampanye Pemasaran*, *Update Data Marketing*, *Konfirmasi Perubahan*, *Notifikasi Status Kampanye*, *Execute Update Query*, dan *Konfirmasi Update*.

Kampanye, *Execute Update Query*, Konfirmasi Perubahan, dan Notifikasi Status Kampanye.

- e. 3 (tiga) *return message* yaitu *Return Data Marketing*, Konfirmasi *Update*, dan Tampilan *Marketing*.
- f. 1 (satu) *conditional flow* dengan kondisi ["**Mengelola Kampanye**"] yang memungkinkan pembuatan, pengeditan, atau *monitoring* kampanye pemasaran.

### 7. Sequence Diagram Generate Reports



Gambar 3. 32 Sequence Diagram Generate Reports

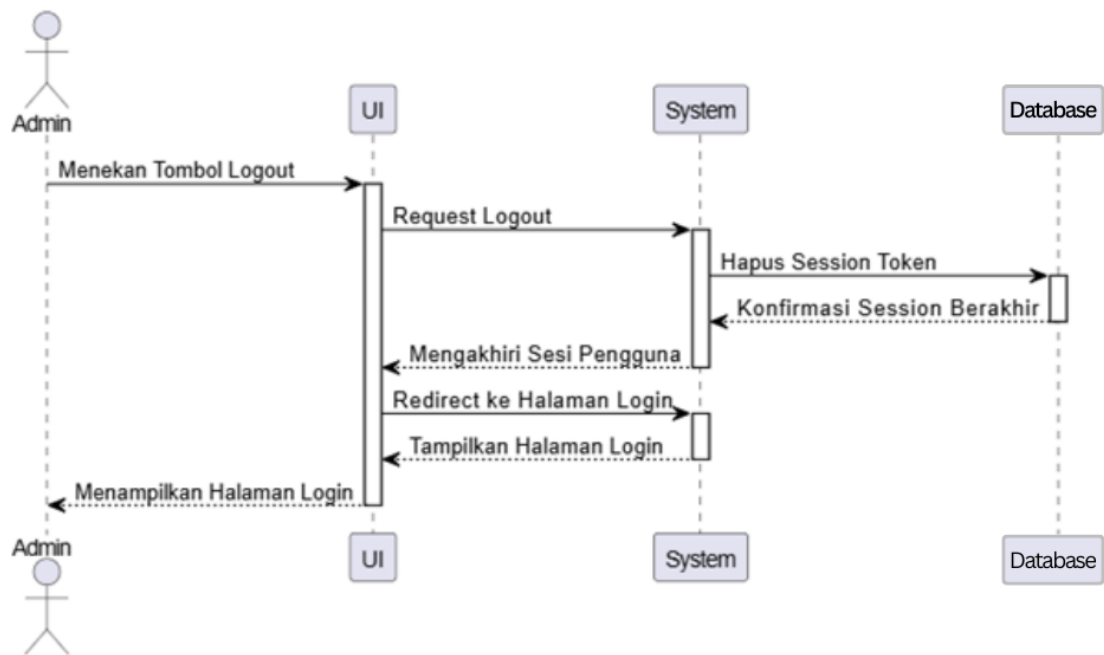
Berikut penjelasan dari *Sequence Diagram Generate Reports*:

- a. 1 (satu) aktor yang melakukan kegiatan yaitu *Admin*.
- b. 3 (tiga) *lifeline* yang menjelaskan komponen sistem yaitu *Report Page* (halaman laporan), *System* (sistem), dan *Database* (basis data).
- c. Tidak ada *fragment* khusus yang digunakan.

- d. 10 (sepuluh) *message* yang menjelaskan alur sistem yaitu Mengakses *Menu Generate Reports, Request* Halaman Pilihan Laporan, Menampilkan *List* Jenis Laporan, Memilih Jenis Laporan, *Request Data* Laporan, *Query Data* Laporan, Mengolah Data, Menampilkan *Preview* Laporan, *Request Download*, dan *Generate File* Laporan.
- e. 2 (dua) *return message* yaitu *Return Data* Laporan dan *Download File* Laporan.
- f. Tidak ada *conditional flow* dalam diagram ini.



## 8. Sequence Diagram Logout

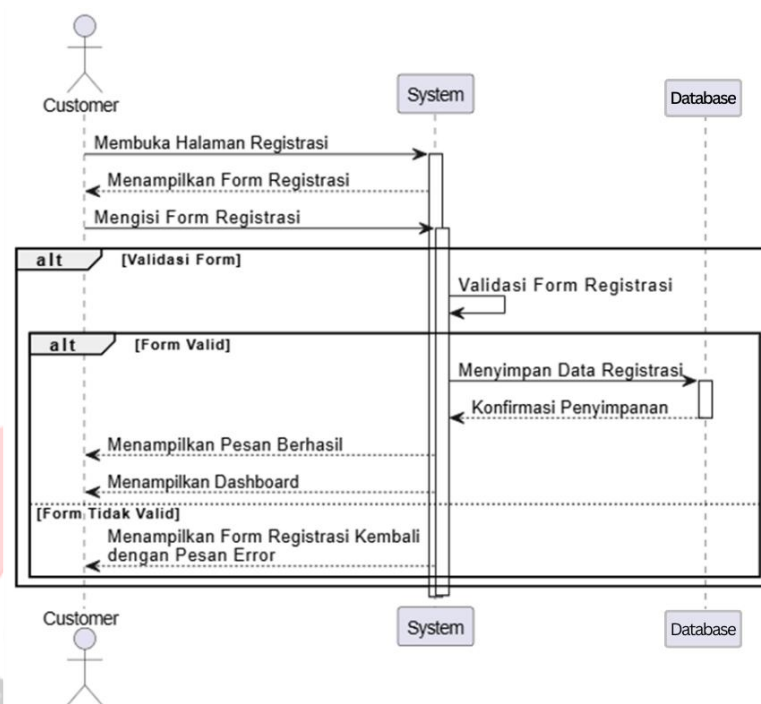


Gambar 3. 33 Sequence Diagram Logout

Berikut penjelasan dari *Sequence Diagram Logout*:

- 1 (satu) aktor yang melakukan kegiatan yaitu *Admin*.
- 3 (tiga) *lifeline* yang menjelaskan komponen sistem yaitu *UI* (antarmuka pengguna), *System* (sistem), dan *Database* (basis data).
- Tidak ada *fragment* khusus yang digunakan.
- 6 (enam) *message* yang menjelaskan alur sistem yaitu *Menekan Tombol Logout*, *Request Logout*, *Hapus Session Token*, *Mengakhiri Sesi Pengguna*, *Redirect ke Halaman Login*, dan *Menampilkan Halaman Login*.
- 2 (dua) *return message* yaitu *Konfirmasi Session Berakhir* dan *Tampilkan Halaman Login*.
- Tidak ada *conditional flow* dalam diagram ini.

## 9. Sequence Diagram Register



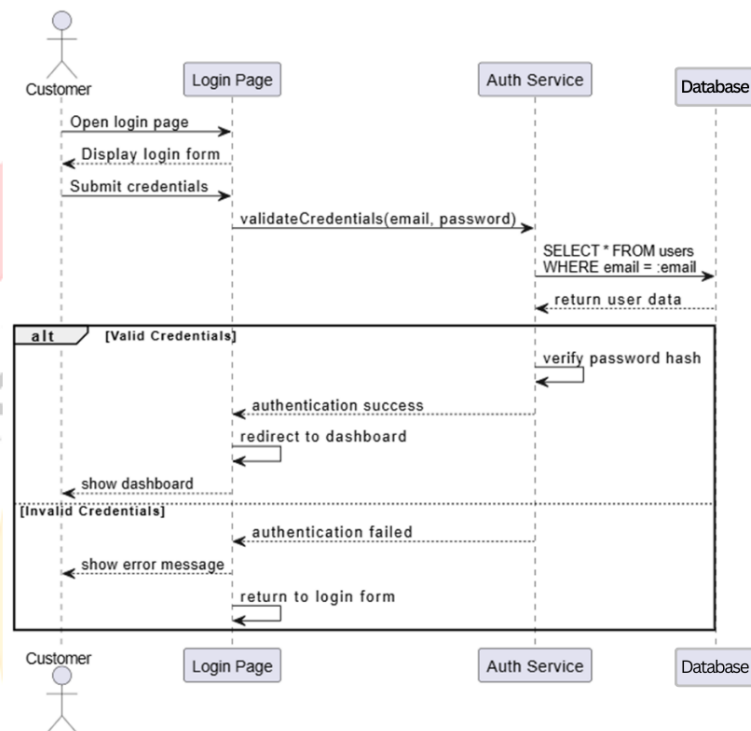
Gambar 3. 34 Sequence Diagram Register

Berikut penjelasan dari *Sequence Diagram Register*:

- 1 (satu) aktor yang melakukan kegiatan yaitu *Customer*.
- 2 (dua) *lifeline* yang menjelaskan komponen sistem yaitu *System* (sistem) dan *Database* (basis data).
- 2 (dua) *fragment alt* yaitu [**Validasi Form**] untuk validasi *input* dan [**Form Valid**] untuk proses penyimpanan data registrasi.
- 8 (delapan) *message* yang menjelaskan alur sistem yaitu Membuka Halaman Registrasi, Menampilkan *Form* Registrasi, Mengisi *Form* Registrasi, Validasi *Form* Registrasi, Menyimpan Data Registrasi, Menampilkan Pesan Berhasil, Menampilkan *Dashboard*, dan Menampilkan *Form* Registrasi Kembali dengan Pesan *Error*.
- 2 (dua) *return message* yaitu Konfirmasi Penyimpanan dan *return* data untuk proses validasi.

- f. 2 (dua) *conditional flow* yaitu [**Validasi Form**] yang mengecek validitas *form* dan [**Form Valid**] untuk menyimpan data jika valid atau [**Form Tidak Valid**] untuk menampilkan pesan *error*.

### 10. Sequence Diagram Login



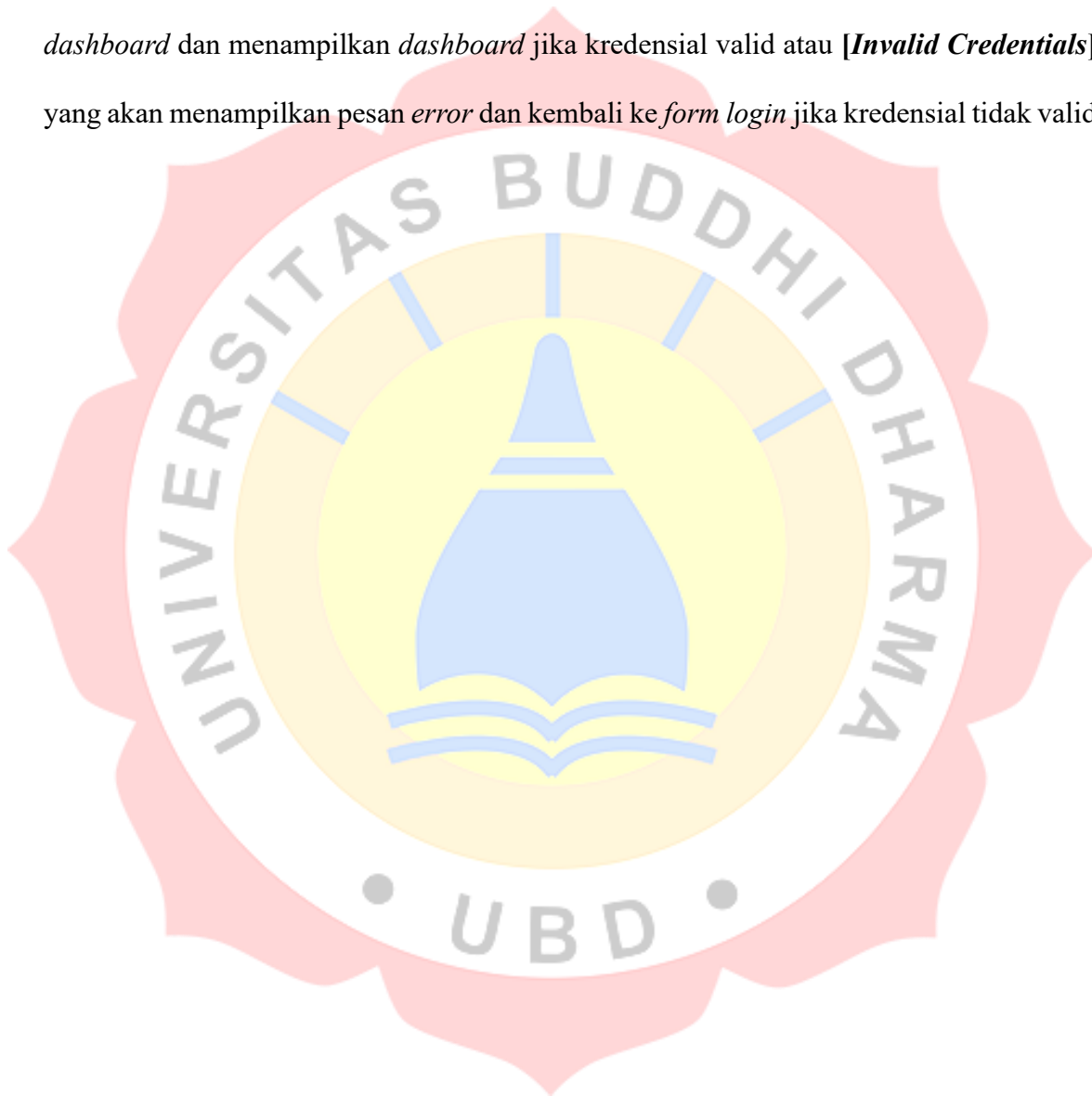
Gambar 3. 35 Sequence Diagram Login

Berikut penjelasan dari *Sequence Diagram Customer Login*:

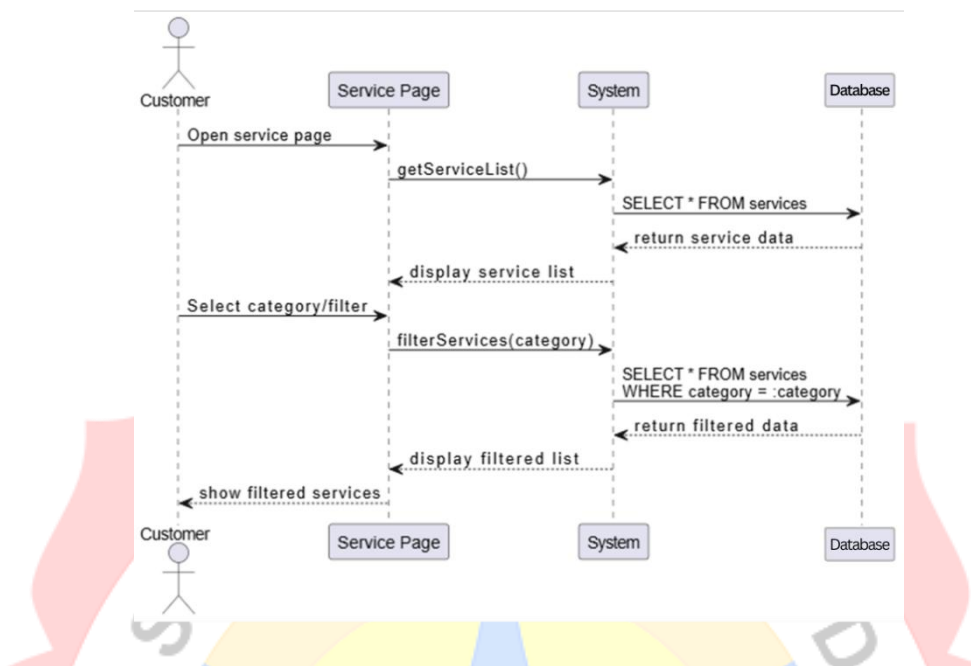
- 1 (satu) aktor yang melakukan kegiatan yaitu *Customer*.
- 3 (tiga) *lifeline* yang menjelaskan komponen sistem yaitu *Login Page* (halaman *login*), *Auth Service* (layanan autentikasi), dan *Database* (basis data).
- 1 (satu) *fragment alt* yaitu [**Valid Credentials**] untuk menangani kondisi kredensial valid atau tidak valid.
- 9 (sembilan) *message* yang menjelaskan alur sistem yaitu *Open login page*, *Display login form*, *Submit credentials*, *validateCredentials(email, password)*, *SELECT \* FROM users*

*WHERE email = email, verify password hash, authentication success, authentication failed, dan show error message.*

- e. 4 (empat) *return message* yaitu *return user data, redirect to dashboard, show dashboard, dan return to login form.*
- f. 1 (satu) *conditional flow* dengan kondisi [**Valid Credentials**] yang akan *redirect* ke *dashboard* dan menampilkan *dashboard* jika kredensial valid atau [**Invalid Credentials**] yang akan menampilkan pesan *error* dan kembali ke *form login* jika kredensial tidak valid



## 11. Sequence Diagram Browse Service

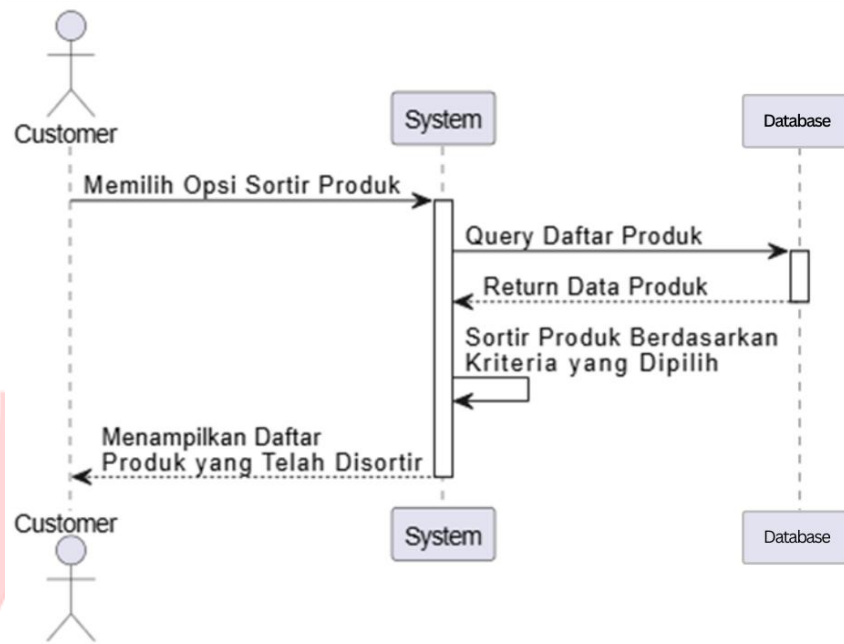


Gambar 3. 36 Sequence Diagram Browse Service

Berikut penjelasan dari *Sequence Diagram Browse Service*:

- 1 (satu) aktor yang melakukan kegiatan yaitu *Customer*.
- 3 (tiga) *lifeline* yang menjelaskan komponen sistem yaitu *Service Page* (halaman layanan), *System* (sistem), dan *Database* (basis data).
- Tidak ada *fragment* khusus yang digunakan.
- 8 (delapan) *message* yang menjelaskan alur sistem yaitu *Open service page*, *getServiceList()*, *SELECT \* FROM services*, *return service data*, *display service list*, *Select category/filter*, *filterServices(category)*, dan *SELECT \* FROM services WHERE category = category*.
- 3 (tiga) *return message* yaitu *return service data*, *return filtered data*, dan *show filtered services*.
- Tidak ada *conditional flow* tetapi ada proses *filtering* layanan berdasarkan kategori yang dipilih.

## 12. Sequence Diagram Sort Services

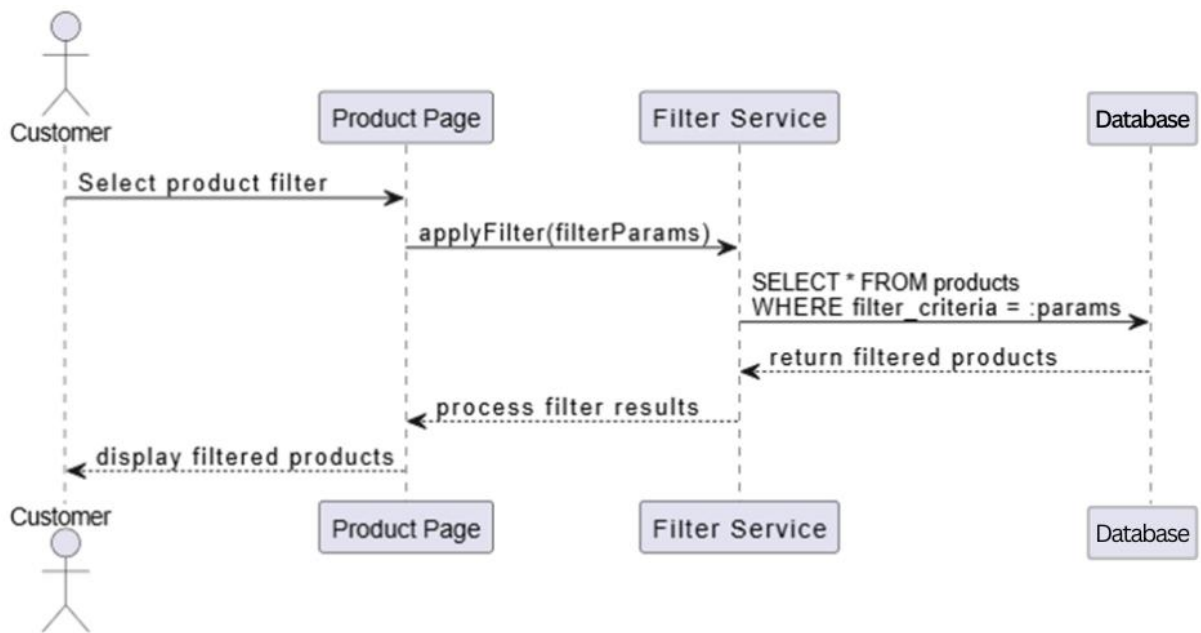


Gambar 3. 37 Sequence Diagram Sort Services

Berikut penjelasan dari *Sequence Diagram Sort Services*:

- 1 (satu) aktor yang melakukan kegiatan yaitu *Customer*.
- 2 (dua) *lifeline* yang menjelaskan komponen sistem yaitu *System* (sistem) dan *Database* (basis data).
- Tidak ada *fragment* khusus yang digunakan.
- 4 (empat) *message* yang menjelaskan alur sistem yaitu *Memilih Opsi Sortir Produk*, *Query Daftar Produk*, *Sortir Produk Berdasarkan Kriteria yang Dipilih*, dan *Menampilkan Daftar Produk yang Telah Disortir*.
- 1 (satu) *return message* yaitu *Return Data Produk* yang sudah diurutkan.
- Tidak ada *conditional flow* dalam diagram ini.

