

**KEMENTERIAN PENDIDIKAN, KEBUDAYAAN, RISET, DAN TEKNOLOGI
INSTITUT TEKNOLOGI DAN BISNIS PALCOMTECH**

SKRIPSI

***E-Catalog Dapunta Inc Menggunakan
Metode Rational Unified Process
(RUP)***



Diajukan Oleh:

- 1. GUSTINA DWI PRATIWI / 021200071**
- 2. ONI LESTARI / 02120004**

**Untuk Memenuhi Sebagian Dari Syarat
Mencapai Gelar Sarjana Komputer**

**PALEMBANG
2024**

**KEMENTERIAN PENDIDIKAN, KEBUDAYAAN, RISET, DAN TEKNOLOGI
INSTITUT TEKNOLOGI DAN BISNIS PALCOMTECH**

SKRIPSI

***E-Catalog Dapunta Inc Menggunakan
Metode *Rational Unified Process*
(RUP)***



Diajukan Oleh:

- 1. GUSTINA DWI PRATIWI / 021200071**
- 2. ONI LESTARI / 02120004**

**Untuk Memenuhi Sebagian Dari Syarat
Mencapai Gelar Sarjana Komputer**

**PALEMBANG
2024**

KEMENTERIAN PENDIDIKAN, KEBUDAYAAN, RISET DAN TEKNOLOGI
INSTITUT TEKNOLOGI DAN BISNIS PALCOMTECH

HALAMAN PENGESAHAN PEMBIMBING SKRIPSI

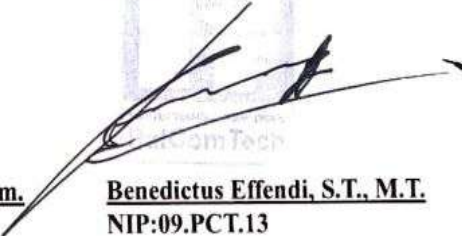
NAMA/NPM : 1. GUSTINA DWI PRATIWI/021200071
2. ONI LESTARI/021200044
PROGRAM STUDI : SISTEM INFORMASI
JENJANG PENDIDIKAN : STRATA SATU
JUDUL : *E-CATALOG DAPUNTA INC*
MENGUNAKAN METODE *RATIONAL*
UNIFIED PROCESS (RUP)

Tanggal: 19 juni 2024
Pembimbing

Mengetahui,
Rektor



Meidyan Permata Putri, S.Kom., M.Kom.
NIDN : 0204058604



Benedictus Effendi, S.T., M.T.
NIP:09.PCT.13

**KEMENTERIAN PENDIDIKAN, KEBUDAYAAN, RISET DAN TEKNOLOGI
INSTITUT TEKNOLOGI DAN BISNIS PALCOMTECH**

HALAMAN PENGESAHAN PENGUJI SKRIPSI

NAMA/NPM : 1. GUSTINA DWI PRATIWI/021200071
2. ONI LESTARI/021200044
PROGRAM STUDI : SISTEM INFORMASI
JENJANG PENDIDIKAN : STRATA SATU
JUDUL : *E-CATALOG DAPUNTA INC*
MENGGUNAKAN METODE RATIONAL
UNIFIED PROCESS (RUP)

Tanggal : 05 Agustus 2024

Penguji 1



Yesi Sriveni, S.Kom., M.Kom.

NIDN : 0218038904

Tanggal : 02 Agustus 2024

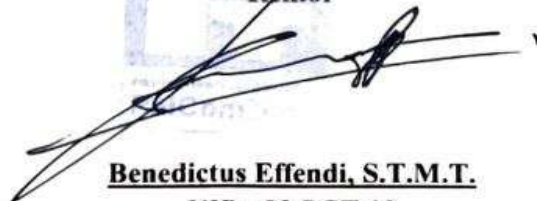
Penguji 2



Jaka Purnama, S.Kom., M.Kom.

NIDN : 0219089401

**Menyetujui,
Rektor**



Benedictus Effendi, S.T.M.T.

NIP : 09.PCT.13

MOTTO:

“Allah tidak membebani seseorang melainkan dengan kesanggupannya”.