### 13. Sequence Diagram Filter Services

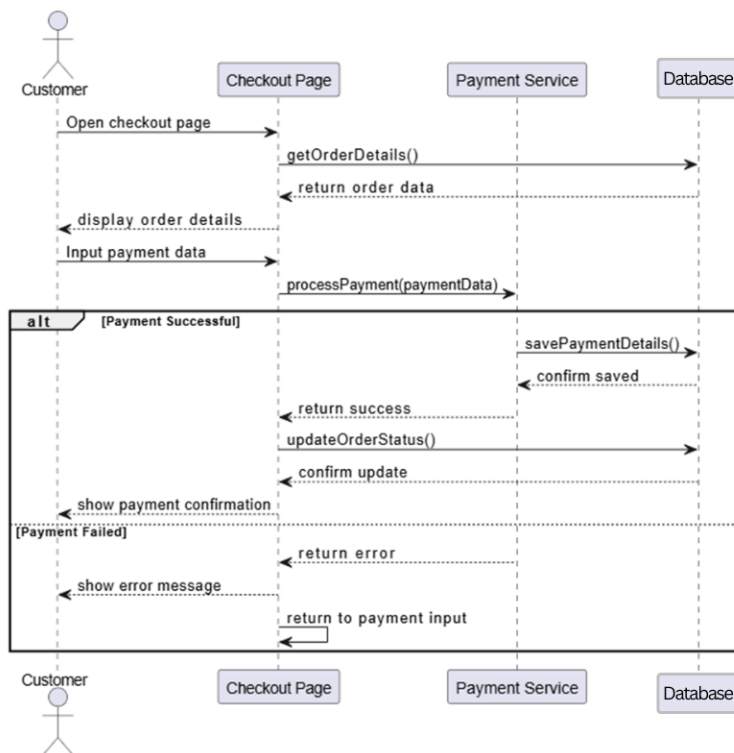


Gambar 3. 38 Sequence Diagram Filter Products

Berikut penjelasan dari *Sequence Diagram Filter Services*:

- 1 (satu) aktor yang melakukan kegiatan yaitu *Customer*.
- 3 (tiga) *lifeline* yang menjelaskan komponen sistem yaitu *Product Page*, *Filter Service*, dan *Database* (basis data).
- Tidak ada *fragment* khusus yang digunakan.
- 6 (enam) *message* yang menjelaskan alur sistem yaitu *Select product filter*, *applyFilter(filterParams)*, *SELECT \* FROM products WHERE filter\_criteria = params*, *return filtered products*, *process filter results*, dan *display filtered products*.
- 1 (satu) *return message* yaitu *return filtered products* dari *database*.
- Tidak ada *conditional flow* dalam diagram ini.

## 14. Sequence Diagram Checkout



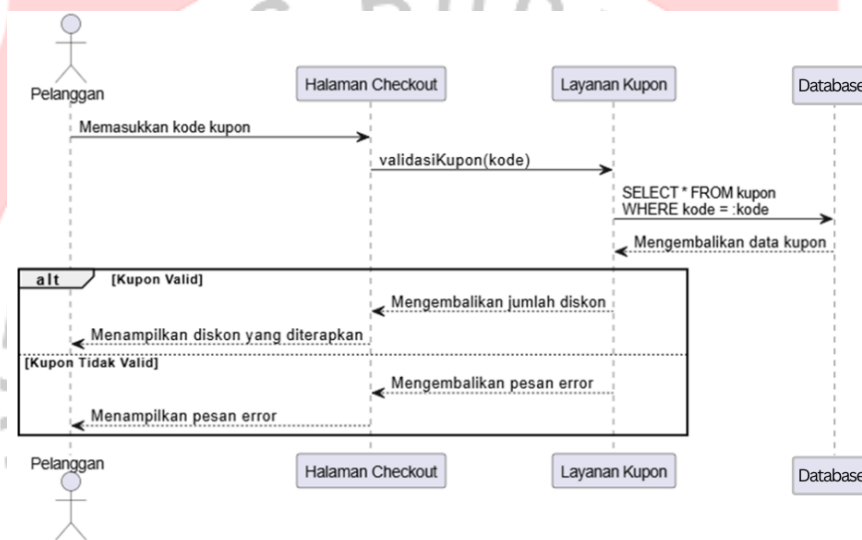
Gambar 3. 39 Sequence Diagram Checkout

Berikut penjelasan dari *Sequence Diagram Checkout*:

- 1 (satu) aktor yang melakukan kegiatan yaitu *Customer*.
- 3 (tiga) *lifeline* yang menjelaskan komponen sistem yaitu *Checkout Page*, *Payment Service*, dan *Database* (basis data).
- 1 (satu) *fragment alt* yaitu **[Payment Successful]** untuk menangani kondisi pembayaran berhasil atau gagal.
- 10 (sepuluh) *message* yang menjelaskan alur sistem yaitu *Open checkout page*, *getOrderDetails()*, *return order data*, *display order details*, *Input payment data*, *processPayment(paymentData)*, *savePaymentDetails()*, *updateOrderStatus()*, *show payment confirmation*, dan *show error message*.

- e. 4 (empat) *return message* yaitu *return order data*, *confirm saved*, *return success*, dan *return error*.
- f. 1 (satu) *conditional flow* dengan kondisi [**Payment Successful**] yang akan menampilkan konfirmasi pembayaran jika berhasil atau [**Payment Failed**] yang akan menampilkan pesan *error* dan kembali ke *input* pembayaran.

15. *Sequence Diagram Apply Coupon*



Gambar 3. 40 *Sequence Diagram Apply Coupon*

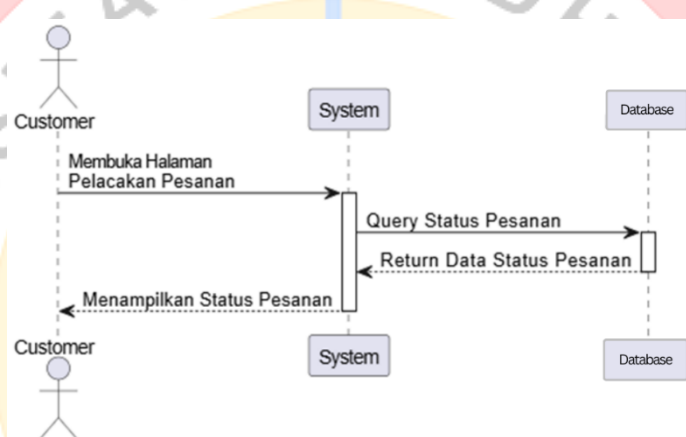
**Berikut penjelasan dari *Sequence Diagram Apply Coupon*:**

- a. 1 (satu) aktor yang melakukan kegiatan yaitu Pelanggan.
- b. 3 (tiga) *lifeline* yang menjelaskan komponen sistem yaitu Halaman *Checkout*, Layanan *Coupon*, dan *Database* (basis data).
- c. 1 (satu) *fragment alt* yaitu [**Coupon Valid**] untuk menangani validasi *coupon*.
- d. 7 (tujuh) *message* yang menjelaskan alur sistem yaitu Memasukkan kode *coupon*, *validasiKupon(kode)*, *SELECT \* FROM kupon WHERE kode = kode*, Mengembalikan

data *coupon*, Mengembalikan jumlah diskon, Menampilkan diskon yang diterapkan, dan Mengembalikan pesan *error*.

- e. 2 (dua) *return message* yaitu Mengembalikan data *coupon* dan Menampilkan pesan *error*.
- f. 1 (satu) *conditional flow* dengan kondisi [**Coupon Valid**] yang akan menampilkan diskon yang diterapkan jika *coupon* valid atau [**Coupon Tidak Valid**] yang akan menampilkan pesan *error*.

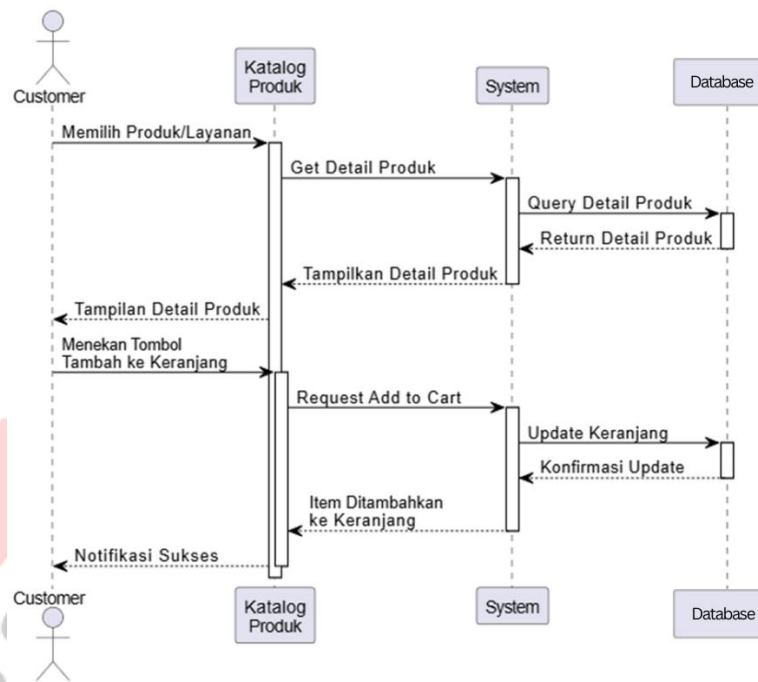
#### 16. Sequence Diagram Track Orders



Gambar 3. 41 Sequence Diagram Track Orders

- a. 1 (satu) aktor yang melakukan kegiatan yaitu *Customer*.
- b. 2 (dua) *lifeline* yang menjelaskan komponen sistem yaitu *System* (sistem) dan *Database* (basis data).
- c. Tidak ada *fragment* khusus yang digunakan.
- d. 3 (tiga) *message* yang menjelaskan alur sistem yaitu Membuka Halaman Pelacakan Pesanan, *Query* Status Pesanan, dan Menampilkan Status Pesanan.
- e. 1 (satu) *return message* yaitu *Return Data* Status Pesanan dari *database*.
- f. Tidak ada *conditional flow* dalam diagram ini.

## 17. Sequence Diagram Add to Cart

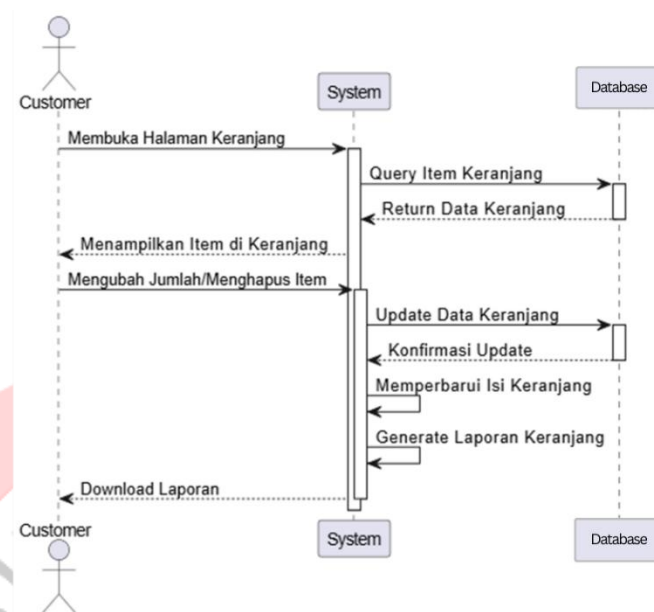


Gambar 3. 42 Sequence Diagram Add to Cart

Berikut penjelasan dari *Sequence Diagram Add to Cart*:

- 1 (satu) aktor yang melakukan kegiatan yaitu *Customer*.
- 3 (tiga) *lifeline* yang menjelaskan komponen sistem yaitu Katalog Produk, *System* (sistem), dan *Database* (basis data).
- Tidak ada *fragment* khusus yang digunakan.
- 8 (delapan) *message* yang menjelaskan alur sistem yaitu Memilih Produk/Layanan, *Get Detail Produk*, *Query Detail Produk*, *Tampilkan Detail Produk*, Menekan Tombol Tambah ke Keranjang, *Request Add to Cart*, *Update Keranjang*, dan Item Ditambahkan ke Keranjang.
- 3 (tiga) *return message* yaitu *Return Detail Produk*, *Konfirmasi Update*, dan Notifikasi Sukses.
- Tidak ada *conditional flow* dalam diagram ini.

## 18. Sequence Diagram Manage Cart

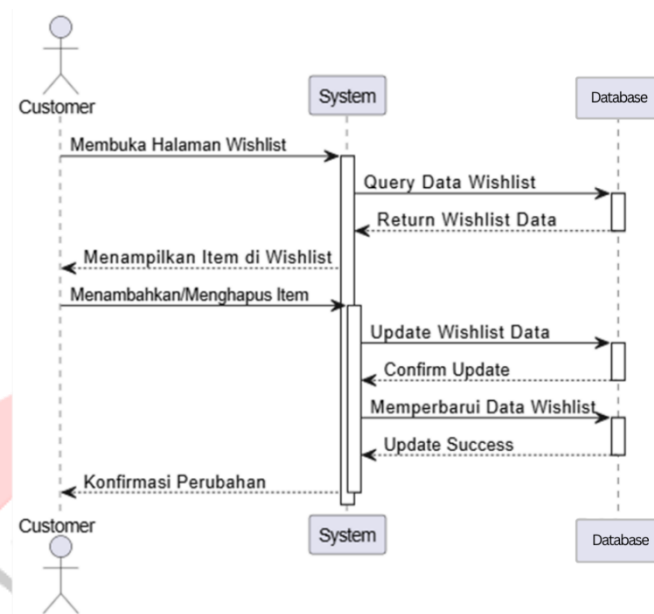


Gambar 3. 43 Sequence Diagram Manage Cart

### Berikut penjelasan dari Sequence Diagram Manage Cart:

- 1 (satu) aktor yang melakukan kegiatan yaitu *Customer*.
- 2 (dua) *lifeline* yang menjelaskan komponen sistem yaitu *System* (sistem) dan *Database* (basis data).
- Tidak ada *fragment* khusus yang digunakan.
- 7 (tujuh) *message* yang menjelaskan alur sistem yaitu Membuka Halaman Keranjang, *Query* Item Keranjang, Menampilkan Item di Keranjang, Mengubah Jumlah/Menghapus Item, *Update Data* Keranjang, Memperbarui Isi Keranjang, dan *Generate* Laporan Keranjang.
- 2 (dua) *return message* yaitu *Return Data* Keranjang dan Konfirmasi *Update*.
- Tidak ada *conditional flow* tetapi ada operasi untuk mengubah jumlah atau menghapus item dari keranjang.

## 19. Sequence Diagram Manage Wishlist

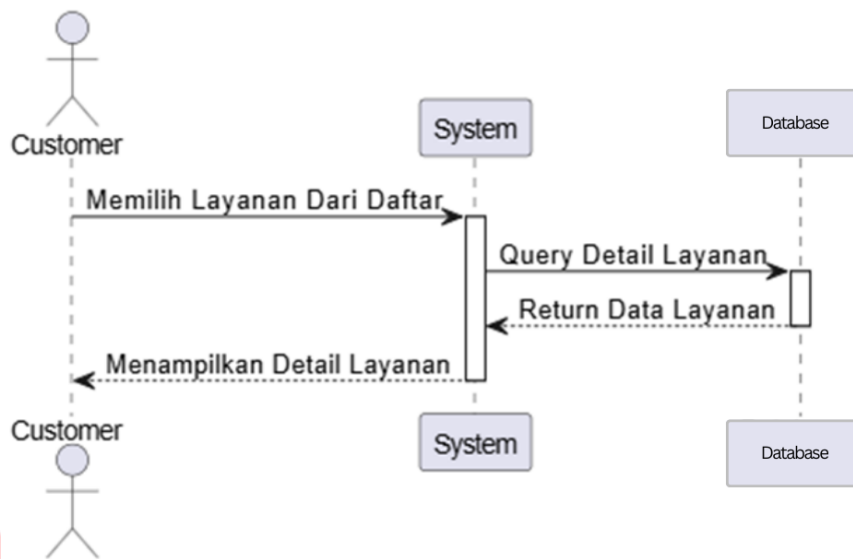


Gambar 3. 44 Sequence Diagram Manage Wishlist

Berikut penjelasan dari *Sequence Diagram Manage Wishlist*:

- 1 (satu) aktor yang melakukan kegiatan yaitu *Customer*.
- 2 (dua) *lifeline* yang menjelaskan komponen sistem yaitu *System* (sistem) dan *Database* (basis data).
- Tidak ada *fragment* khusus yang digunakan.
- 7 (tujuh) *message* yang menjelaskan alur sistem yaitu *Membuka Halaman Wishlist*, *Query Data Wishlist*, *Menampilkan Item di Wishlist*, *Menambahkan/Menghapus Item*, *Update Wishlist Data*, *Memperbarui Data Wishlist*, dan *Update Success*.
- 3 (tiga) *return message* yaitu *Return Wishlist Data*, *Confirm Update*, dan *Konfirmasi Perubahan*.
- Tidak ada *conditional flow* tetapi ada operasi untuk menambahkan atau menghapus item dari *wishlist*.

## 20. Sequence Diagram View Service Details

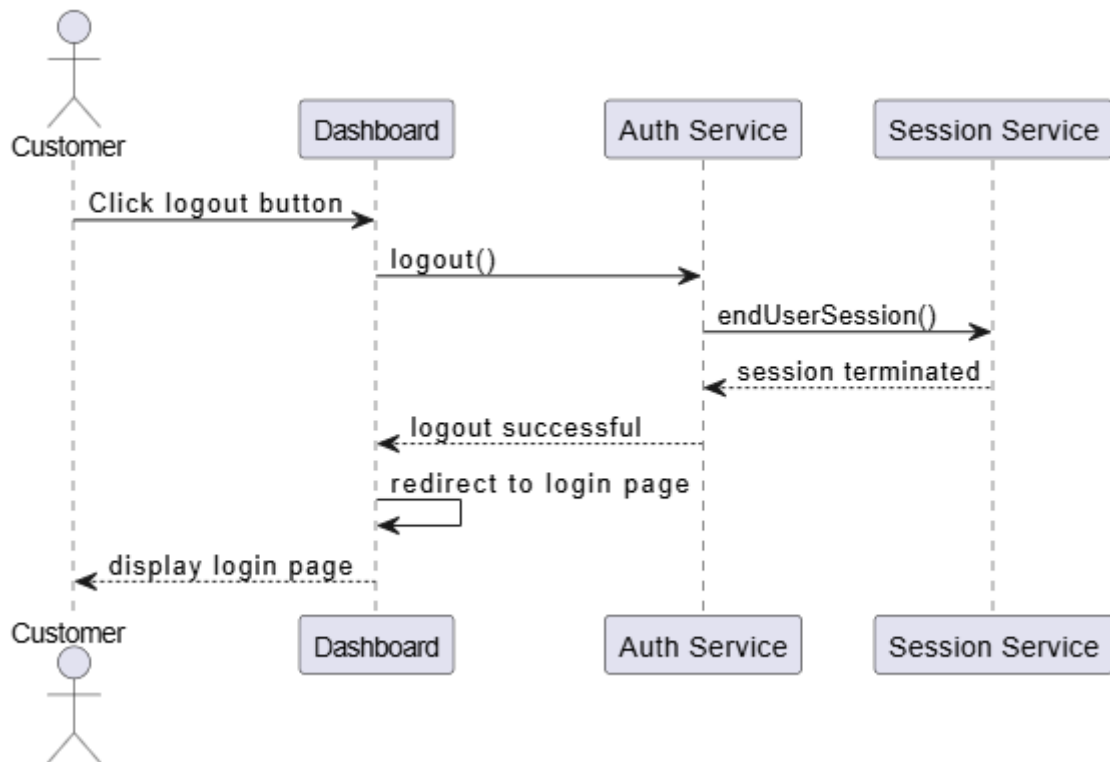


Gambar 3. 45 Sequence Diagram View Service Details

Berikut penjelasan dari *Sequence Diagram View Service Details*:

- 1 (satu) aktor yang melakukan kegiatan yaitu *Customer*.
- 2 (dua) *lifeline* yang menjelaskan komponen sistem yaitu *System* (sistem) dan *Database* (basis data).
- Tidak ada *fragment* khusus yang digunakan.
- 3 (tiga) *message* yang menjelaskan alur sistem yaitu *Memilih Layanan Dari Daftar*, *Query Detail Layanan*, dan *Menampilkan Detail Layanan*.
- 1 (satu) *return message* yaitu *Return Data Layanan* dari *database*.
- Tidak ada *conditional flow* dalam diagram ini.

## 21. Sequence Diagram Logout

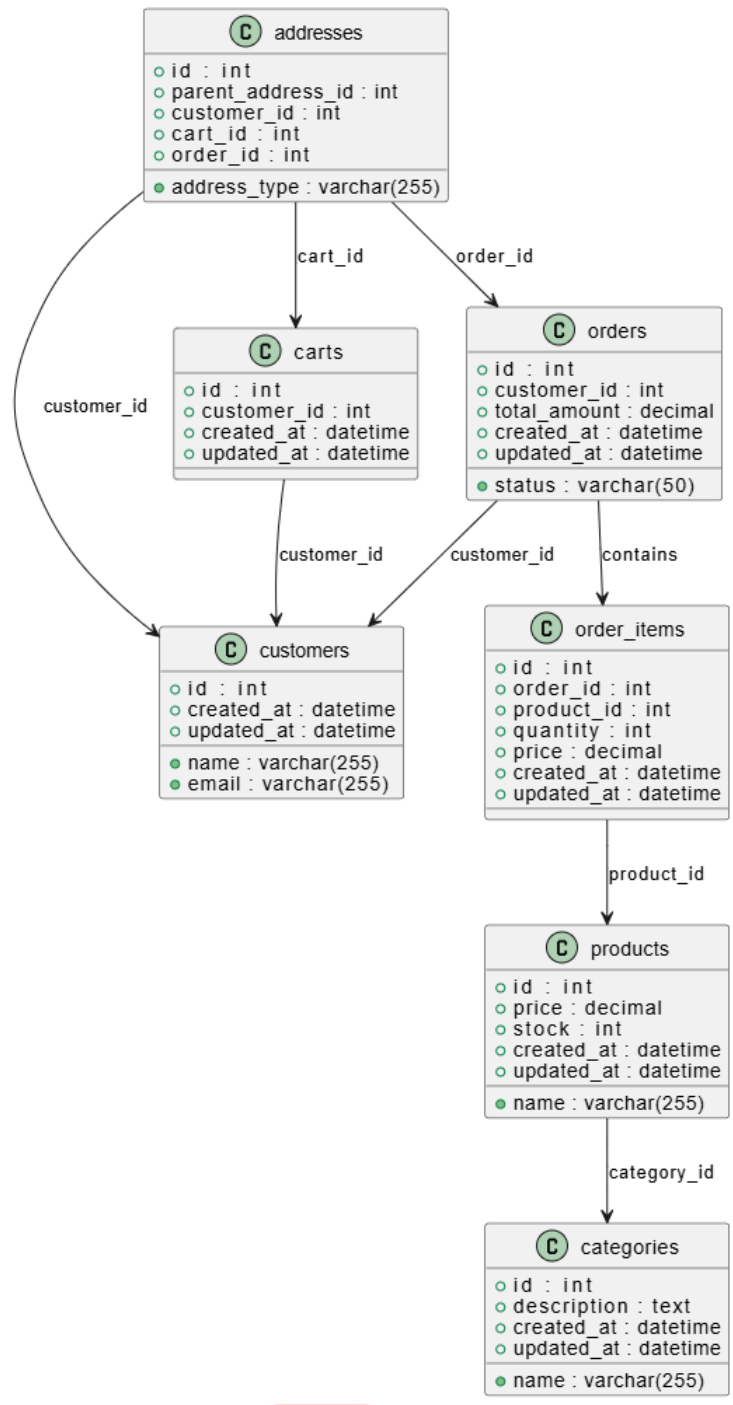


Gambar 3. 46 Sequence Diagram Logout

Berikut penjelasan dari *Sequence Diagram Logout*:

- 1 (satu) aktor yang melakukan kegiatan yaitu *Customer*.
- 3 (tiga) *lifeline* yang menjelaskan komponen sistem yaitu *Dashboard*, *Auth Service* (layanan autentikasi), dan *Session Service* (layanan sesi).
- Tidak ada *fragment* khusus yang digunakan.
- 6 (enam) *message* yang menjelaskan alur sistem yaitu *Click logout button*, *logout()*, *endUserSession()*, *session terminated*, *logout successful*, dan *redirect to login page*.
- 3 (tiga) *return message* yaitu *session terminated*, *logout successful*, dan *display login page*.
- Tidak ada *conditional flow* dalam diagram ini.

### 3.8.5 Class Diagram



Gambar 3. 47 Class Diagram

### 3.8.6 Rancangan Database

#### a. *Addresses*

Tabel ini digunakan untuk menyimpan seluruh data alamat yang terkait dengan pelanggan, keranjang, atau pesanan dalam aplikasi. Tabel ini memiliki 6 atribut dengan id sebagai *primary key*.

Tabel 3. 23 Struktur Data *Addresses*

No	Nama Kolom	Tipe Data	Constraint	Deskripsi
1	id	Int(10)	Auto Increment, <i>Primary Key</i>	Kode unik untuk setiap alamat
2	parent_address_id	Int(10)	Foreign Key	Merujuk ke alamat induk (self-reference)
3	customer_id	Int(10)	Foreign Key	Merujuk ke tabel <i>customers</i>
4	cart_id	Int(10)	Foreign Key	Merujuk ke tabel <i>carts</i>
5	order_id	Int(10)	Foreign Key	Merujuk ke tabel <i>orders</i>
6	address_id	Varchar(255)	Foreign Key	Jenis alamat

#### b. *Customers*

Tabel ini digunakan untuk menyimpan data pelanggan yang terdaftar dalam aplikasi. Tabel ini memiliki 5 atribut dengan id sebagai *primary key*.

Tabel 3. 24 Struktur Data *Customers*

No	Nama Kolom	Tipe Data	Constraint	Deskripsi
1	Id	Int(10)	Auto Increment, <i>Primary Key</i>	Kode unik untuk pelanggan
2	Name	Varchar(255)	Not Null	Nama pelanggan
3	Email	Varchar(255)	Not Null	Email pelanggan
4	Created_at	Datetime	Default CURRENT_TIMESTAMP	Waktu pembuatan data
5	Updated_at	Datetime	Default CURRENT_TIMESTAMP	Waktu pembuatan data

c. *Carts*

Tabel ini digunakan untuk menyimpan data keranjang belanja pelanggan. Tabel ini memiliki 4 atribut dengan id sebagai *primary key*

Tabel 3. 25 Struktur Data *Carts*

No	Nama Kolom	Tipe Data	Constraint	Deskripsi
1	id	Int(10)	Auto Increment, <i>Primary Key</i>	Kode unik untuk keranjang
2	<i>customer_id</i>	Int(10)	Foreign Key	Merujuk ke tabel <i>customers</i>
3	created_at	Datetime	Default CURRENT_TIMESTAMP	Waktu pembuatan keranjang
4	updated_at	Datetime	Default CURRENT_TIMESTAMP	Waktu pembuatan keranjang

d. *Orders*

Tabel ini digunakan untuk menyimpan data pesanan pelanggan dalam aplikasi.

Tabel ini memiliki 6 atribut dengan id sebagai *primary key*.

Tabel 3. 26 Struktur Data *Orders*

No	Nama Kolom	Tipe Data	Constraint	Deskripsi
1	Id	Int(10)	Auto Increment, <i>Primary Key</i>	Kode unik untuk pesanan
2	<i>customer_id</i>	Int(10)	Foreign Key	Merujuk ke tabel <i>customers</i>
3	total_amount	Decimal(10, 2)	Not Null	Total jumlah pesanan
4	status	Varchar(50)	Not Null	Status pesanan
5	created_at	Datetime	Default CURRENT_TIMESTAMP	Waktu pembuatan pesanan
6	updated_at	Datetime	Default CURRENT_TIMESTAMP	Waktu pembuatan pesanan

e. *Order\_items*

Tabel ini digunakan untuk menyimpan rincian item pada setiap pesanan. Tabel ini memiliki 7 atribut dengan id sebagai *primary key*.

Tabel 3. 27 Struktur Data *Order\_items*

No	Nama Kolom	Tipe Data	Constraint	Deskripsi
1	id	Int(10)	Auto Increment, <i>Primary Key</i>	Kode unik untuk item pesanan
2	<i>order_id</i>	Int(10)	Foreign Key	Merujuk ke tabel <i>orders</i>
3	<i>service_id</i>	Int(10)	Foreign Key	Merujuk ke tabel <i>services</i>
4	quantity	Int(10)	Not Null	Jumlah layanan dalam pesanan
5	price	Decimal(10, 2)	Not Null	Harga layanan
6	created_at	Datetime	Default CURRENT_TIMESTAMP	Waktu pembuatan data
7	updated_at	Datetime	Default CURRENT_TIMESTAMP	Waktu pembuatan data

f. *Services*

Tabel ini digunakan untuk menyimpan data layanan yang tersedia dalam aplikasi.

Tabel ini memiliki 6 atribut dengan id sebagai *primary key*.

Tabel 3. 28 Struktur Data *Services*

No	Nama Kolom	Tipe Data	Constraint	Deskripsi
1	id	Int(10)	Auto Increment, <i>Primary Key</i>	Kode unik untuk layanan
2	name	Varchar(255)	Not Null	Nama layanan

3	price	Decimal(10, 2)	Not Null	Harga layanan
4	created_at	Datetime	Default CURRENT_TIMESTAMP	Waktu pembuatan data
5	updated_at	Datetime	Default CURRENT_TIMESTAMP	Waktu pembuatan data

g. *Categories*

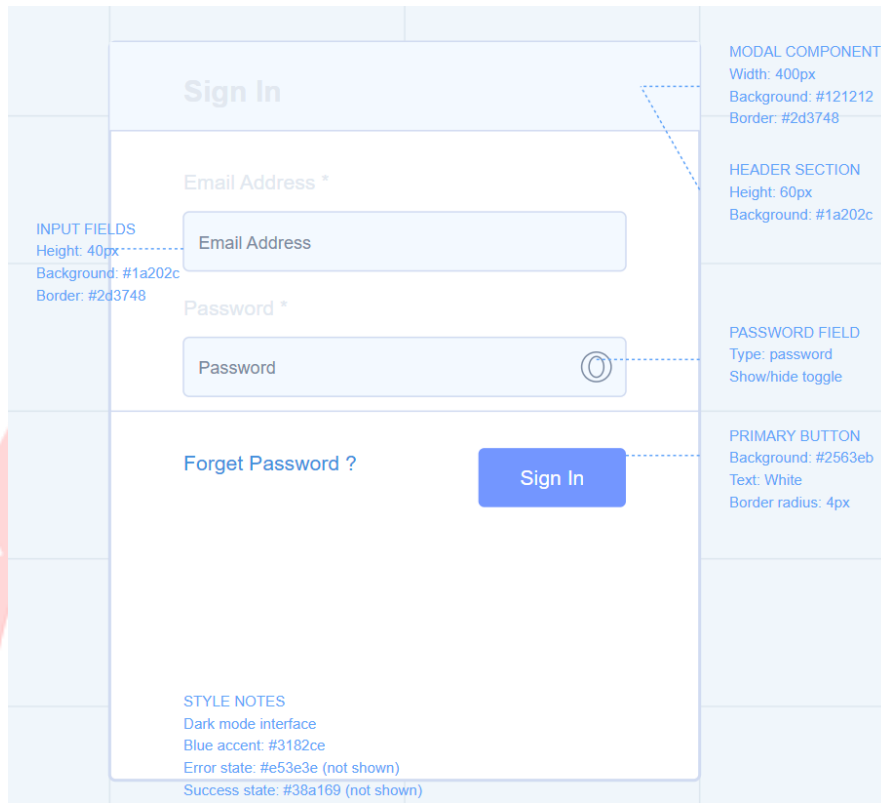
Tabel ini digunakan untuk menyimpan data kategori layanan dalam aplikasi. Tabel ini memiliki 4 atribut dengan id sebagai *primary key*.

Tabel 3. 29 Struktur Data *Categories*

No	Nama Kolom	Tipe Data	Constraint	Deskripsi
1	id	Int(10)	Auto Increment, <i>Primary Key</i>	Kode unik untuk kategori
2	name	Varchar(255)	Not Null	Nama kategori
3	description	Text	Null	Deskripsi Kategori
4	created_at	Datetime	Default CURRENT_TIMESTAMP	Waktu pembuatan data
5	updated_at	Datetime	Default CURRENT_TIMESTAMP	Waktu pembuatan data

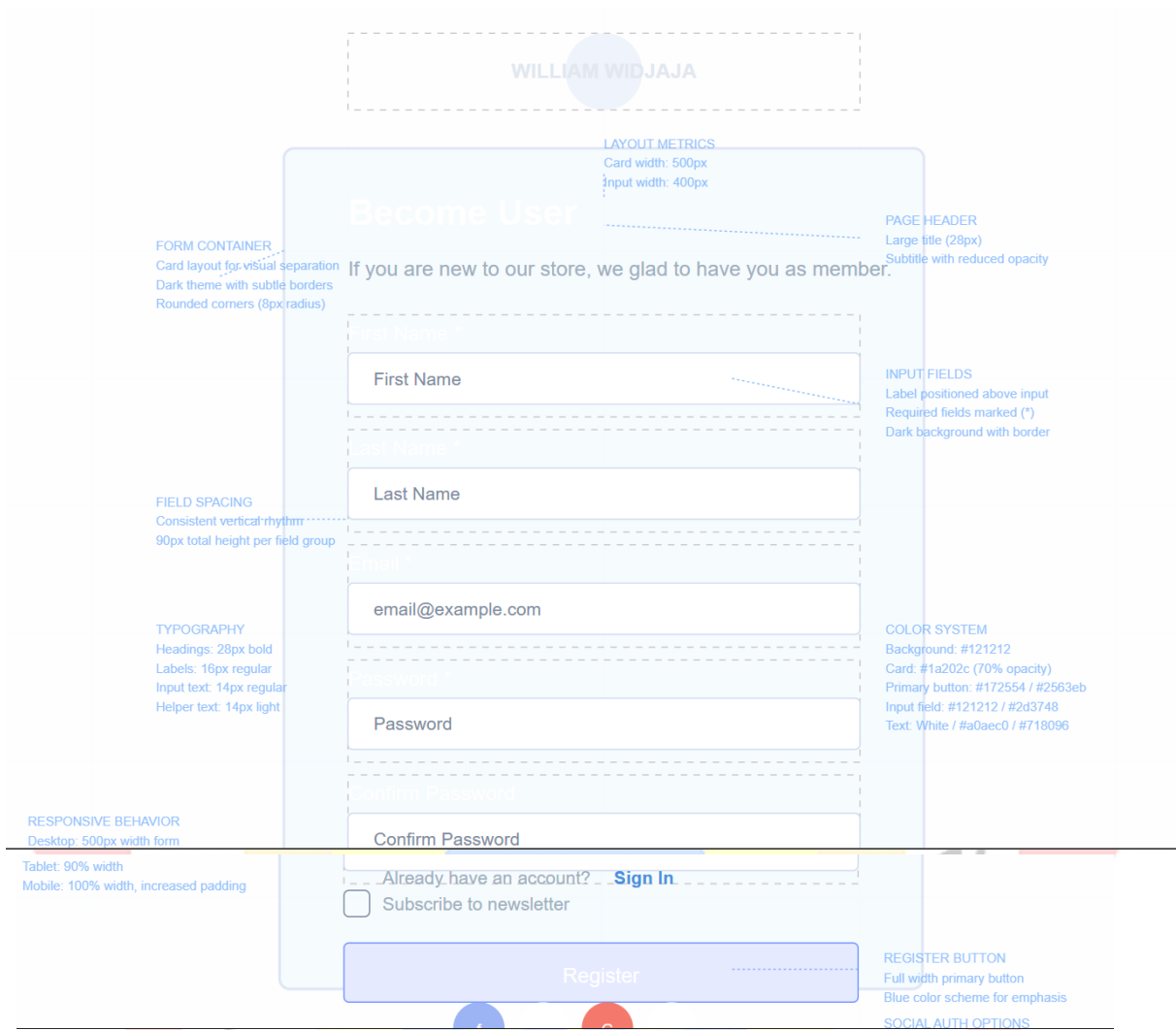
### 3.8.7 Desain Rancangan Sistem

#### 1. Tampilan *Login Customer*



Gambar 3. 48 Tampilan *Login Customer*

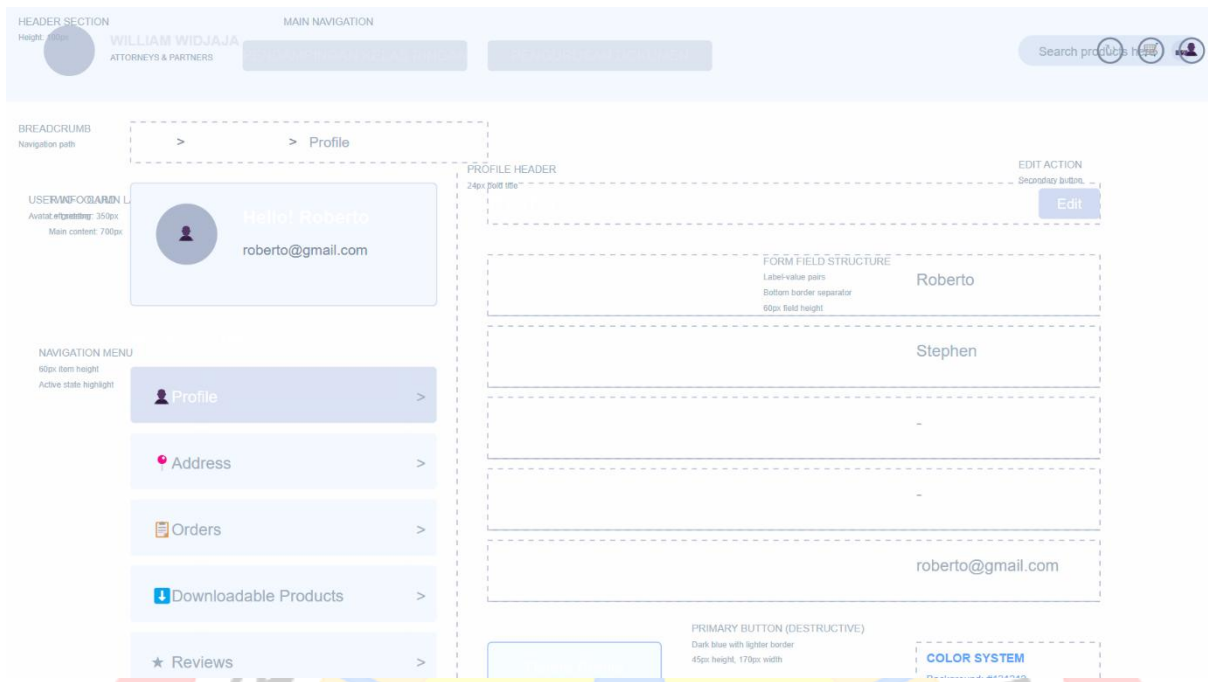
## 2. Tampilan Register Customer



Gambar 3. 49 Tampilan Register Customer



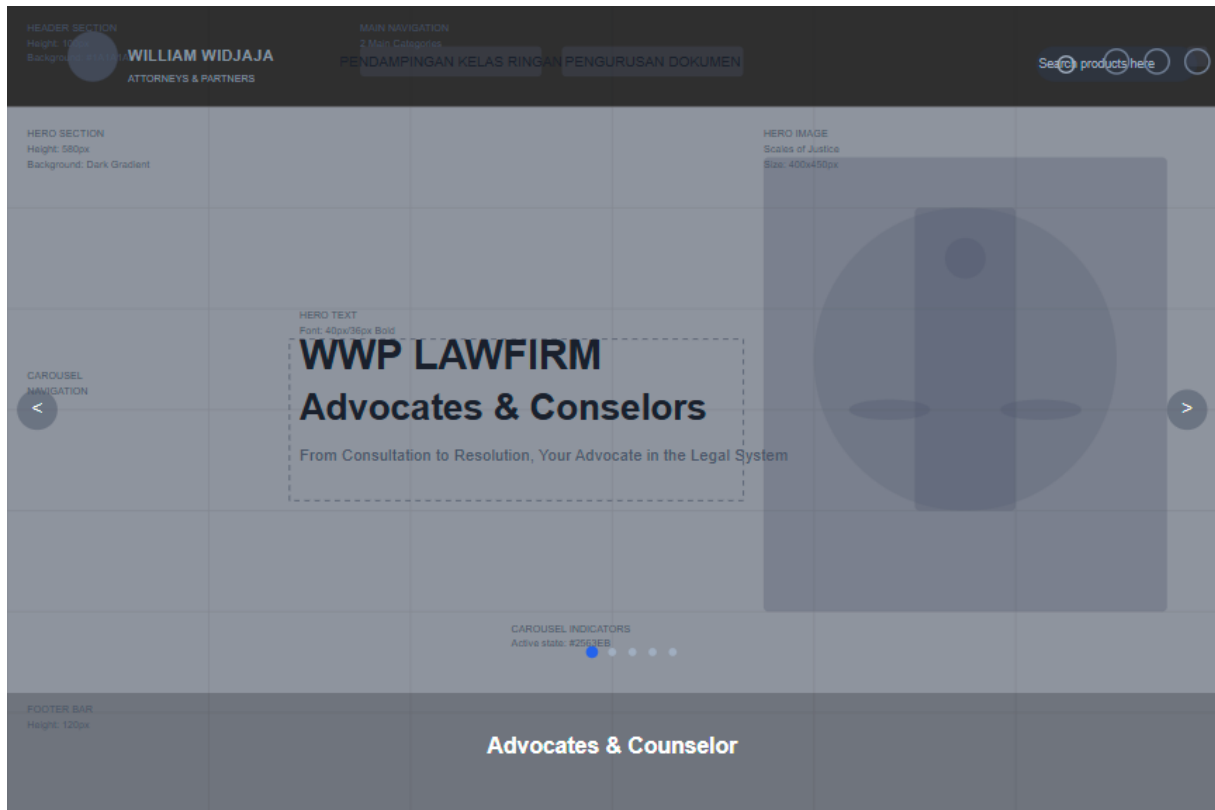
### 3. Tampilan *Profile Customer*



Gambar 3. 50 Tampilan *Profile Customer*



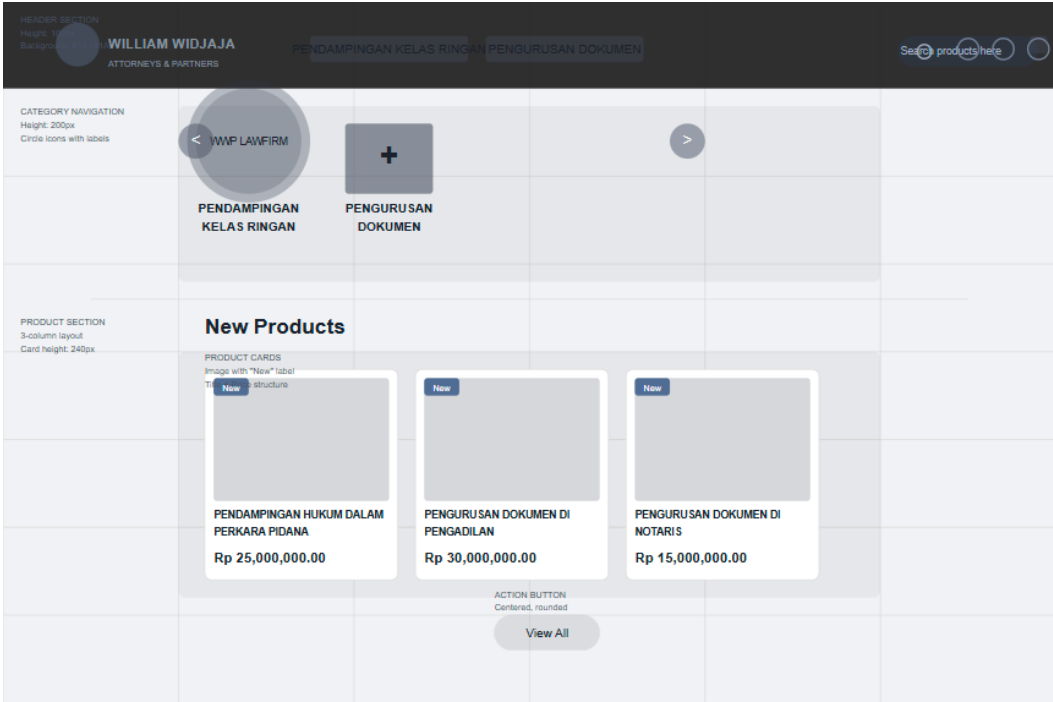
#### 4. Tampilan *Dashboard Customer*



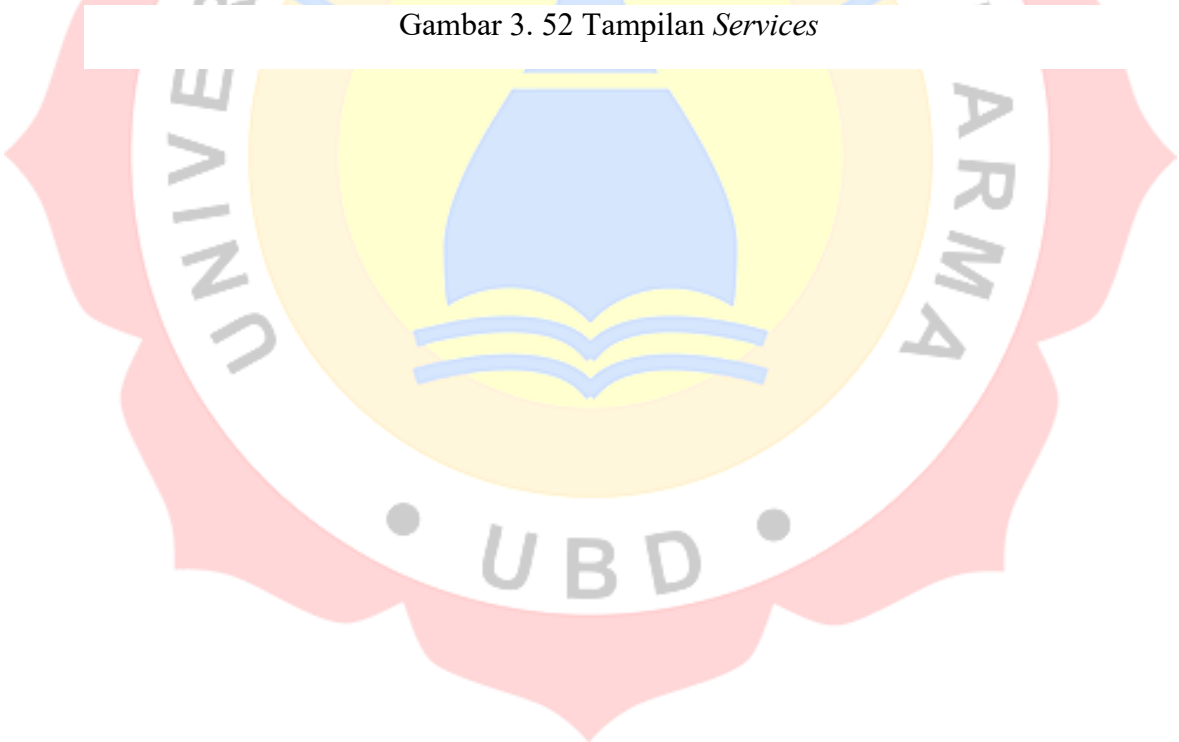
Gambar 3. 51 Tampilan *Dashboard Customer*