(QS. Al-Baqarah : 286)

Kupersembahkan Kepada:

1. Allah Subhanahu wa Ta'ala
2. Kedua Orang Tua Tercinta
3. Teman-Teman Seperjuangan
4. Para pendidik yang kami hormati

KATA PENGANTAR

Puji syukur kepada Allah Subhanahu wa Ta'ala., yang telah memberikan rahmat dan karunia-Nya sehingga penulis dapat menyelesaikan laporan skripsi yang berjudul “*E-Catalog* Dapunta Inc Menggunakan Metode *Rational Unified Process* (RUP)”.

Laporan ini merupakan salah satu syarat untuk menyelesaikan Program Studi S1 Sistem Informasi Institut Teknologi dan Bisnis PalComTech Palembang. Sebagai rasa syukur dan hormat, melalui kesempatan ini penulis menyampaikan ucapan terimakasih kepada:

1. Kepada orang tua yang tercinta, teman dan sahabat yang terkasih.
2. Rektor Institut Teknologi dan Bisnis PalComTech Bapak Benedictus Effendi, S.T.,M.T.
3. Wakil Rektor I Ibu Adelin, S.T.,M.Kom.
4. Ketua Program Studi Sistem Informasi Program Sarjana Ibu Dini Hari Pertiwi, S.Kom., M.Kom.
5. Dosen Pembimbing Meidyan Permata Putri, S.Kom., M.Kom.

Penulis menyadari bahwa penulisan laporan skripsi ini masih sangat jauh dari kesempurnaan dan masih banyak kekurangan, untuk itu kritik dan saran yang membangun sangat diharapkan untuk penyempurnaan skripsi ini kedepan.

Palembang, 15 Juli 2024

Penulis

DAFTAR ISI

HALAMAN JUDUL.....	i
HALAMAN PENGESAHAN PEMBIMBING.....	Error! Bookmark not defined.
HALAMAN PENGESAHAN PENGUJI	Error! Bookmark not defined.
HALAMAN MOTTO DAN PERSEMBAHAN	x
KATAPENGANTAR.....	v
DAFTAR ISI.....	vi
DAFTAR GAMBAR.....	x
DAFTAR TABEL.....	xi
DAFTAR LAMPIRAN.....	xii
ABSTRACT	xi
ABSTRAK	xii
BAB I PENDAHULUAN	
1.1 Latar Belakang	13
1.2 Perumusan Masalah	3
1.3 Batasan Masalah.....	3
1.4 Tujuan Penelitian.....	4
1.5 Manfaat Penelitian	4
1.6 Sistem Penulisan	4
BAB II GAMBARAN UMUM PERUSAHAAN	
2.1 Profil Perusahaan	6
2.1.1 Sejarah Perusahaan.....	6
2.2 Visi dan Misi	6
2.2.1 Visi	6
2.2.2 Misi	6
2.3 Struktur Organisasi.....	7
2.4 Tugas Wewenang.....	8

BAB III TINJAUAN PUSTAKA

3.1 E-Catalog.....	21
3.2 Rational Unified Process.....	21
3.3 Unified Modelling Language (UML).....	22
3.4 Use Case Diagram.....	22
3.5 Activity Diagram.....	24
3.6 Class Diagram.....	25
3.7 Sequence Diagram.....	26
3.8 Hasil Penelitian Terdahulu.....	28

BAB IV METODE PENELITIAN

4.1 Lokasi dan Waktu Penelitian.....	21
4.1.1 Lokasi Penelitian.....	21
4.1.2 Waktu Penelitian.....	21
4.2 Jenis Data.....	22
4.2.1 Data Primer.....	22
4.2.2 Data Sekunder.....	22
4.3 Teknik Pengumpulan Data.....	23
4.3.1 Wawancara.....	23
4.3.2 Observasi.....	23
4.3.3 Studi Pustaka.....	24
4.3.4 Dokumentasi.....	24
4.4 Alat dan Teknik Pengembangan Sistem.....	24
4.4.1 Alat Pengembangan Sistem.....	24
4.4.2 Teknik Pengembangan Sistem.....	25
4.5 Alat dan Teknik Pengujian.....	25
4.5.1 White Box Testing.....	25