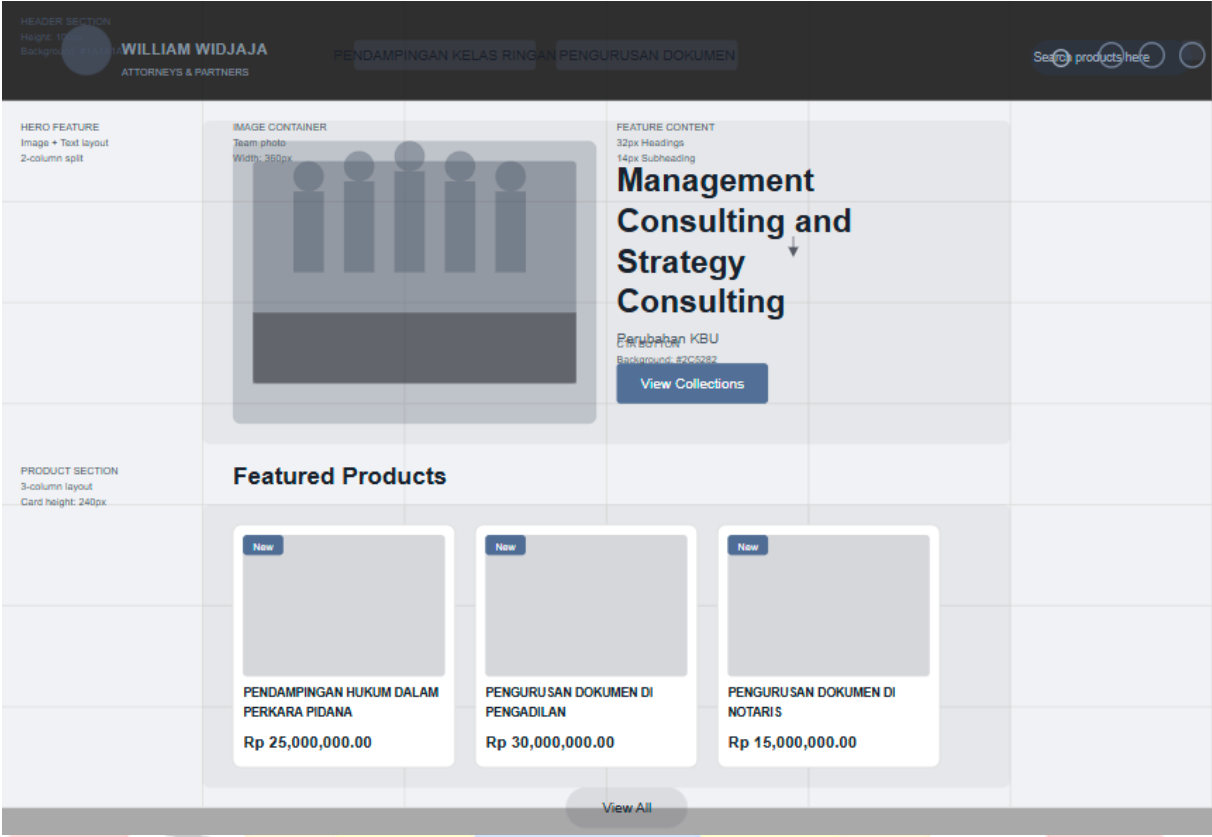
5. Tampilan Services



Gambar 3. 52 Tampilan Services



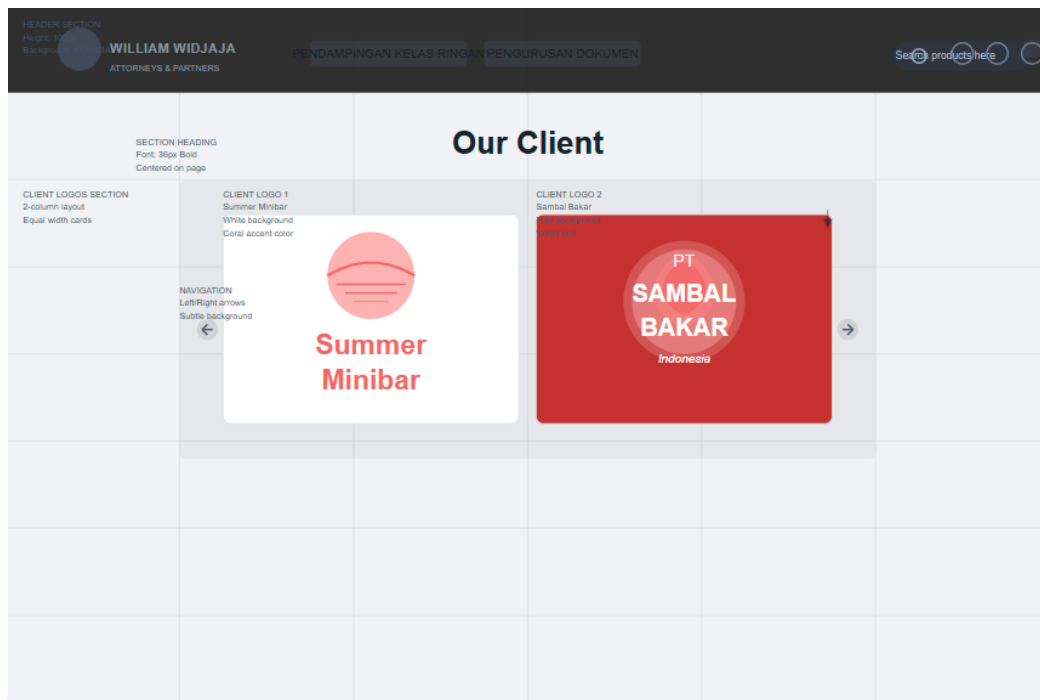
6. Tampilan Katalog *Service*



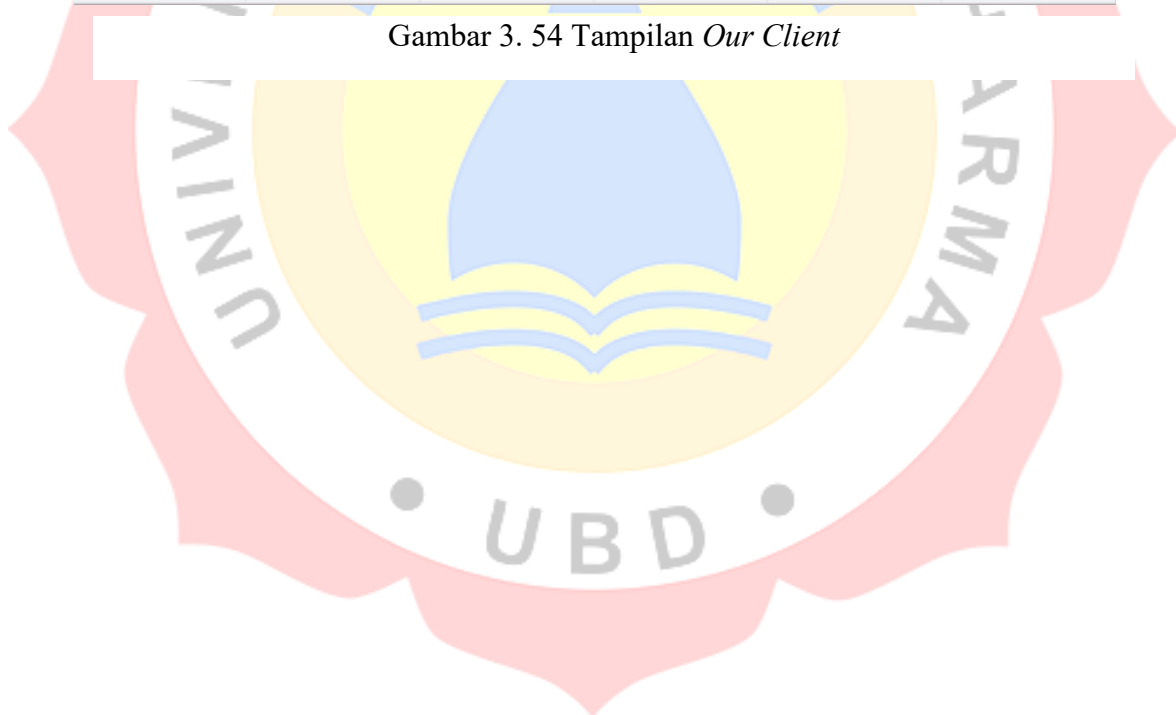
Gambar 3. 53 Tampilan Katalog *Service*



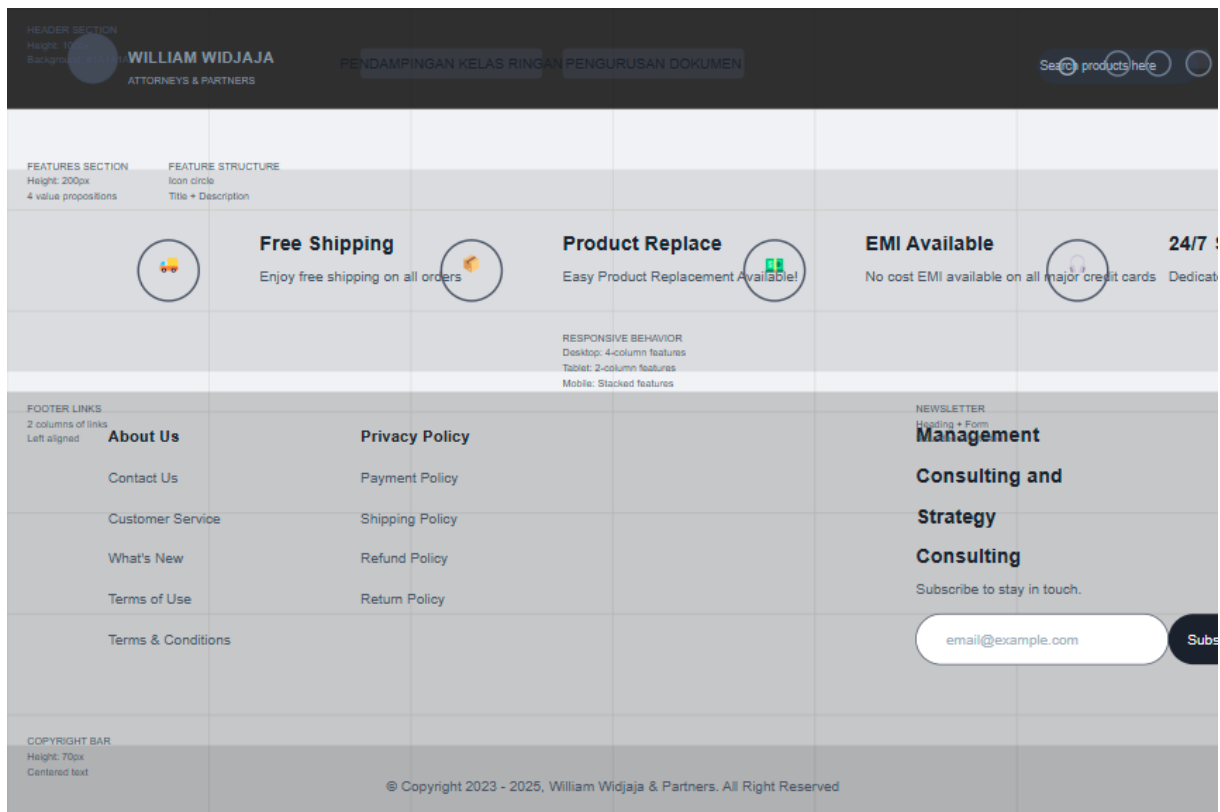
## 7. Tampilan *Our Client*



Gambar 3. 54 Tampilan *Our Client*



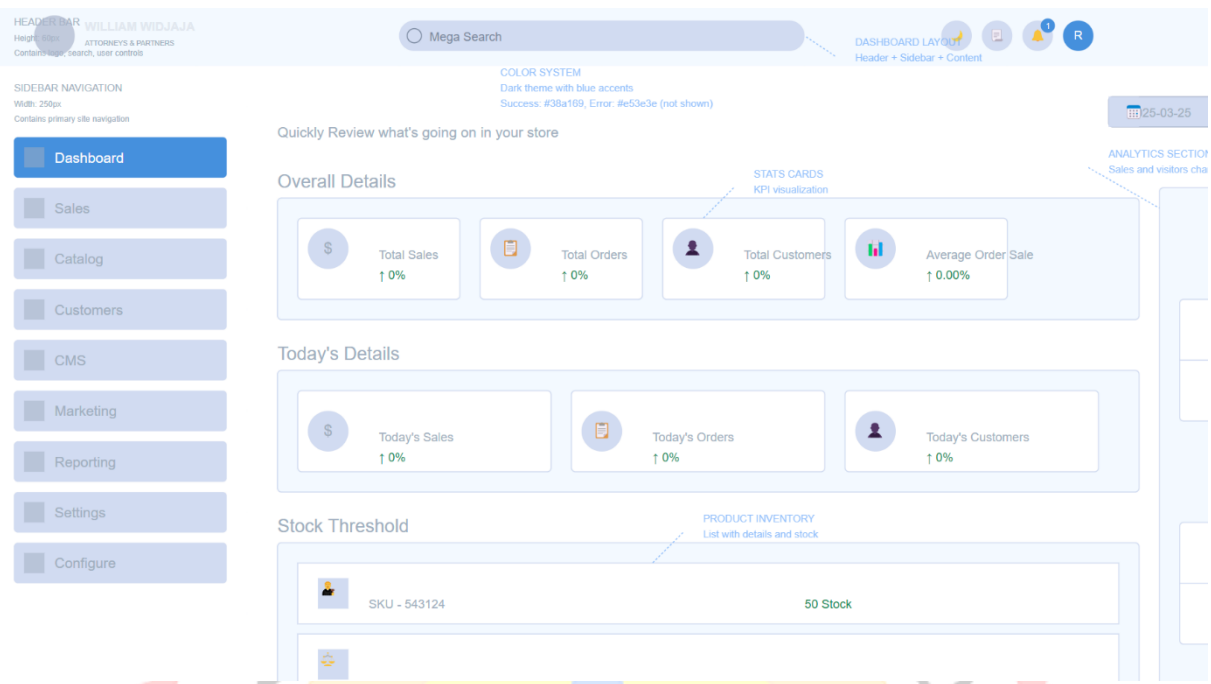
## 8. Tampilan *Footer*



Gambar 3. 55 Tampilan *Footer*



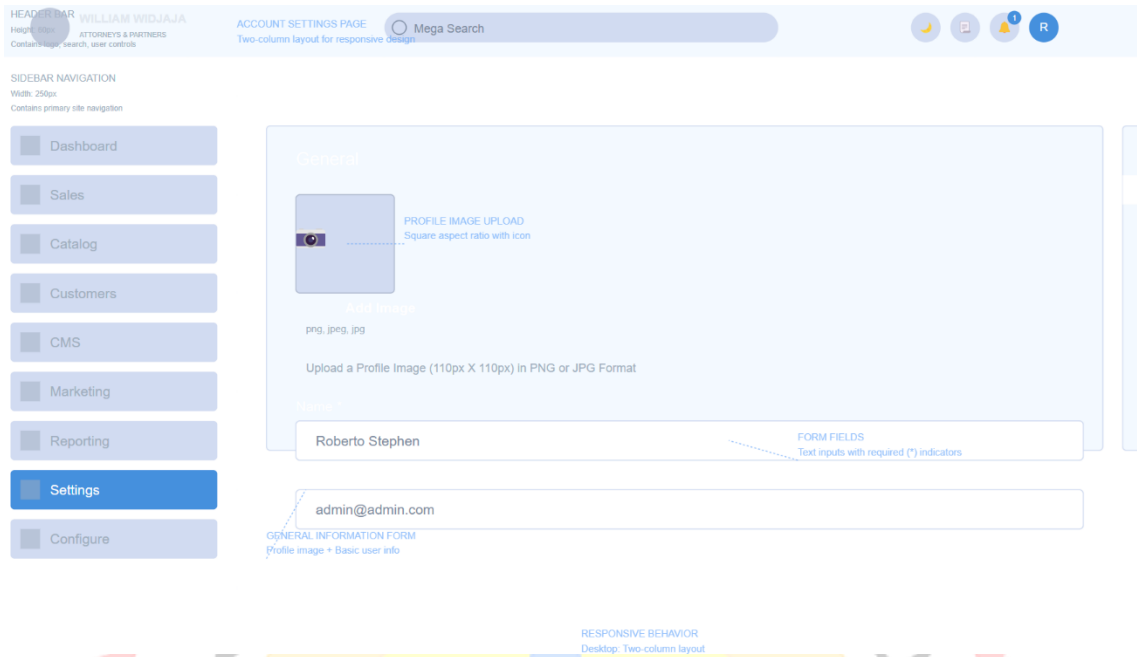
## 9. Tampilan *Dashboard Admin*



Gambar 3. 56 Tampilan *Dashboard Admin*



## 10. Tampilan Account Profile Admin



Gambar 3. 57 Tampilan Account Profile Admin



## 11. Tampilan *Order Admin*

The screenshot displays an administrative interface for order management. The header includes the user name 'WILLIAM WIDJAJA' and a 'Mega Search' bar. A sidebar on the left lists navigation options such as Dashboard, Sales, Orders, Shipments, Invoices, Refunds, Transactions, Catalog, Customers, CMS, Marketing, and Reporting. The main content area shows a table of orders with the following data:

Order ID / Date / Status	Grand Total / Pay Via / Channel	Customer / Email / Location	Items
2025-01-12 16:16:50 <span>Completed</span>	Pay By - Cash On Delivery Default	roberto2626@gmail.com Tangerang, Banten, ID	
2025-01-10 14:58:30 <span>Completed</span>	Pay By - Cash On Delivery Default	roberto@gmail.com Tangerang, Banten, ID	
2025-01-10 14:41:19 <span>Completed</span>	Pay By - Cash On Delivery Default	roberto@gmail.com Tangerang, Banten, ID	<small>ITEM INDICATORS Quantity badge (red)</small>
2025-01-10 14:21:26 <span>Completed</span>	Pay By - Cash On Delivery Default	roberto@gmail.com Tangerang, Banten, ID	

Annotations in the image include: 'DATA TABLE COMPONENT' pointing to the table structure, 'STATUS BADGES' pointing to the green 'Completed' pill, and 'ITEM INDICATORS' pointing to the red quantity badge.

Gambar 3. 58 Tampilan *Order Admin*



## 12. Tampilan *Shipments Admin*

The screenshot displays the 'SHIPMENTS MANAGEMENT PAGE' with a table-based layout. The interface includes a header bar with a 'Mega Search' field, a sidebar navigation menu, and a main content area with a table and pagination controls.

**NAVIGATION SIDEBAR:**

- Dashboard
- Sales
  - Orders
  - Shipments
  - Invoices
  - Refunds
  - Transactions
- Catalog
- Customers
- CMS
- Marketing
- Reporting

**SHIPMENTS MANAGEMENT PAGE:**

Table-based layout with filtering

Search: 4 Results

PAGINATION CONTROLS: Per page selector + page navigation

Export Filter

ID	Order ID	Total Quantity	Inventory Source	Shipment To	Order Date
			Default	Roberto Stephen	
			Default	Roberto Stephen	
			Default	Roberto Stephen	
1			Default	Roberto Stephen	

**Annotations:**

- DATA TABLE COMPONENT:** 8 columns with header row, Row height: 50px (more compact), Color-coded links and values
- COLUMN STRUCTURE:** ID | Order ID | Quantity | Source | Recipient | Dates | Actions
- DATE FOF:** YYYY-MM, Consistent

Gambar 3. 59 Tampilan *Shipments Admin*



### 13. Tampilan Invoice Admin

The screenshot shows an admin dashboard for 'WILLIAM WIDIAJA ATTORNEYS & PARTNERS'. The main content area is titled 'INVOICES MANAGEMENT PAGE' and displays a table of invoices. The table has 5 columns: ID, Order ID, Grand Total, Status, and Invoice Date. There are 4 results shown, all with a 'Paid' status. Annotations include: 'PAGINATION CONTROLS' (Per page selector + page navigation), 'Export' button, 'Filter' button, 'STATUS BADGES' (Colored pill shapes, Green: Paid, Other states: Pending, Canceled, etc.), 'DATA TABLE COMPONENT' (5 columns with header row, Row height: 50px (compact view), Status badges for payment state), and 'CURRENCY FORMAT' (Including currency code (Rp), Thousands separator, 2 decimal places).

ID	Order ID	Grand Total	Status	Invoice Date
			Paid	
			Paid	
			Paid	
			Paid	

Gambar 3. 60 Tampilan Invoice Admin



## 14. Tampilan *Refunds Admin*

HEADER BAR WILLIAM WIDJAJA ATTORNEYS & PARTNERS  
Height: 40px  
Contains header, search, user controls

NAVIGATION SIDEBAR  
SIDE BAR NAVIGATION  
Width: 250px  
Contains primary site navigation

REFUNDS MANAGEMENT PAGE  
Table-based layout with filtering

Search 0 Results  
ZERO RESULTS STATE  
Shows count when empty

CONSISTENT UI CONTROLS  
Same controls across all list views  
Maintains interface familiarity

Export Filter

ID	Order ID	Refunded Amount	Billed To	Refund Date
No Records Available.				

DATA TABLE COMPONENT  
6 columns with header row  
Consistent with other data tables  
Standardized column widths

EMPTY STATE HANDLING  
Clear messaging when no records  
Maintains table structure

ADDITIONAL CONSIDERATIONS  
Could include "Create Refund" button  
Filter options specific to refunds  
Date range selection for refund periods

Gambar 3. 61 Tampilan *Refunds Admin*



## 15. Tampilan *Transactions Admin*

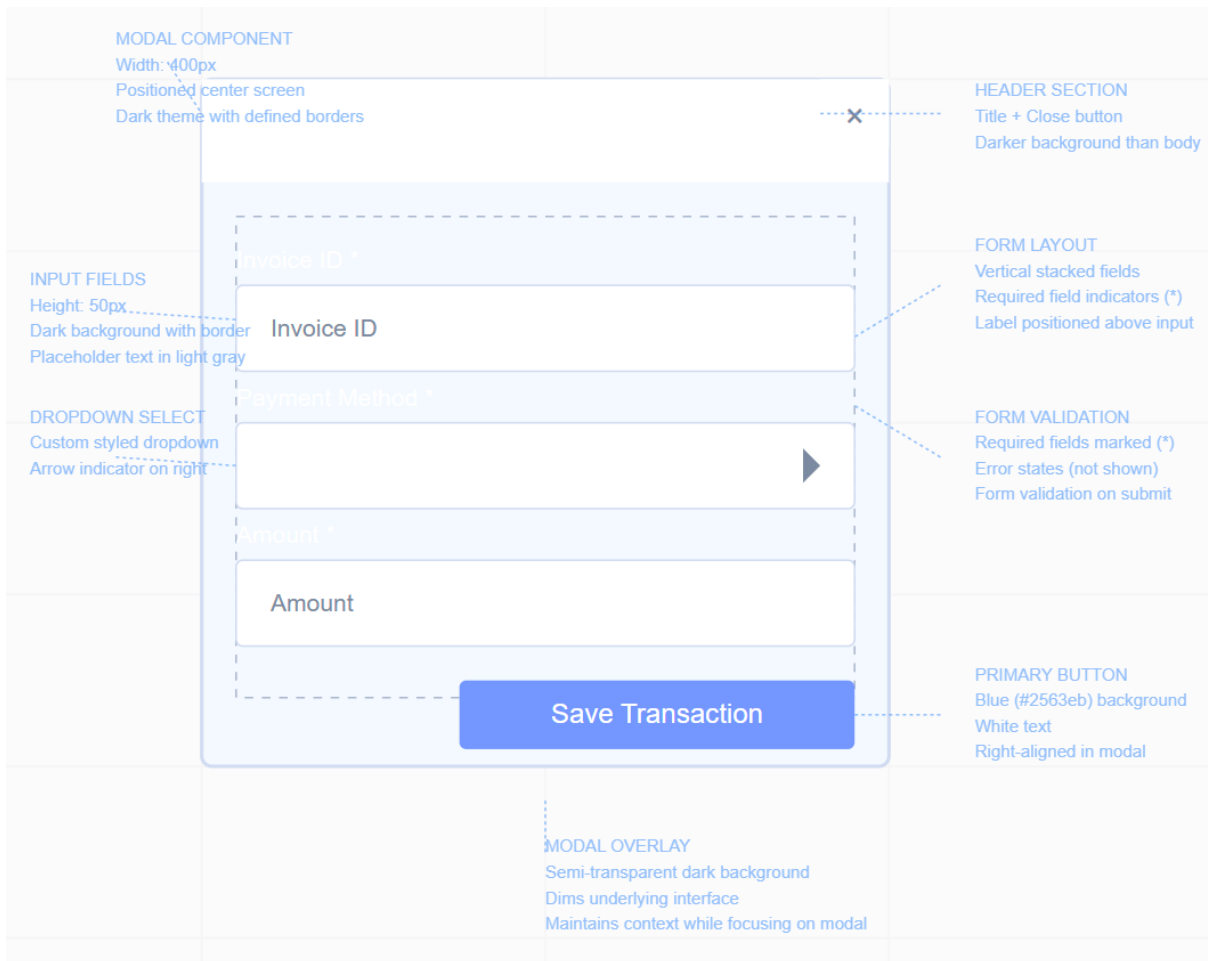
The screenshot displays the 'TRANSACTIONS MANAGEMENT PAGE' in an admin dashboard. The interface includes a header bar with user information (WILLIAM WIDIJAJA, ATTORNEYS & PARTNERS), a search bar (Mega Search), and navigation controls. A sidebar on the left lists menu items: Dashboard, Sales, Orders, Shipments, Invoices, Refunds, Transactions (highlighted), Catalog, Customers, CMS, Marketing, and Reporting. The main content area shows a table with 4 results, filtered and paginated (1 of 1, 10 per page). The table columns are ID, Transaction ID, Amount, Invoice ID, Order ID, Status, and Date. The Status column contains green 'Paid' badges. Annotations point to specific UI elements: 'DATA TABLE COMPONENT' (7 columns, 50px row height), 'TRANSACTION IDs' (complex hash-based identifiers, monospace font, two lines), 'CURRENCY FORMAT' (including IDR, thousands separator, 2 decimal places), 'RELATIONSHIP LINKS' (links to invoices and orders, blue text), and 'STATUS BADGES' (colored pill shapes, Green: Paid, other states: Processing, Failed, etc.).

ID	Transaction ID	Amount	Invoice ID	Order ID	Status	Date
	8d9a77af166a12fb3aa9a7d3e1df2d77		4	4	Paid	
	9c8aebf6f6e703b36d9020d5988b27b9		3	3	Paid	
	3df504064345367d8305cb3e58ae4066		2	2	Paid	
	518a58605e74a47aca93e459a56ae849		1	1	Paid	

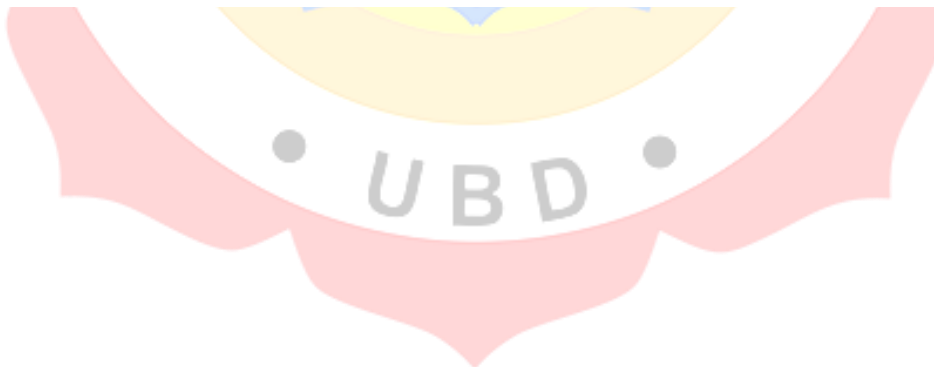
Gambar 3. 62 Tampilan *Transactions Admin*



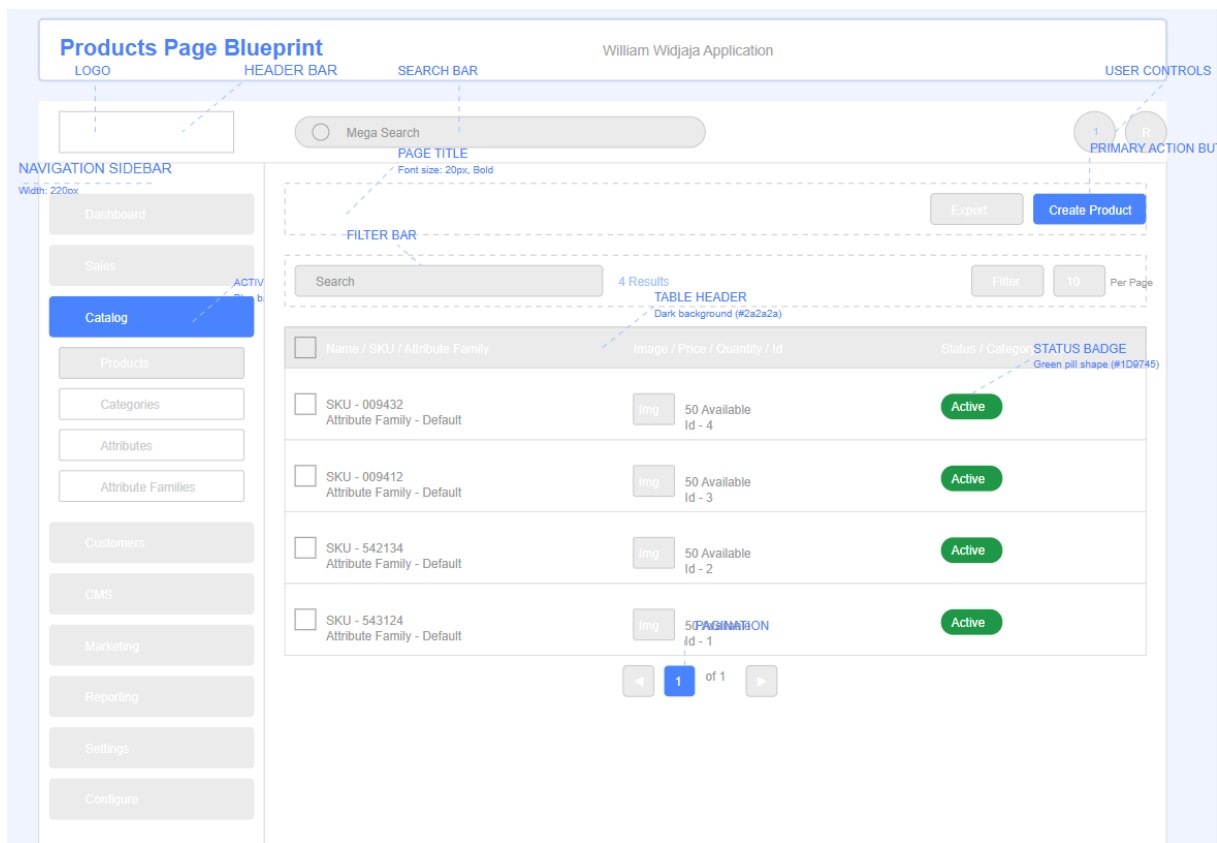
## 16. Tampilan *Create Transactions Admin*



Gambar 3. 63 Tampilan *Create Transactions Admin*



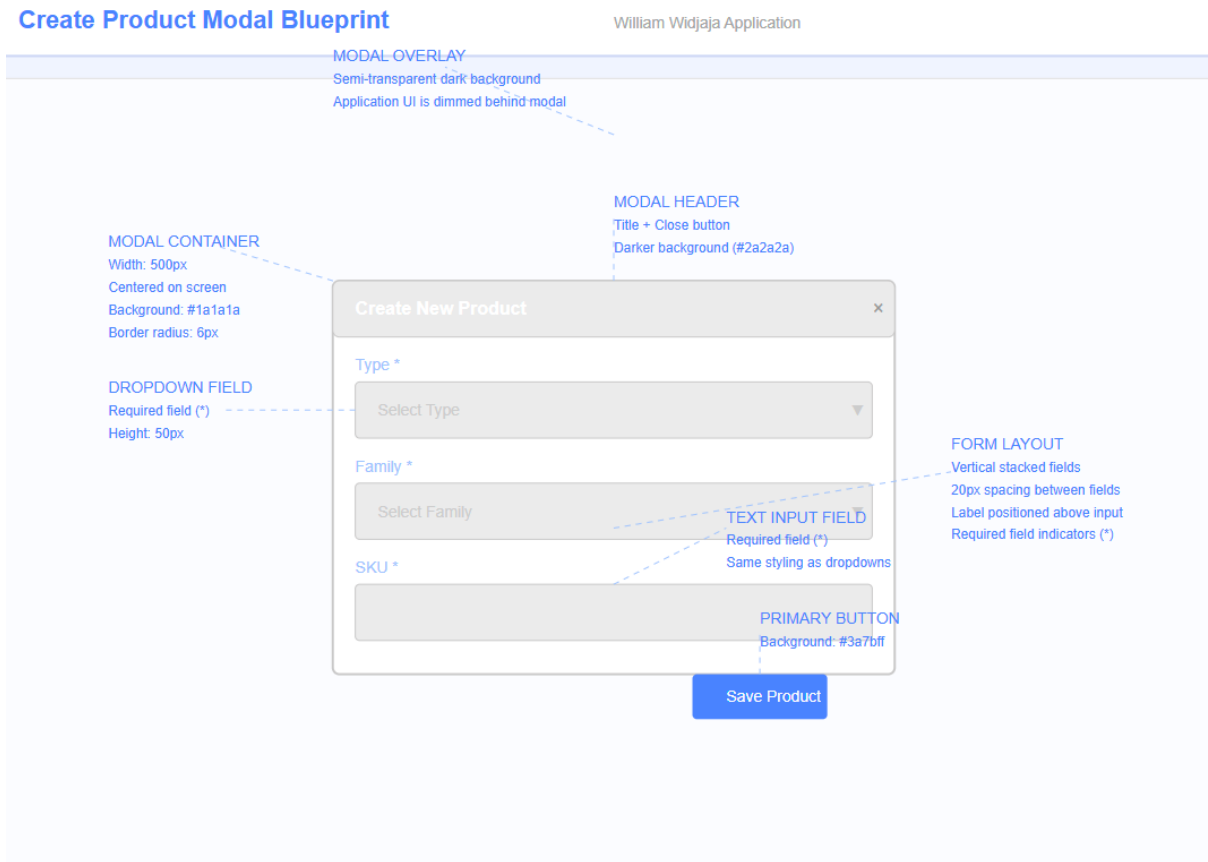
## 17. Tampilan *Catalog Services Admin*



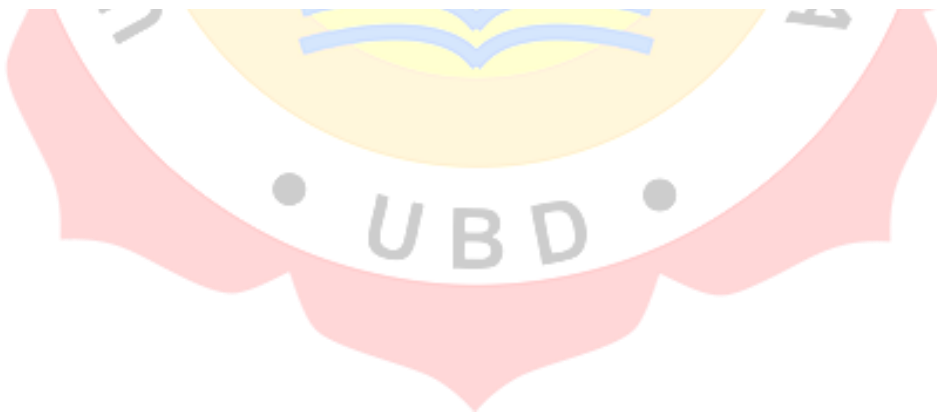
Gambar 3. 64 Tampilan *Catalog Services Admin*



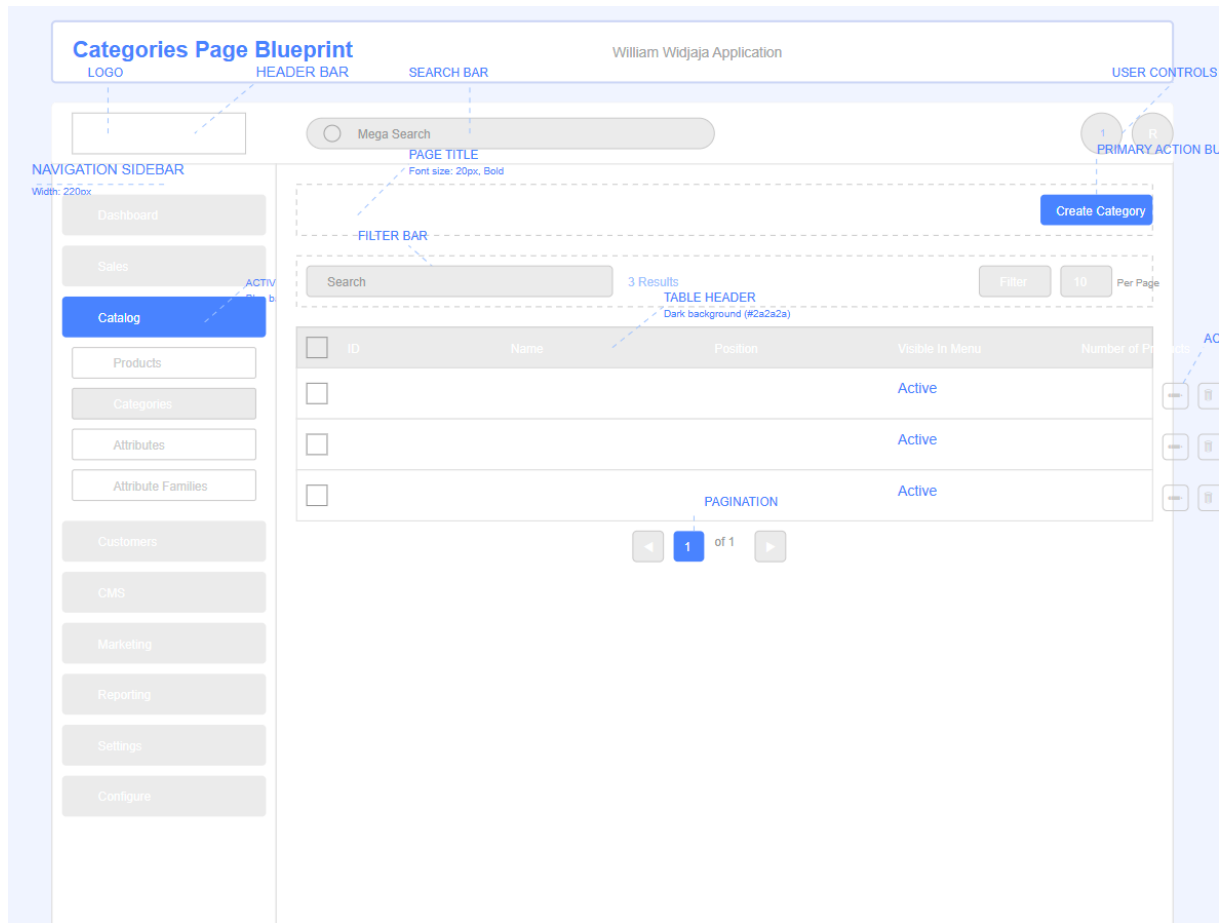
## 18. Tampilan *Create Services Admin*



Gambar 3. 65 Tampilan *Create Services Admin*



## 19. Tampilan *Catalog Categories Admin*



Gambar 3. 66 Tampilan *Catalog Categories Admin*



## 20. Tampilan *Add Categories Admin*

**Add New Category Form Blueprint** William Widjaja Application

Mega Search

Dashboard

Sales

**Catalog**

Products

Categories

Attributes

Attribute Families

Customers

CMS

Marketing

Reporting

Settings

Configure

FORM HEADER  
Page title with action context

SECTION HEADER

REQUIRED FIELD  
Indicated with asterisk

TWO-COLUMN LAYOUT

Name \*

Name

TREEVIEW CONTROL

Parent Category  
Expandable hierarchy

Root

PENDAMPINGAN KELAS RINGAN

PENGURUSAN DOKUMEN

Description \*

RICH TEXT EDITOR  
With formatting toolbar

Magic AI

CHECKBOX GROUP  
Multiple selection

Position \*

Enter Position

DROPPDOWN FIELD  
With selection indicator

Display Mode \*

Products and Description

TOGGLE SWITCH

Visible in Mobile (gray)

COLLAPSIBLE SECTION  
With expand/collapse control

Back Save Category

Gambar 3. 67 Tampilan *Add Categories Admin*



## 21. Tampilan Lanjutan *Add Categories Admin*

The screenshot displays the 'Customers Page Blueprint' for the 'William Widjaja Application'. It features a navigation sidebar on the left with options like Dashboard, Sales, Catalog, Customers (highlighted), Customers, Groups, Reviews, CMS, Marketing, Reporting, Settings, and Configure. The main content area includes a 'Mega Search' bar, a 'Filter Bar' with a search input and 'Filter' button, and a table of customer data. The table has columns for 'Customer Name / Email / Contact Number', 'Status / Gender / Group', and 'Revenue / Order(s) / Address(s)'. Three customer records are listed, all with an 'Active' status. A pagination control shows '1 of 1' results. Annotations point to various UI elements: LOGO, HEADER BAR, SEARCH BAR, USER CONTROLS, NAVIGATION SIDEBAR, PAGE TITLE (Font size: 20px, Bold), FILTER BAR, TABLE HEADER (Dark background (#2a2a2a)), and PAGINATION.

Customer Name / Email / Contact Number	Status / Gender / Group	Revenue / Order(s) / Address(s)
admin@wwplawfirm.com 123321	Active	0 Order(s) 0 Address(s)
roberto2626@gmail.com N/A	Active	1 Order(s) 1 Address(s)
roberto@gmail.com N/A	Active	3 Order(s) 1 Address(s)

Gambar 3. 68 Lanjutan *Add Categories Admin*



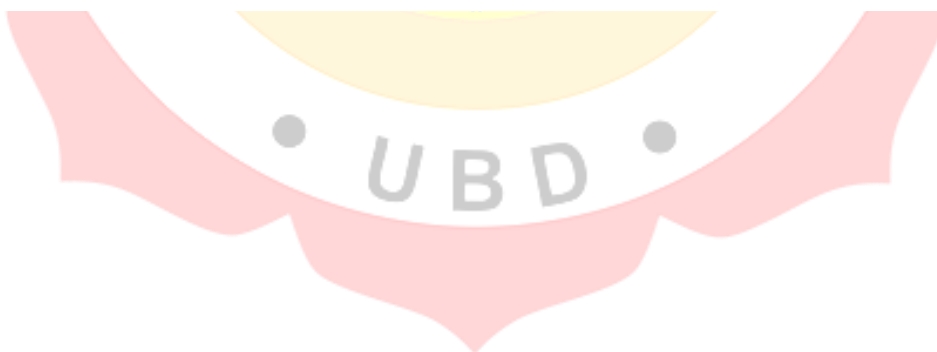
## 22. Tampilan *Customers Admin*

The image shows a wireframe of a web application interface for a 'Customers Admin' section. The interface is titled 'Category Form Media & SEO Blueprint' and is part of a 'William Widjaja Application'. It features a sidebar on the left with navigation options: Dashboard, Sales, Catalog (highlighted), Products, Categories, Attributes, Attribute Families, Customers, CMS, Marketing, Reporting, Settings, and Configure. The main content area is divided into several sections:

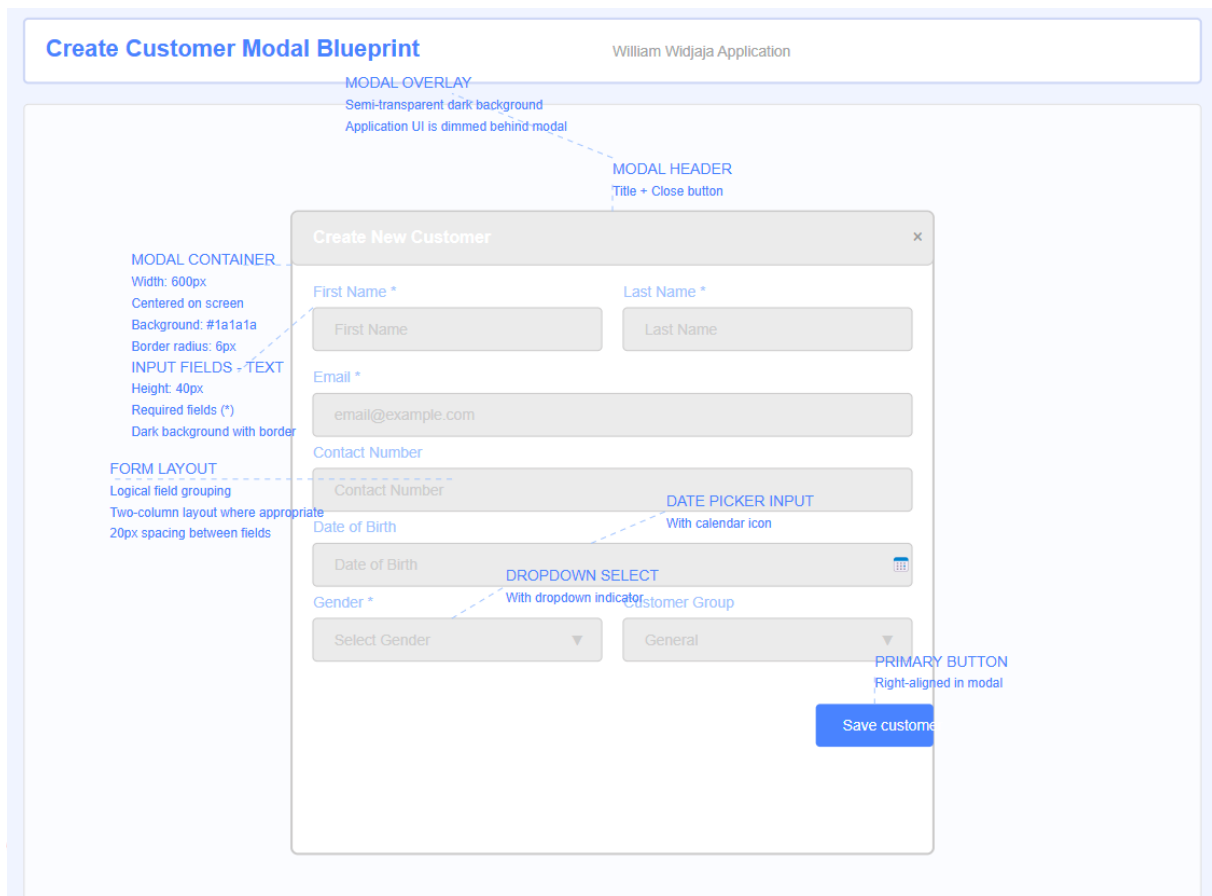
- HEADER BAR:** Contains a search bar and a 'Mega Search' button.
- MEDIA UPLOAD SECTION:** Labeled 'For logo and banner images', it contains two image upload controls. The first is an 'IMAGE PLACEHOLDER' for a logo with a resolution of 110px X 110px. The second is an 'IMAGE UPLOAD CONTROL' for a banner with an aspect ratio of 1320px X 300px.
- SEO SECTION:** Labeled 'Search engine optimization fields', it includes a 'BASE URL' field (non-editable), a 'Meta Title' field (SEO TEXT FIELD), a 'Slug' field (REQUIRED SEO FIELD), a 'Meta Keywords' field (MULTILINE TEXT FIELD), and a 'Meta Description' field.
- STICKY FOOTER:** Located at the bottom right, it contains action buttons.