BAB V HASIL DAN PEMBAHASAN

5.1 Hasil.....	26
5.1.1 Inception.....	26
5.1.2 Elaboration.....	28
5.1.3 Construction.....	80

5.1.4 Pengujian White Box Testing.....	89
5.1.5 Transition	99
5.2 Pembahasan.....	99
BAB VI PENUTUP	
6.1 Kesimpulan	98
6.2 Saran.....	98
DAFTAR PUSTAKA.....	xv
HALAMAN LAMPIRAN.....	xvi

DAFTAR GAMBAR

Gambar 2. 1 Struktur Orgnisasi DAPUNTA INC	7
Gambar 3. 1 Kerangka Pemikiran	31
Gambar 5. 1 Alur yang berjalan pada Dapunta Inc.....	27
Gambar 5. 2 <i>Use Case Diagram</i> yang diusulkan	29
Gambar 5. 3 <i>Activity Diagram</i> Login.....	31
Gambar 5. 4 <i>Activity Diagram</i> Konsumen.....	32
Gambar 5. 5 <i>Activity Diagram</i> Konsumen.....	33
Gambar 5. 6 <i>Activity Diagram</i> Konsumen.....	34
Gambar 5. 7 <i>Activity Diagram</i> Owner	36
Gambar 5. 8 <i>Activity Diagram</i> Owner	37
Gambar 5. 9 <i>Activity Diagram</i> Owner	38
Gambar 5. 10 <i>Activity Diagram</i> Admin	39
Gambar 5. 11 <i>Activity Diagram</i> Admin	40
Gambar 5. 12 <i>Activity Diagram</i> Admin	41
Gambar 5. 13 <i>Activity Diagram</i> Admin.....	42
Gambar 5. 14 <i>Activity Diagram</i> Admin	43
Gambar 5. 15 <i>Activity Diagram</i> Admin	44
Gambar 5. 16 <i>Activity Diagram</i> Logout.....	45
Gambar 5. 17 <i>Class Diagram</i> yang Diusulkan	47
Gambar 5. 18 <i>Sequence Diagram</i> Kelola Data Produk.....	48
Gambar 5. 19 <i>Sequence Diagram</i> Edit Data Kategori Produk.....	49
Gambar 5. 20 <i>Sequence Diagram</i> Hapus Data Kategori Produk	49
Gambar 5. 21 <i>Sequence Diagram</i> Tambah Data Kategori Produk	50
Gambar 5. 22 <i>Gambar Sequence Diagram</i> Edit Banner Produk.....	51
Gambar 5. 23 <i>Sequence Diagram</i> Edit Banner Produk	51
Gambar 5. 24 <i>Sequence Diagram</i> Tambah Banner	52
Gambar 5. 25 <i>Sequence Diagram</i> Edit Data Produk	52
Gambar 5. 26 <i>Sequence Diagram</i> Hapus Data Produk	53
Gambar 5. 27 <i>Sequence Diagram</i> Tambah Data Produk	53