Annotations with dashed lines point to various elements: 'Mega Search', 'IMAGE PLACEHOLDER', 'IMAGE UPLOAD CONTROL', 'SEO SECTION', 'BASE URL', 'SEO TEXT FIELD', 'REQUIRED SEO FIELD', 'MULTILINE TEXT FIELD', and 'STICKY FOOTER'. A 'TEXT I' label is also present near the top right.

Gambar 3. 69 Tampilan *Customers Admin*



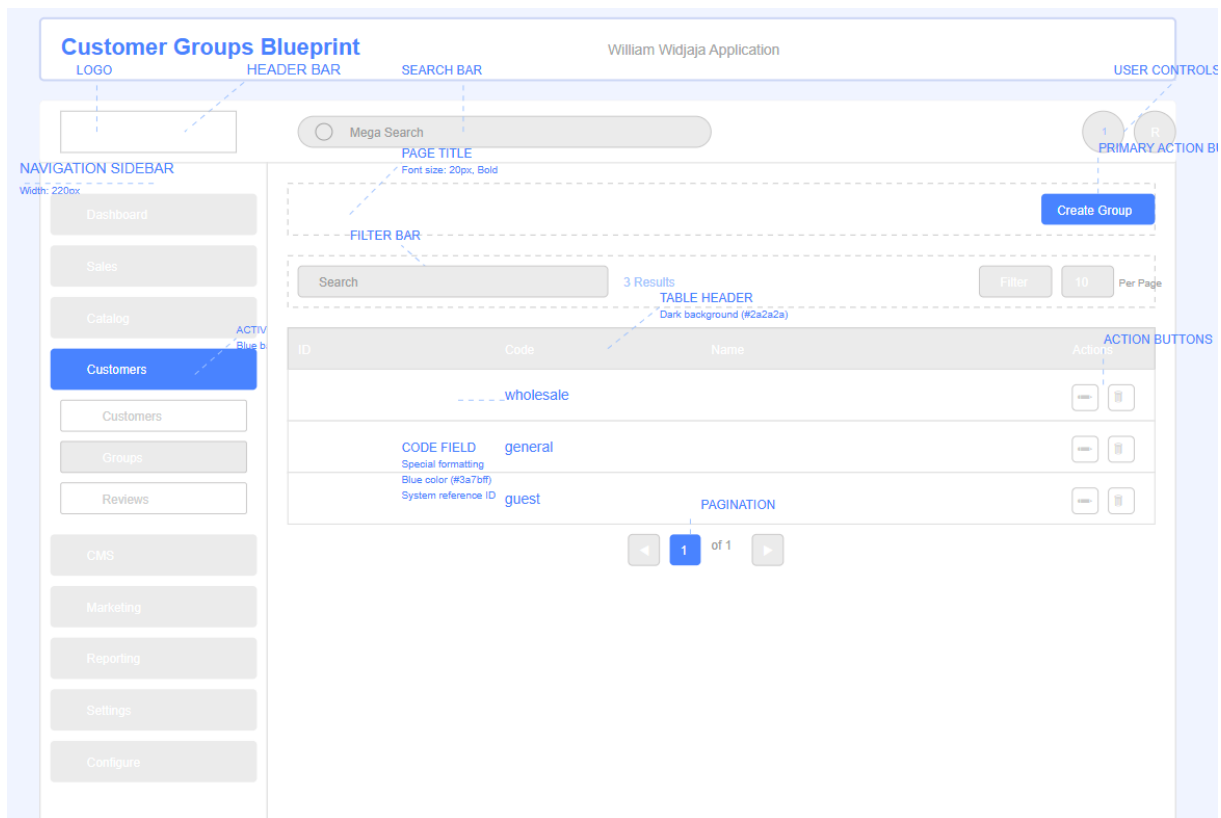
## 23. Tampilan *Create New Customer*



Gambar 3. 70 Tampilan *Create New Customer*



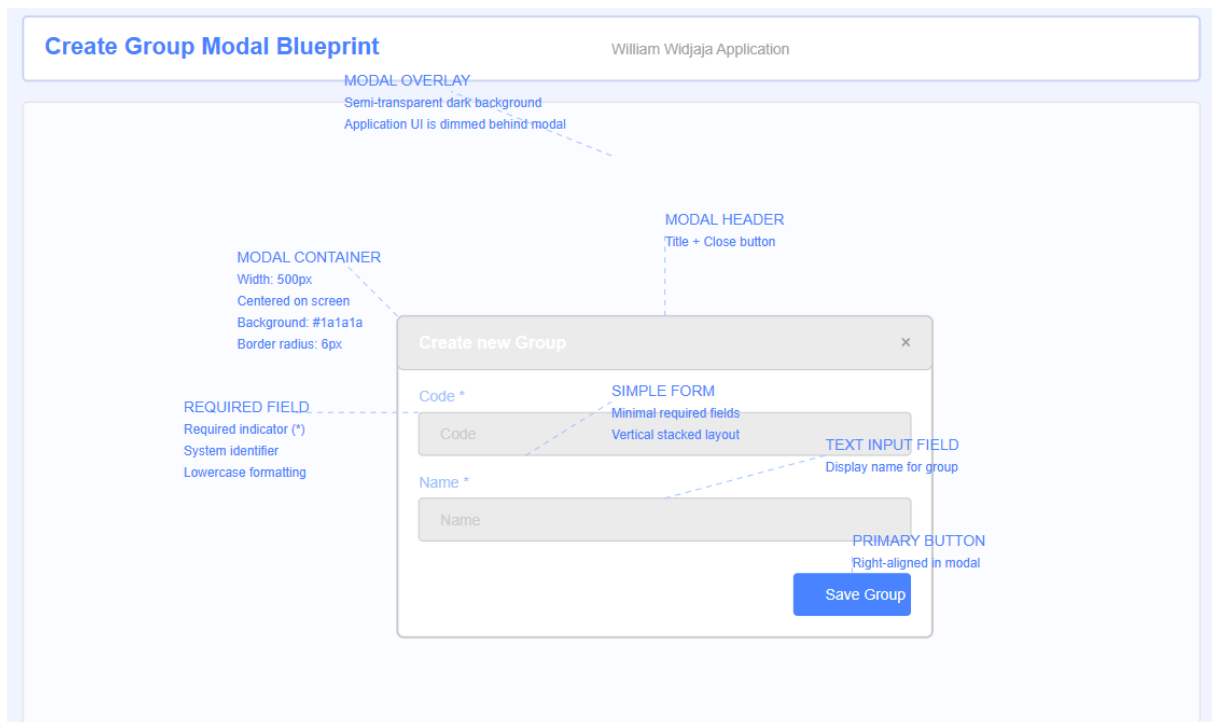
## 24. Tampilan *Groups Admin*



Gambar 3. 71 Tampilan *Groups Admin*



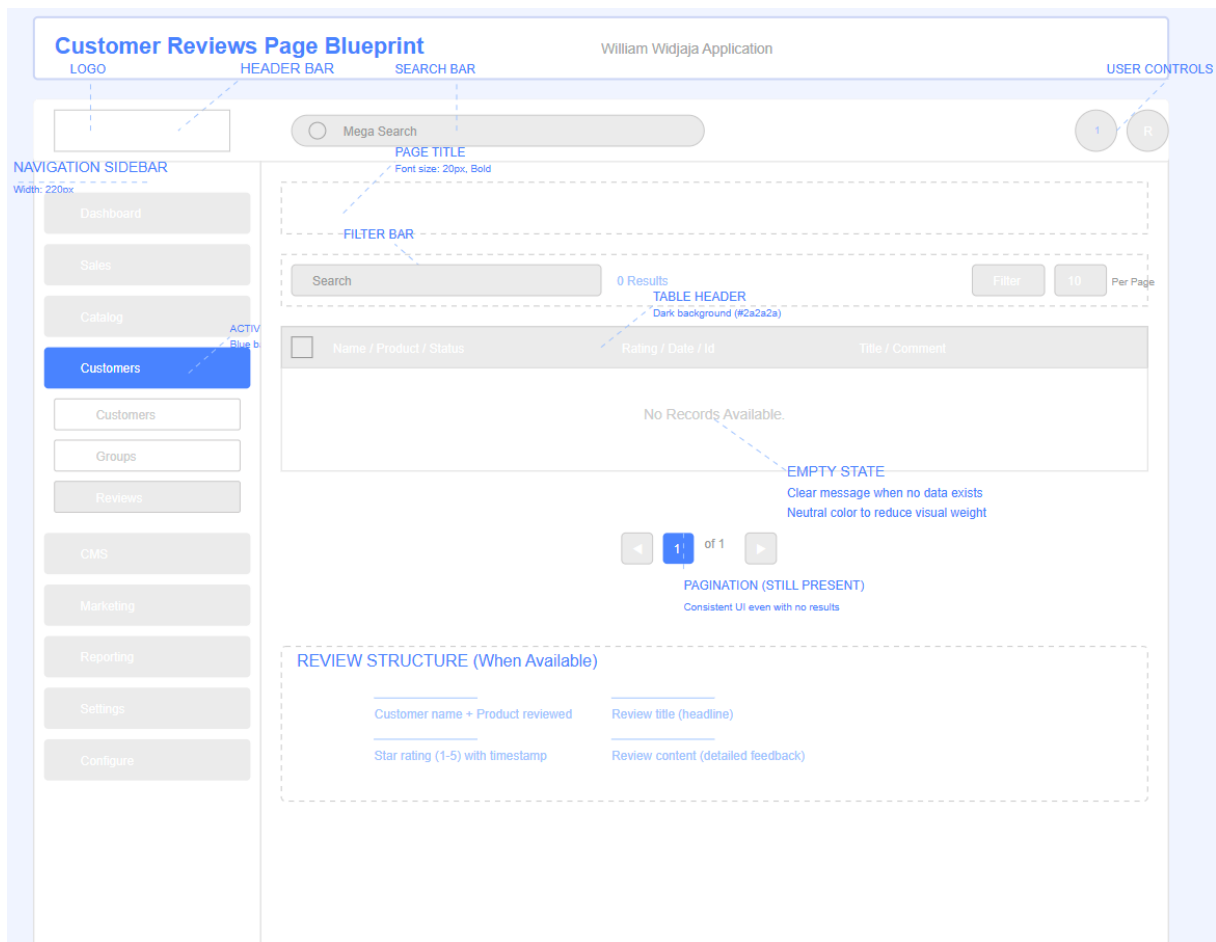
## 25. Tampilan *Create New Group*



Gambar 3. 72 *Create New Group*



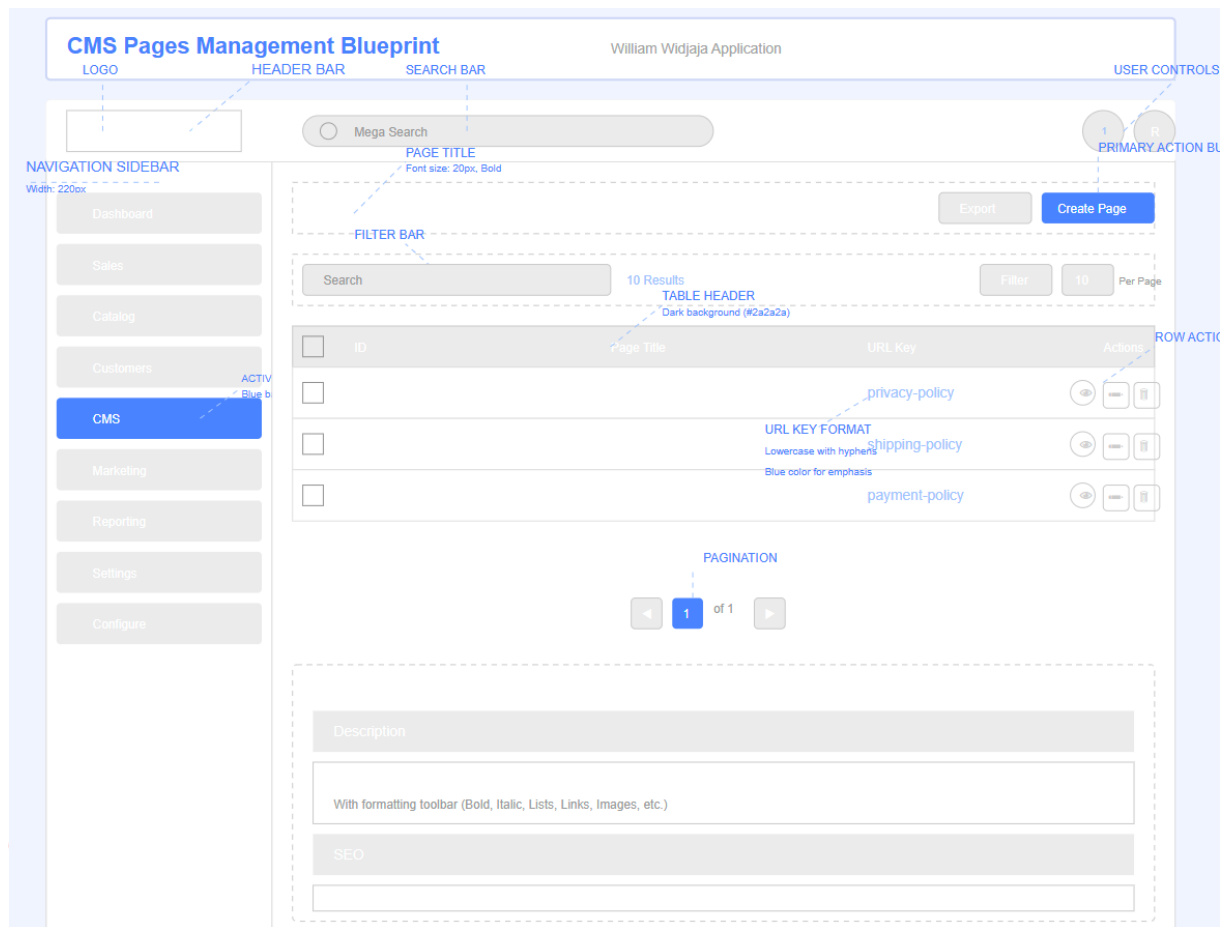
## 26. Tampilan *Reviews Admin*



Gambar 3. 73 Tampilan *Reviews Admin*



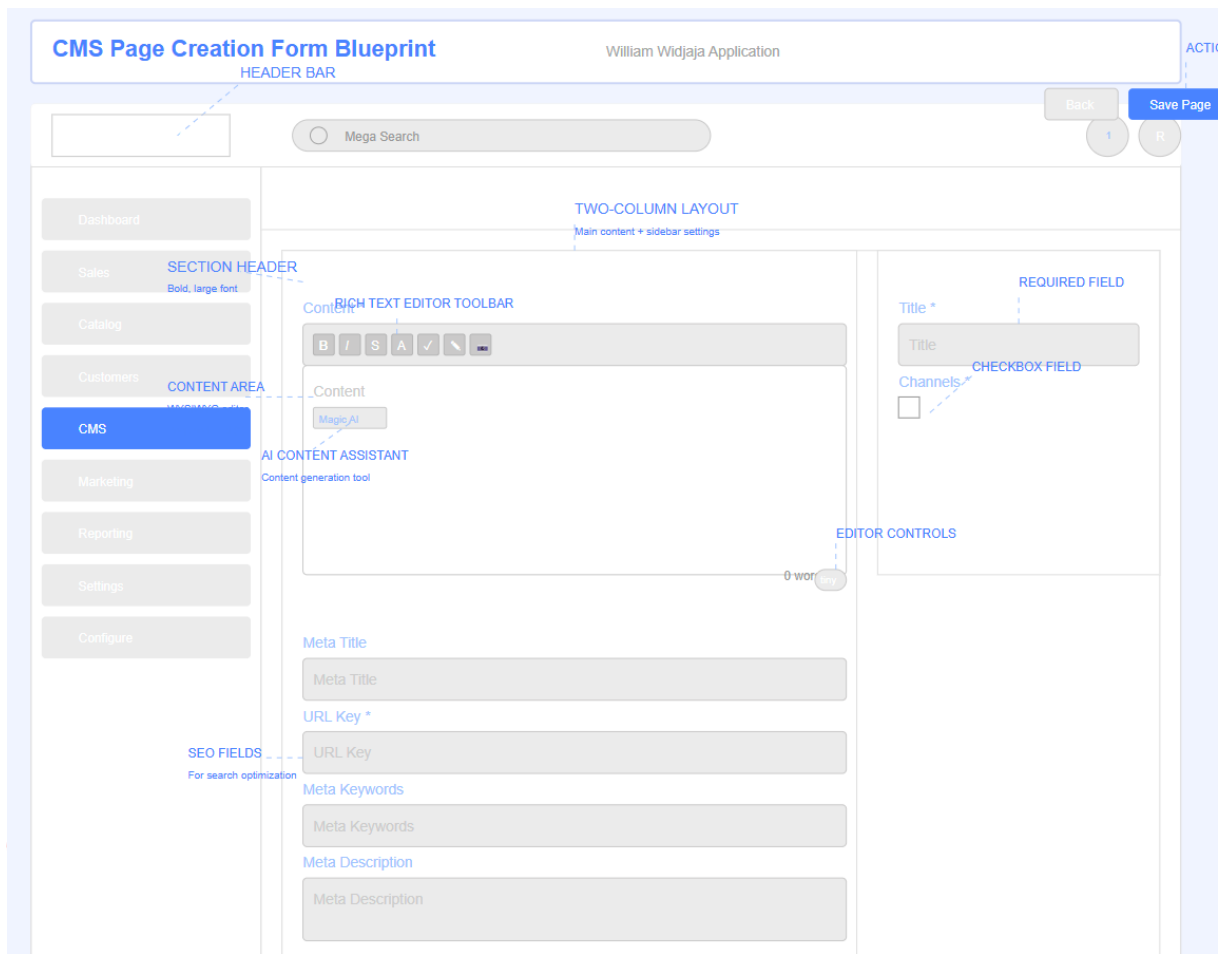
## 27. Tampilan CMS Promotions Admin



Gambar 3. 74 Tampilan CMS Promotions Admin



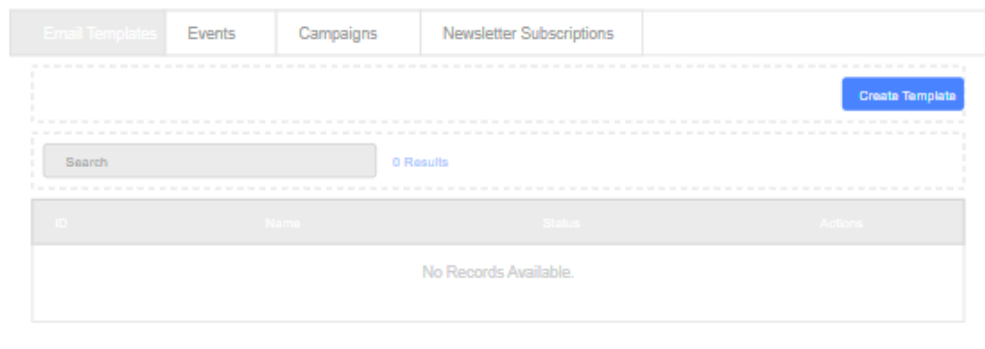
## 28. Tampilan *Create Catalog Rule Admin*



Gambar 3. 75 Tampilan *Create Catalog Rule Admin*

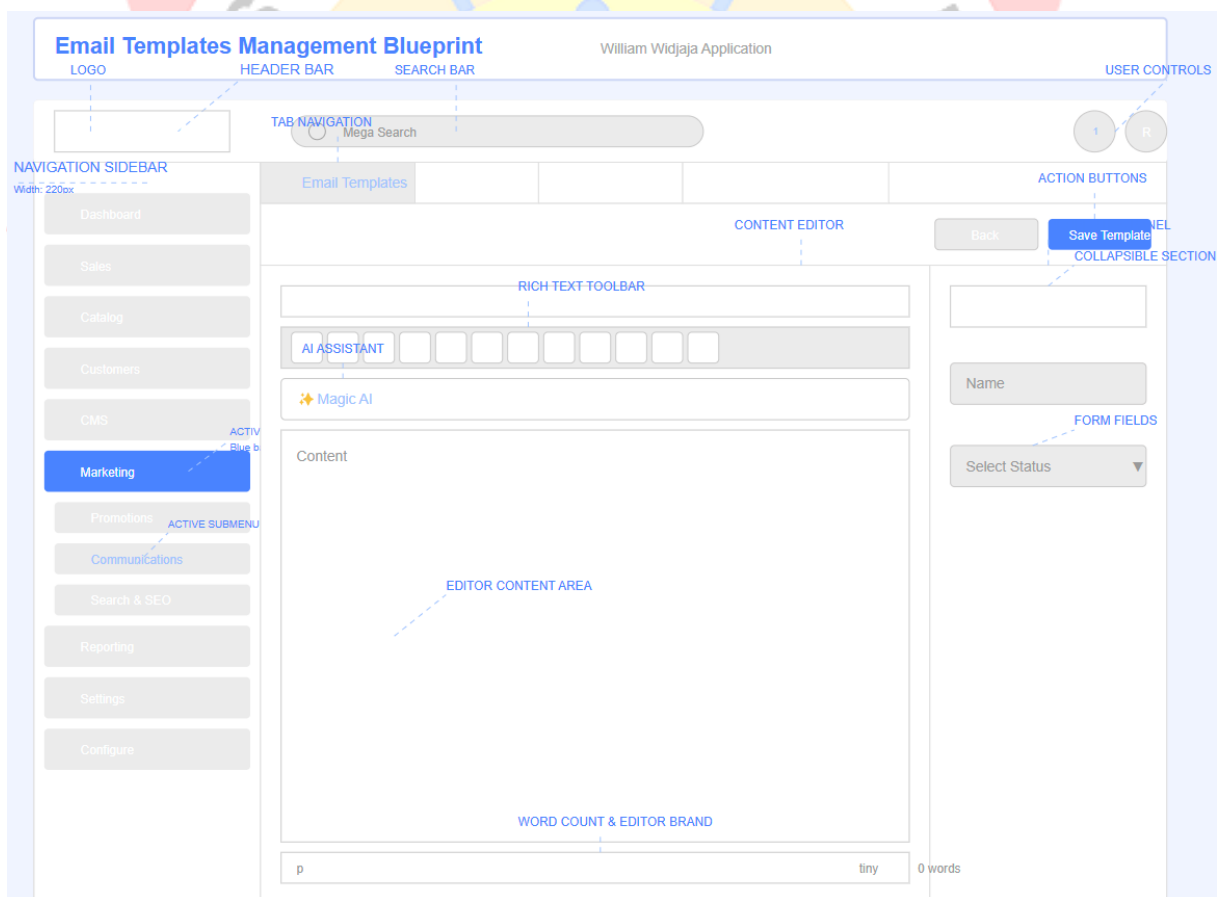


## 29. Tampilan *Communications Admin*



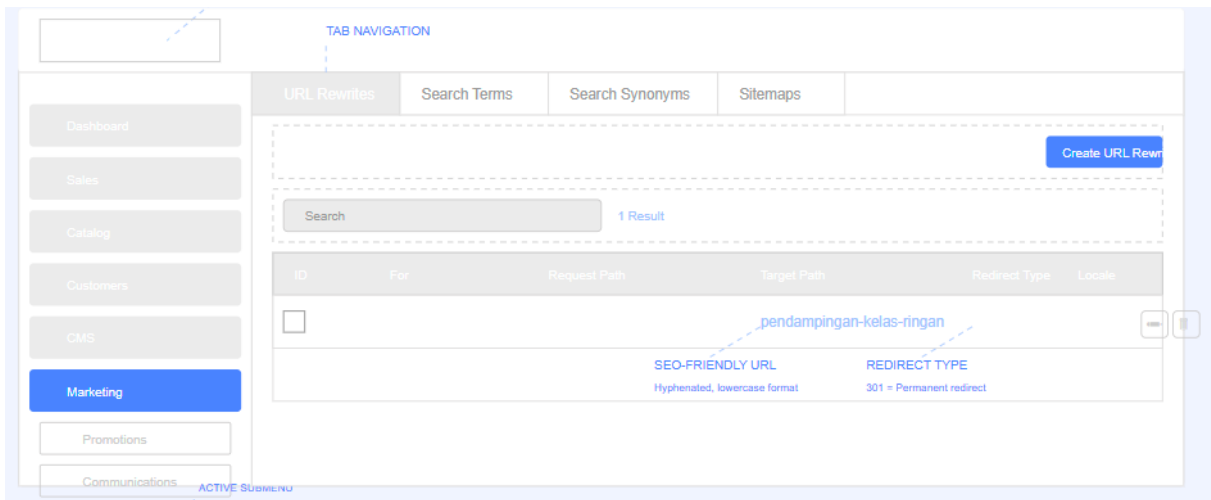
Gambar 3. 76 Tampilan *Communications Admin*

## 30. Tampilan *Create Template Admin*



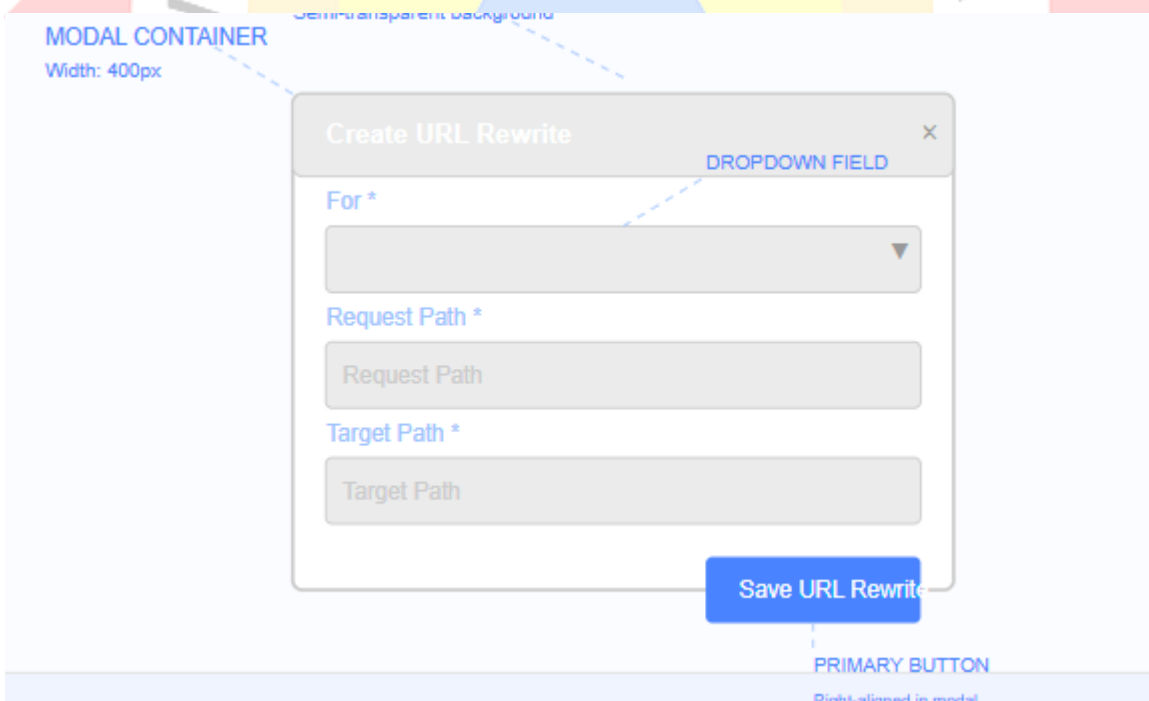
Gambar 3. 77 Tampilan *Create Template Admin*

### 31. Tampilan Search & SEO Admin



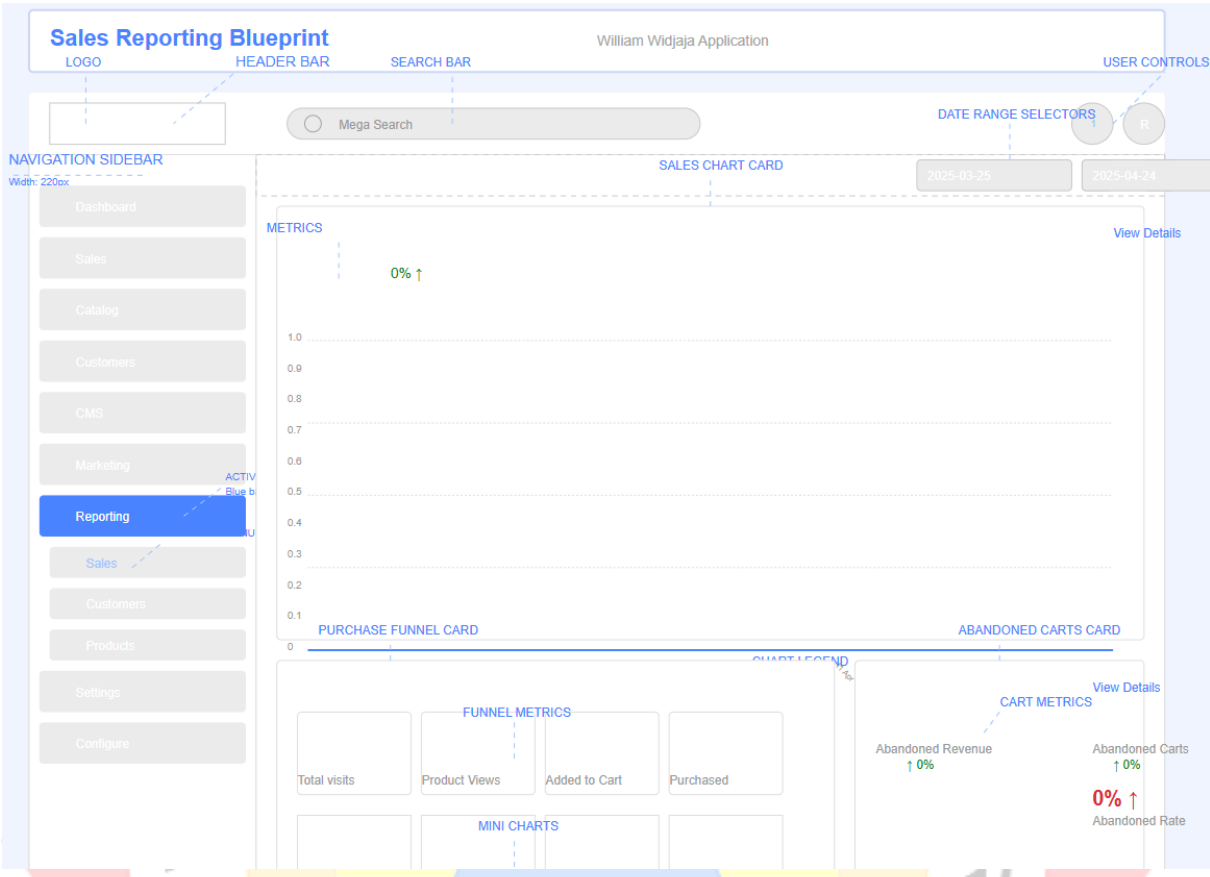
Gambar 3. 78 Tampilan Search & SEO Admin