Gambar 5. 28. <i>Sequence Diagram</i> Edit Data Merk	54
Gambar 5. 29. <i>Sequence Diagram</i> Hapus Data Merk.....	54
Gambar 5. 30. <i>Sequence Diagram</i> Tambah Data Merk	55
Gambar 5. 31. <i>Sequence Diagram</i> Perbarui Status Data Order	55
Gambar 5. 32. <i>Sequence Diagram</i> Edit Data User.....	56
Gambar 5. 33. <i>Sequence Diagram</i> Tambah Data User	56
Gambar 5. 34. <i>Sequence Diagram</i> Konsumen Checkout.....	57
Gambar 5. 35. <i>Sequence Diagram</i> Tambah Keranjang.....	58
Gambar 5. 36. <i>Sequence Diagram</i> Konsumen Tambah Alamat.....	58
Gambar 5. 37. Desain Halaman Login Admin.....	71
Gambar 5. 38. Desain Halaman Dashboard Admin	72
Gambar 5. 39. Desain Halaman Packaging.....	73
Gambar 5. 40. Desain Halaman Pesanan Dikirim.....	73
Gambar 5. 41. Desain <i>Interface</i> Data Produk	74
Gambar 5. 42. Desain <i>Interface</i> Tambah Produk	74
Gambar 5. 43. Desain <i>Interface</i> Pesan	75
Gambar 5. 44. Desain <i>Interface</i> Daftar Pelanggan	75
Gambar 5. 45. Desain <i>Interface</i> Review Pelanggan	76
Gambar 5. 46. Desain <i>Interface</i> Register Pelanggan	76
Gambar 5. 47. Desain <i>Interface</i> Profil Pelanggan.....	77
Gambar 5. 48. Desain <i>Interface</i> Keranjang Pelanggan	78
Gambar 5. 49. Desain <i>Interface</i> Pesanan Lacak Pesanan	78
Gambar 5. 50. Desain <i>Interface</i> Login Owner	79
Gambar 5. 51. Desain <i>Interface</i> Dashboard Owner	80
Gambar 5. 52. Tampilan <i>Interface</i> Halaman Login	81
Gambar 5. 53. Tampilan <i>Interface</i> Halaman Dasbord	81
Gambar 5. 54. Tampilan <i>Interface</i> Halaman Packaging	82
Gambar 5. 55. Tampilan <i>Interface</i> Halaman Pesanan Dikirim	83
Gambar 5. 56. Tampilan <i>Interface</i> Data Produk	83
Gambar 5. 57. Tampilan <i>Interface</i> Tambah Produk	84
Gambar 5. 58. Tampilan <i>Interface</i> Pesan	84

Gambar 5. 59. Tampilan <i>Interface</i> Daftar Pelanggan	85
Gambar 5. 60. Tampilan <i>Interface</i> Review Pelanggan	85
Gambar 5. 61. Tampilan <i>Interface</i> Login Pelanggan	86
Gambar 5. 62. Tampilan <i>Interface</i> Profil Pelanggan.....	86
Gambar 5. 63. Tampilan <i>Interface</i> keranjang Pelanggan	87
Gambar 5. 64. Tampilan <i>Interface</i> Lacak Pesanan	88
Gambar 5. 65. Tampilan <i>Interface</i> Login Owner	88
Gambar 5. 66. Tampilan Dashboard Owner.....	89

DAFTAR TABEL

Tabel 3.1 Tabel <i>Use Case Diagram</i>	24
Tabel 3.2 Tabel <i>Activity Diagram</i>	25
Tabel 3.3 <i>Class Diagram</i>	25
Tabel 3.4 Simbol-Simbol <i>Sequence Diagram</i>	28
Tabel 3.5 Penelitian Terdahulu	29
Tabel 4.1 Jadwal Penelitian.....	32
Tabel 5.1 Tabel Admin	60
Tabel 5.2 Tabel Admin Roles	61
Tabel 5.3 Tabel Atribut	61
Tabel 5.4 Tabel <i>Banner</i>	62
Tabel 5.5 Tabel <i>Cart</i>	63
Tabel 5.6 Tabel <i>Order</i>	64
Tabel 5.7 Tabel <i>Order Detail</i>	64
Tabel 5.8 Tabel Produk	66
Tabel 5.9 Tabel Stok Produk	67
Tabel 5.10 Tabel Tag Produk.....	68
Tabel 5.11 Tabel Rview Produk	68
Tabel 5.12 Tabel Tag	69
Tabel 5.13 Tabel Transaksi	70
Tabel 5.14 Tabel <i>Wishlist</i>	70
Tabel 5.15 Tabel Pengujian White Box Testing	91

DAFTAR LAMPIRAN

1. Lampiran 1. *Form* Persetujuan Topik dan Judul (Fotokopi)
2. Lampiran 2. Surat Balasan dari Perusahaan (Fotokopi)
3. Lampiran 3. *Form* Konsultasi (Fotokopi)
4. Lampiran 4. Surat Pernyataan (Fotokopi)
5. Lampiran 5. *Form* Revisi Ujian Pra Sidang(Fotokopi)
6. Lampiran 6. *Form* Revisi Ujian Kompre (asli)
7. Lampiran 7. *Listing Code* (Fotokopi)

ABSTRACT

GUSTINA DWI PRATIWI AND ONI LESTARI.

Dapunta Inc E-catalog Uses the Rational Unified Process (RUP) Method

Dapunta has a problem, namely that Dapunta sells graphic design services only using Instagram, and has not used an e-catalog type website, causing several problems that occur, namely transactions, and customer data and turnover recaps are not captured properly because they are separated outside of Instagram. making it difficult for the Dapunta admin to combine the data so that the population experiences data collection errors at Dapunta Inc. The research method used is the RUP method in which software development is carried out repeatedly (iteratively). The system development process includes initial stages, elaboration, construction and transition. System testing is carried out using the white box method using the base path testing technique of analyzing and examining the internal structure and program code to find each base path that can be executed within a certain time. This system is equipped with features such as register, login, Admin, Customer, Owner, Orders, Products, Messages, Orders, Profile and Track Services.

Keywords: Sales System, Rup Method, White Box Testing

ABSTRAK

GUSTINA DWI PRATIWI DAN ONI LESTARI.

E-catalog Dapunta Inc Menggunakan Metode Rational Unified Process (RUP)

Dapunta memiliki sebuah permasalahan, yakni Dapunta melakukan penjualan jasa desain grafis hanya dengan menggunakan instagram, dan belum menggunakan sebuah website jenis e-catalog, menyebabkan beberapa masalah yang terjadi yaitu transaksi, dan data pelanggan dan rekapan omset tidak terekap dengan baik karena terpisah diluar instagram hal ini membuat admin dapunta kesulitan dalam menggabungkan data-data tersebut sehingga berpopulasi mengalami kesalahan pendataan di Dapunta inc. Metode penelitian yang digunakan adalah metode RUP yang pengembangan perangkat lunak dilakukan secara berulang-ulang (iterative). Proses pengembang sistem meliputi tahapan inception, elaboration, construction dan transition. Pengujian sistem dilakukan dengan menggunakan metode white box dengan menggunakan teknik basis path testing menganalisis dan meneliti struktur internal dan kode program untuk menemukan setiap jalur basis yang dapat dieksekusi dalam waktu tertentu. Sistem ini dilengkapi dengan fitur-fitur seperti register, login, Admin, Pelanggan, Owner, Pesanan, Produk, Pesan, Order, Profil dan Lacak Pelayanan.

Kata Kunci : Sistem Penjualan, Metode Rup, Pengujian White box Testing

BAB I

PENDAHULUAN

1.1 Latar Belakang

Kemajuan teknologi telah membawa perubahan signifikan dalam berbagai aspek kehidupan manusia. Dari revolusi industri hingga era digital, teknologi telah mengubah cara kita bekerja, berkomunikasi, dan menjalani kehidupan sehari-hari. Dalam konteks bisnis, kemajuan teknologi memainkan peran penting dalam meningkatkan efisiensi operasional, memperluas jangkauan pasar, meningkatkan kinerja dan daya saing perusahaan. Salah satu contoh penerapan teknologi adalah E-Catalog. E-catalog telah menjadi salah satu solusi yang efektif bagi bisnis dalam menghadapi era digital. Sejak awal tahun 2000-an, teknologi e-catalog telah mengalami perkembangan yang signifikan, menjadi salah satu platform yang memungkinkan bisnis untuk memperluas jangkauan pasar, meningkatkan efisiensi operasional, dan memberikan kenyamanan bagi pelanggan dengan menyediakan akses 24/7 untuk melihat produk atau jasa yang ditawarkan oleh perusahaan (Amelia, 2022). Dapunta Digital merupakan perusahaan yang bergerak di bidang multimedia dalam bidang jasa desain grafis yang membantu masyarakat yang kesulitan memasuki era digital. Mulai dari pembuatan logo perusahaan, brand, iklan, Banner, video animasi, *social media story* dan melayani