### 32. Tampilan Create URL Rewrite



Gambar 3. 79 Tampilan Create URL Rewrite

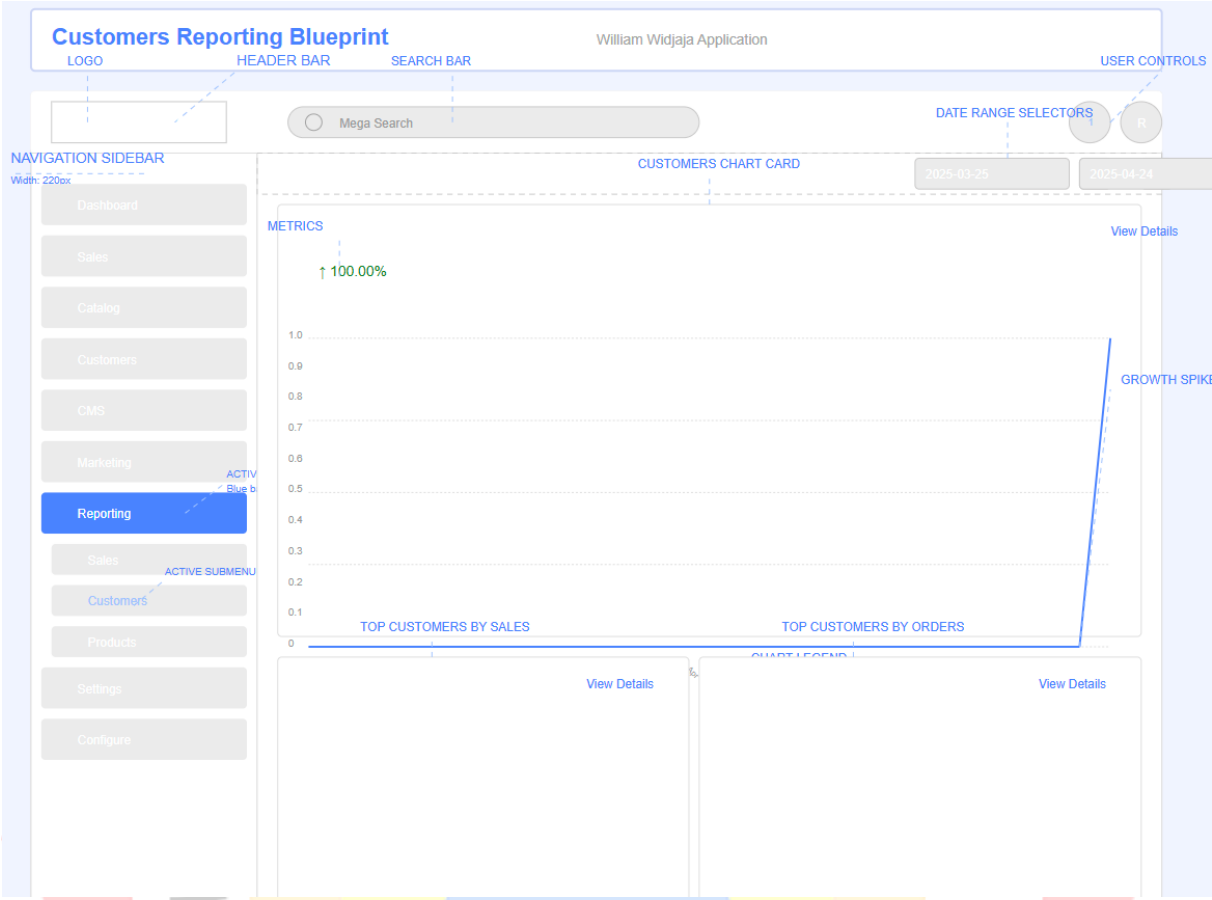
33. Tampilan *Reporting Sales Admin*



Gambar 3. 80 Tampilan *Reporting Sales Admin*



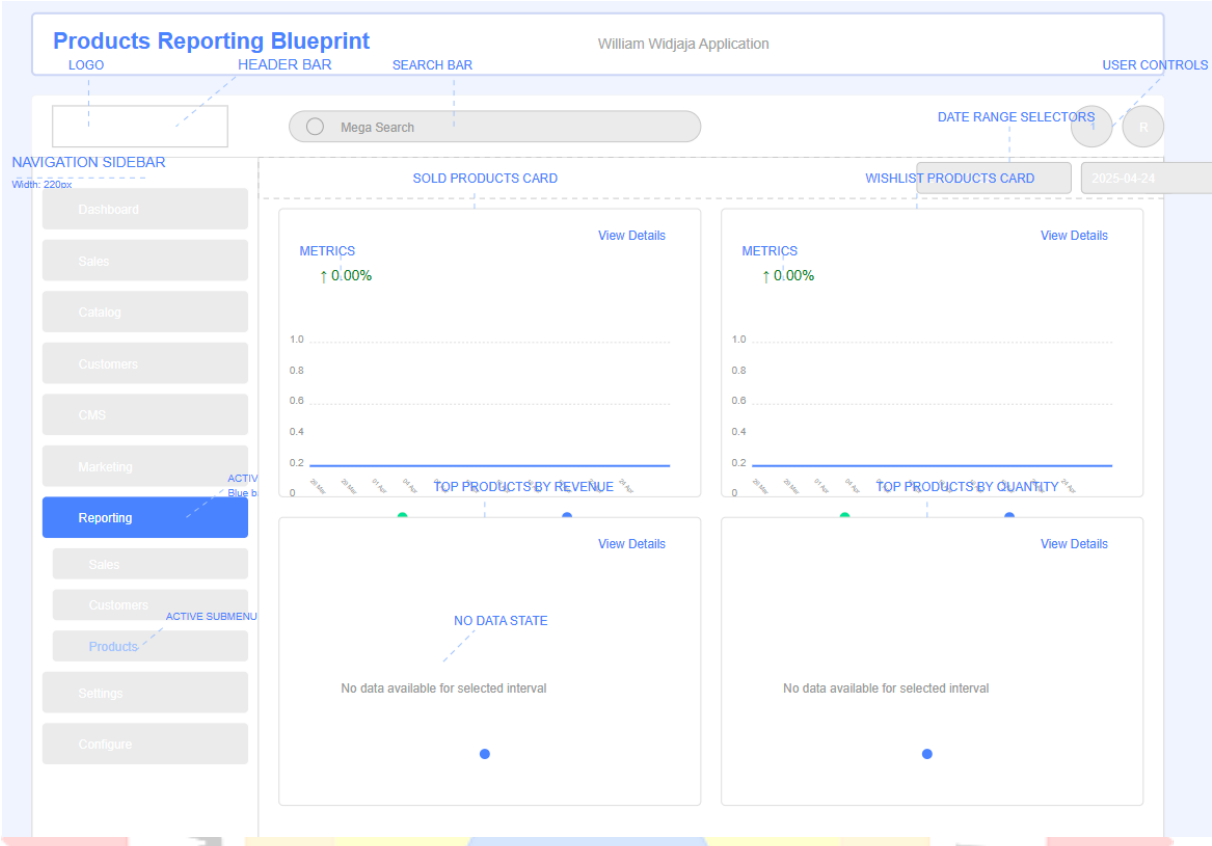
34. Tampilan *Reporting Customers*



Gambar 3. 81 Tampilan *Reporting Customers*



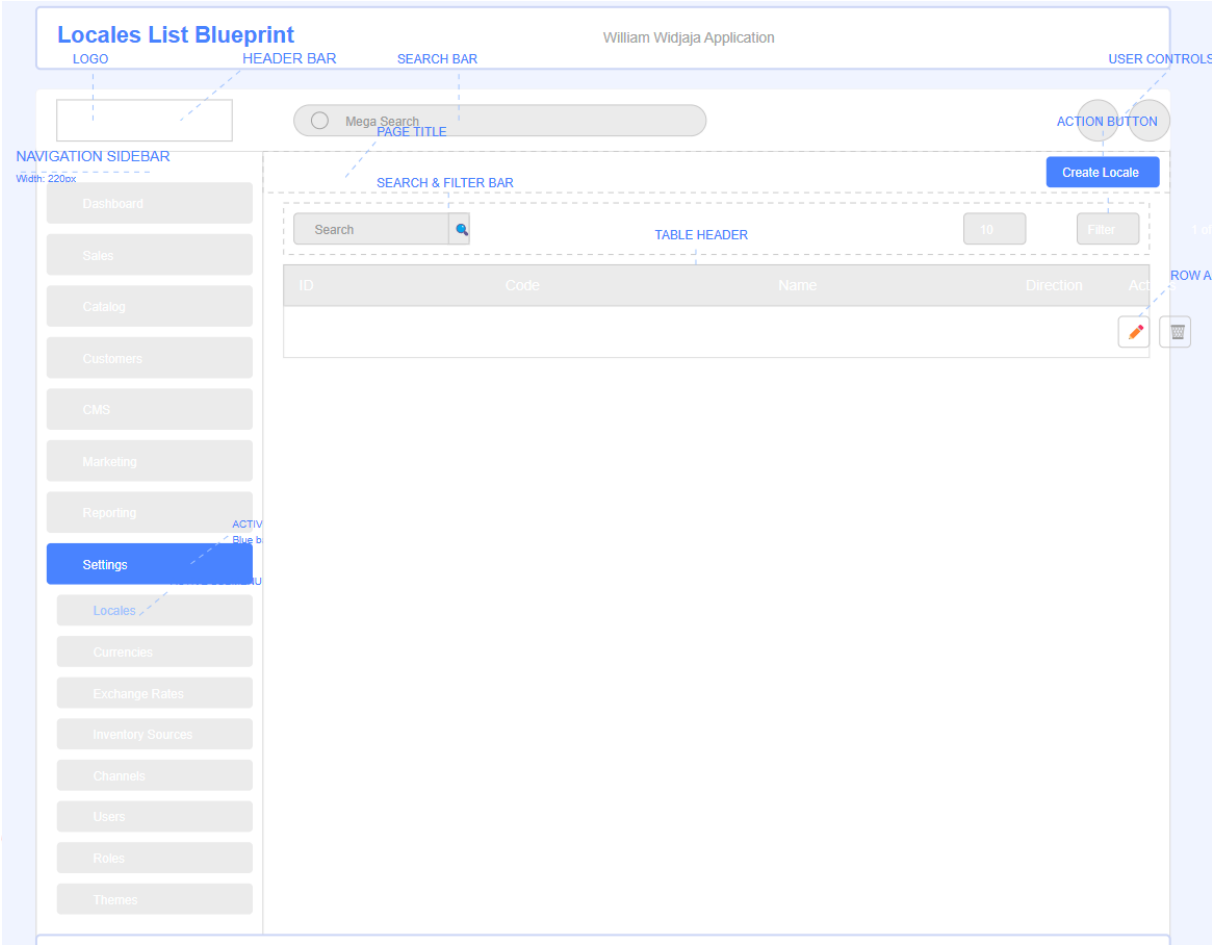
35. Tampilan *Reporting Services*



Gambar 3. 82 Tampilan *Reporting Services*



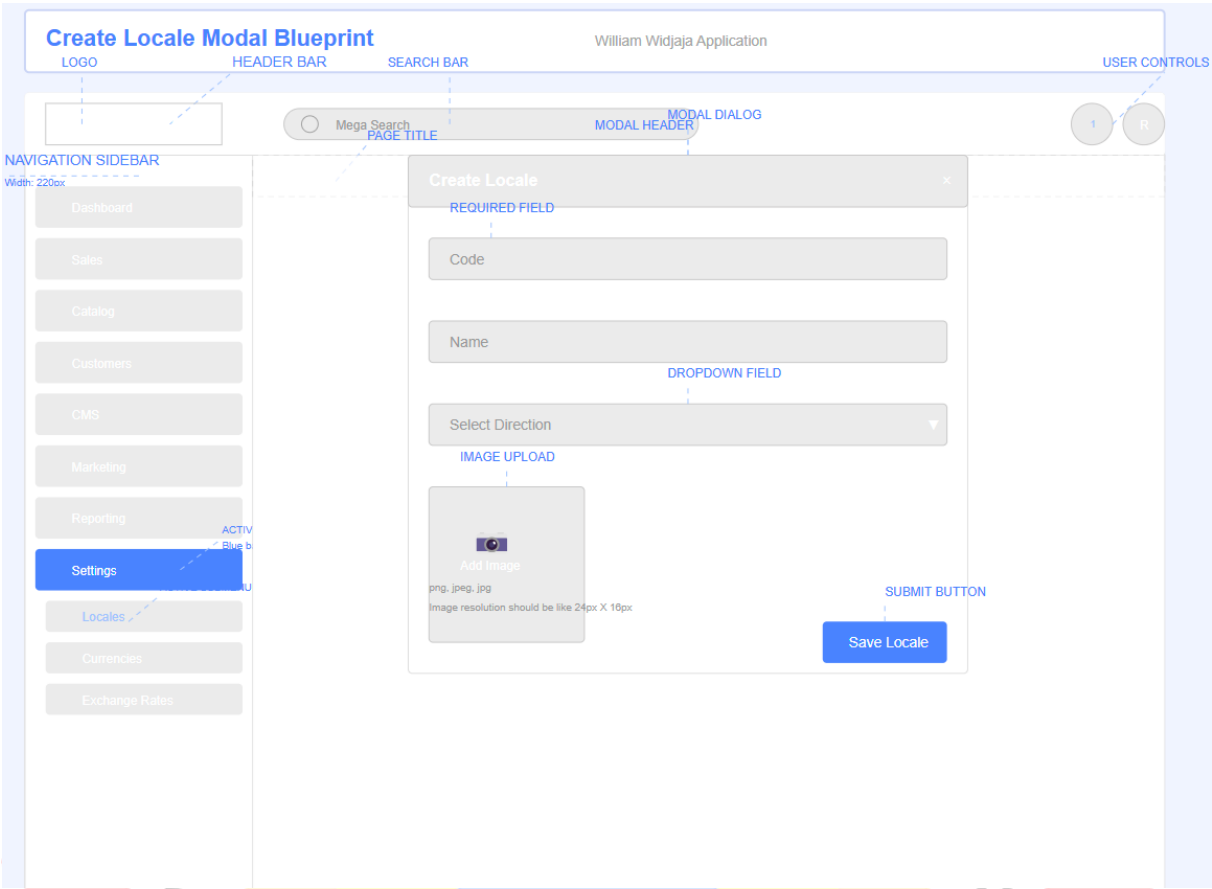
36. Tampilan *Settings Locales*



Gambar 3. 83 Tampilan *Settings Locales*

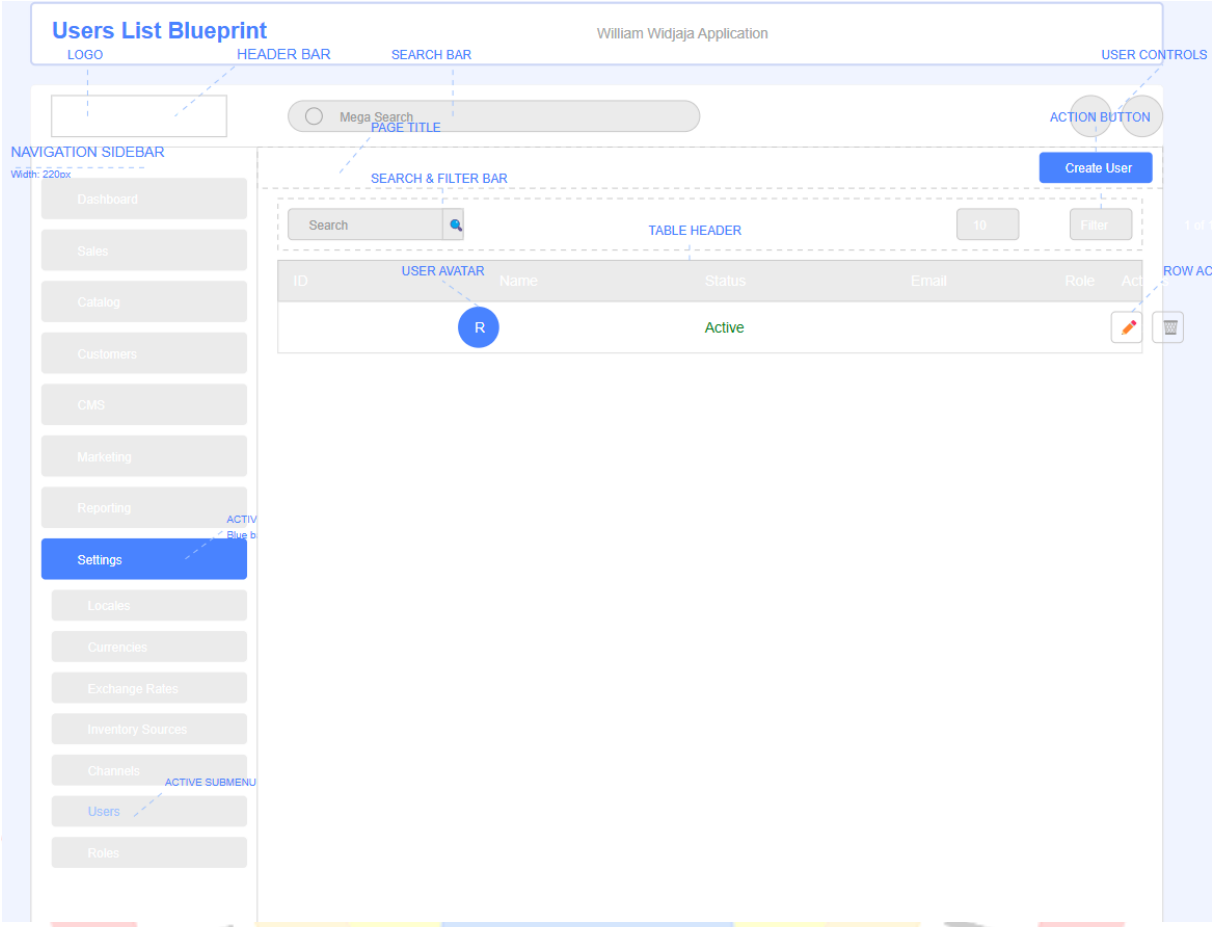


37. Tampilan *Create Locale*



Gambar 3. 84 Tampilan *Create Locale*

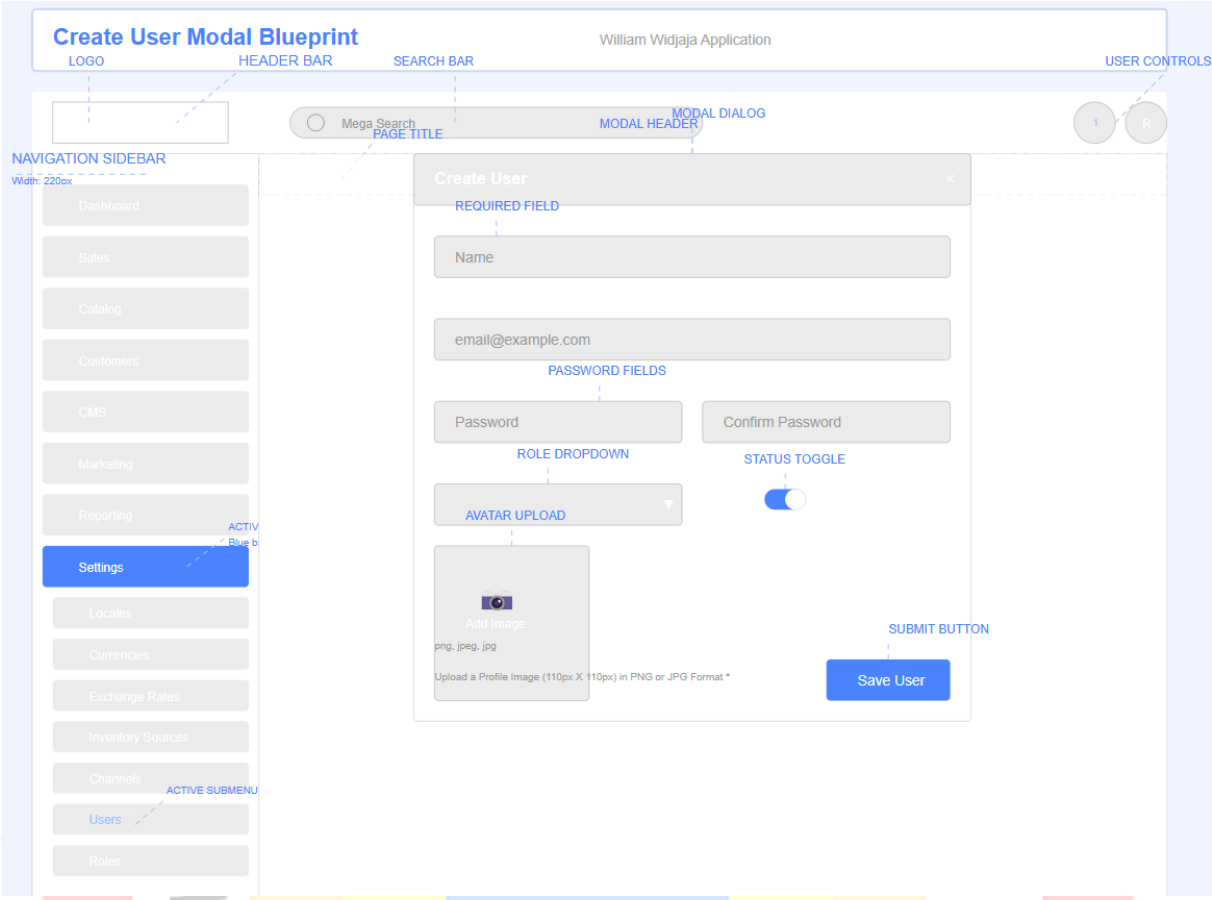
38. Tampilan *Settings Users*



Gambar 3. 85 Tampilan *Settings Users*



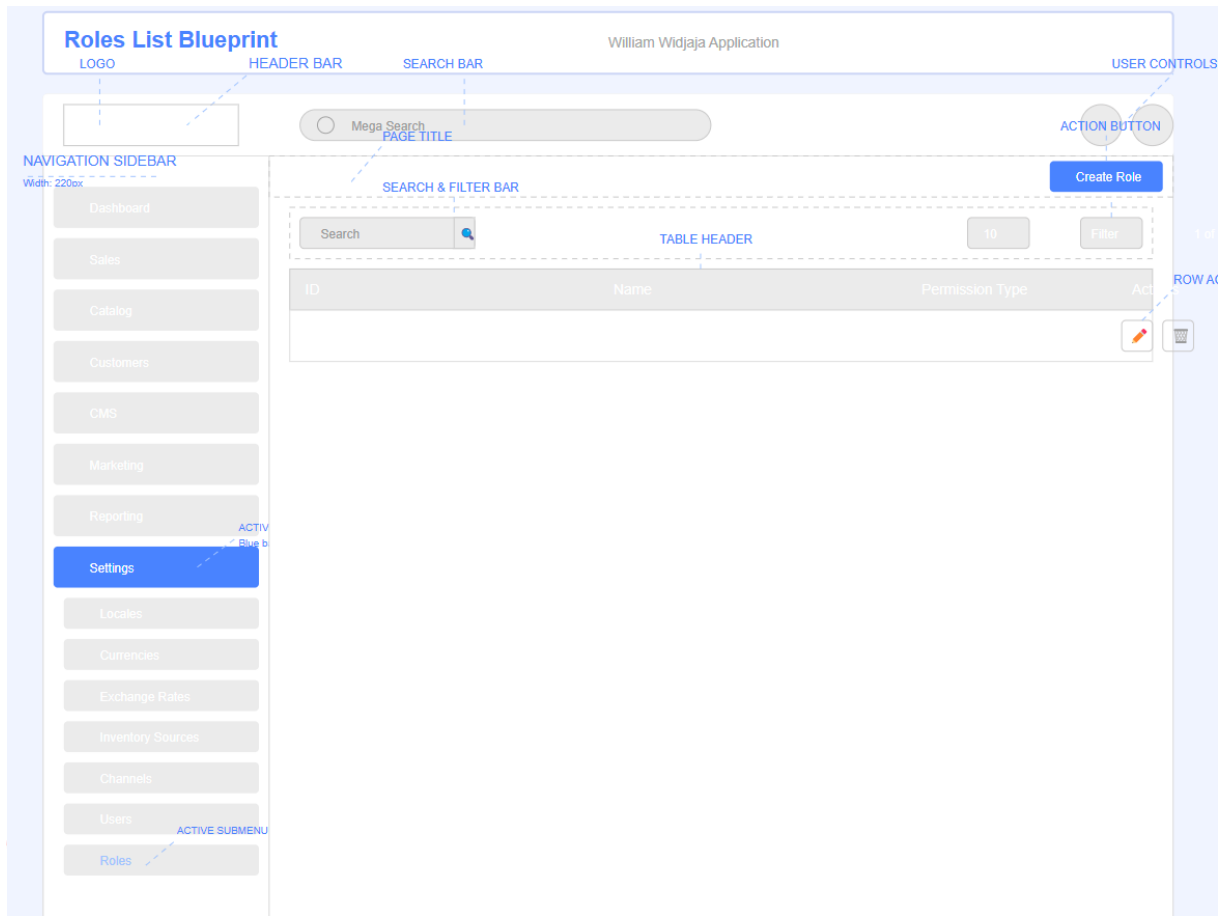
39. Tampilan *Create User*



Gambar 3. 86 Tampilan *Create User*



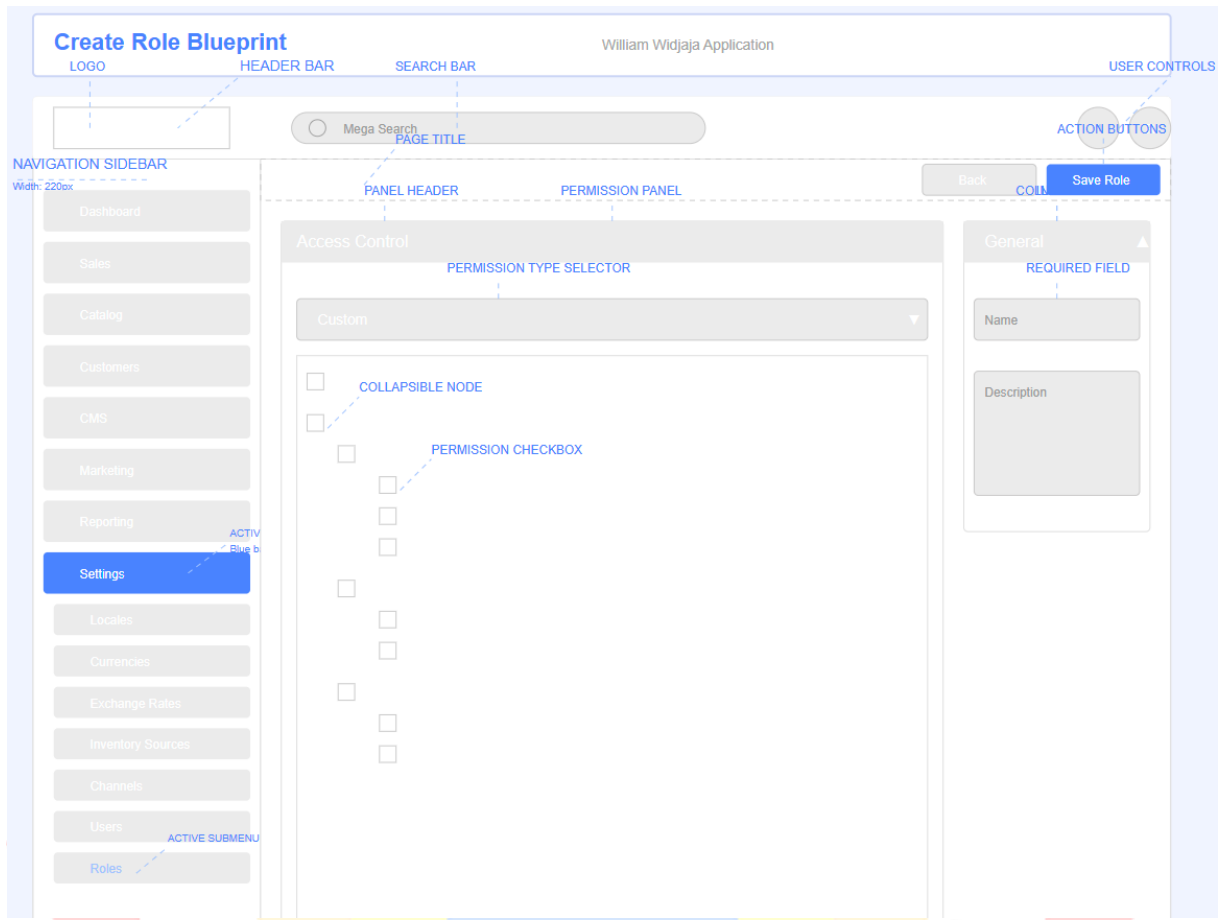
## 40. Tampilan *Settings Roles*



Gambar 3. 87 Tampilan *Settings Roles*



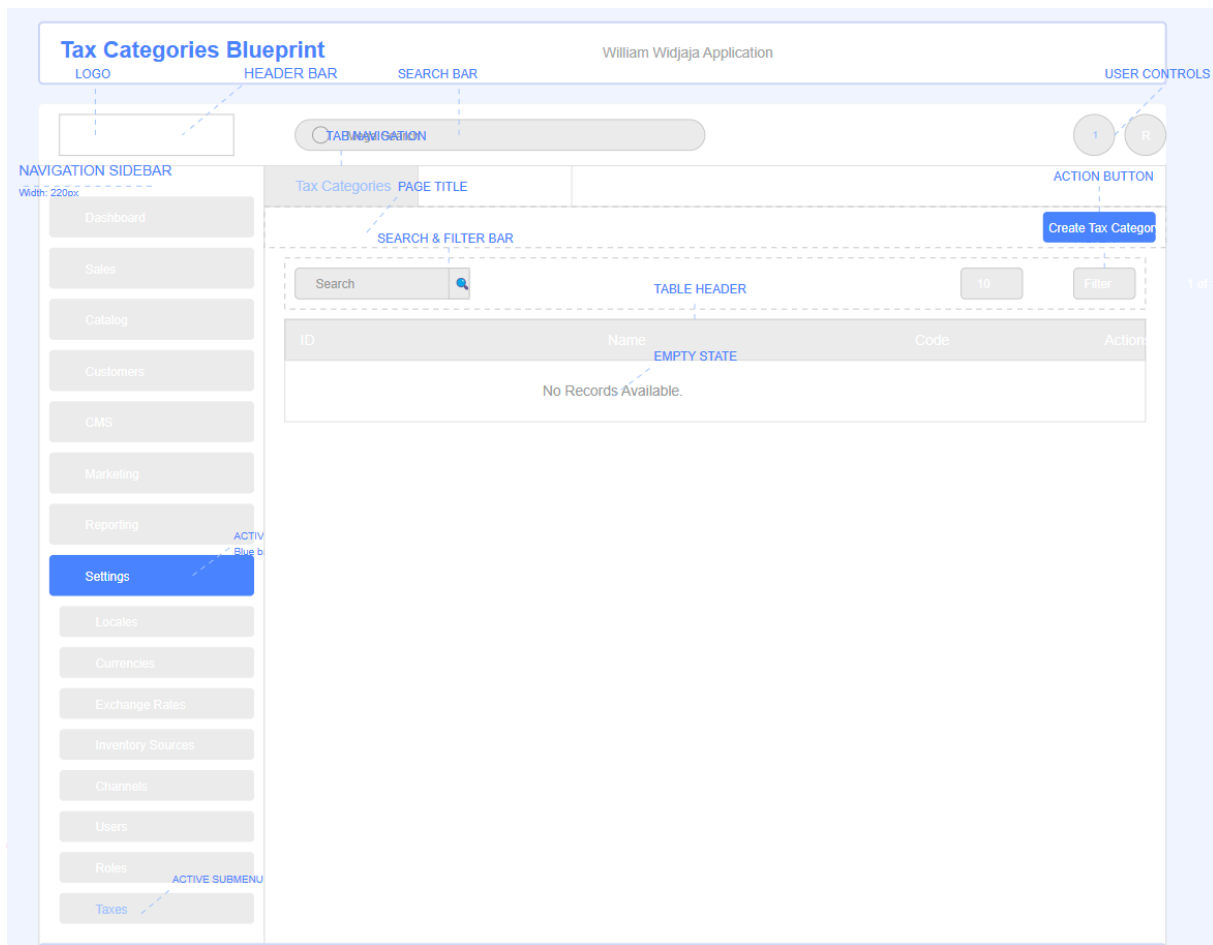
## 41. Tampilan *Create Role*



Gambar 3. 88 Tampilan *Create Role*



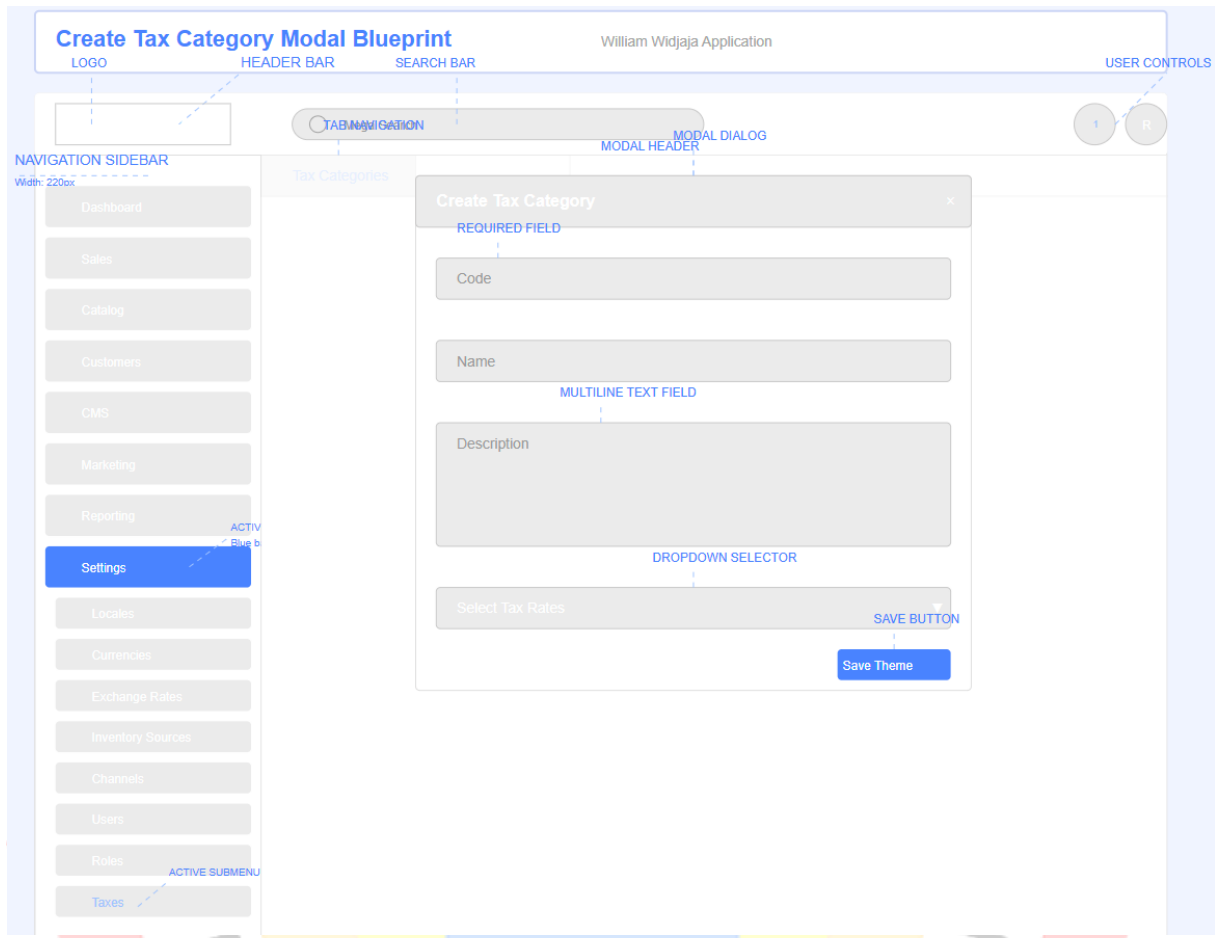
## 42. Tampilan *Tax Categories*



Gambar 3. 89 Tampilan *Tax Categories*



### 43. Tampilan *Create Tax Categories*



Gambar 3. 90 Tampilan *Create Tax Categories*