hingga urusan cetak mencetak bahan promosi. Dapunta telah berdiri sejak tahun 2019 dan telah dipercaya oleh lebih dari 700 klien. Dapunta mempunyai prinsip pelayanan prima, respon cepat, akurat, kualitas produk, kemudahan akses, ramah dan efisien. Sebagai salah satu yang bergerak di bidang multimedia, Dapunta memiliki sebuah permasalahan, yakni Dapunta melakukan penjualan jasa desain grafis hanya dengan menggunakan instagram, dan belum menggunakan sebuah website jenis *e-catalog*, menyebabkan beberapa masalah yang terjadi yaitu data transaksi, data pelanggan dan rekapan omset tidak terekap dengan baik karena terpisah diluar instagram tentu hal ini membuat admin dapunta mengalami kesulitan dalam menggabungkan data-data tersebut sehingga berpopulasi mengalami kesalahan pendataan di percetakan Dapunta inc.

Berdasarkan masalah ini, penulis menemukan solusi yakni *E-catalog* berbasis web yang dapat memudahkan pendataan transaksi, pelanggan, serta omset terekap tidak terpisah dalam suatu sistem menggunakan metode RUP yang pengembangan perangkat lunak dilakukan berulang-ulang (*iterative*). RUP memiliki empat fase pengembangan sistem yaitu *inception*, *elaboration*, *construction*, dan *transition*. Maka penulis mengangkat judul “E-Catalog Dapunta Inc Menggunakan Metode *Unified Modeling Language* (RUP)”.

1.2 Perumusan Masalah

Rumusan masalah penelitian ini adalah bagaimana membangun sebuah E-Catalog Dapunta Inc menggunakan metode RUP.

1.3 Batasan Masalah

Batasan masalah penelitian ini adalah:

1. Objek Penelitian adalah Dapunta Inc.
2. Aplikasi *E-Catalog* ini akan dibangun berbasis *website*.
3. *E-Catalog* ini dikembangkan menggunakan metode *Rational Unified Process* (RUP), dengan proses pemodelan menggunakan *Unified Modelling Language* (UML) berupa *Use Case Diagram*, *Activity Diagram*, *Sequence Diagram*, dan *Class Diagram*.
4. *E-catalog* dibangun menggunakan *framework* Laravel versi 10 dan *database* MySQL.
5. Pengujian *e-catalog* menggunakan metode *white box testing*.
6. *E-catalog* ini memiliki 4 tipe hak akses, yakni Admin, Pelanggan dan *Owner*.
7. Metode perancangan sistem yang digunakan yaitu RUP
8. Admin dapat mengelola pesanan, packing, pesanan dikirim, data produk, tambah produk, pesan, daftar pelanggan, review pelanggan.
9. Pelanggan dapat mengelola keranjang, checkout, dan melacak pesanan.
10. Owner dapat melihat omset penjualan.

1.4 Tujuan Penelitian

Tujuan utama penelitian ini adalah membangun E-Catalog Dapunta Inc menggunakan metode RUP.

1.5 Manfaat Penelitian

Manfaat penelitian ini terbagi menjadi 3, yaitu:

1. Manfaat Bagi Penulis

Manfaat penelitian bagi penulis adalah mengimplementasikan hasil belajar dan membuat sebuah sistem berbasis web.

2. Manfaat Bagi Perusahaan

Manfaat bagi perusahaan adalah dapat menggunakan e-catalog untuk mengatasi masalah data transaksi, data pelanggan dan rekapan omset tidak terekap dengan baik karena terpisah diluar instagram. *E-catalog* berbasis web yang dapat memudahkan pendataan transaksi, pelanggan, serta omset terekap tidak terpisah dalam suatu sistem yang ada pada Dapunta Inc.

3. Manfaat Bagi Akademik

Manfaat bagi akademik adalah dapat menjadi referensi dalam penelitian dikemudian hari.

1.6 Sistem Penulisan

Untuk mempermudah pembahasan dan penyusunan penulisan laporan penelitian ini menjadi lebih mudah dipahami dan terarah, dimana penulisan menjabarkan dalam 6 bab permasalahan dan masing-masing

bab akan diuraikan menjadi beberapa sub bab. Sistematika laporan penelitian diuraikan sebagai berikut:

BAB I PENDAHULUAN

Pada bab ini berisikan latar belakang, perumusan masalah, batasan masalah, tujuan penelitian, manfaat penelitian, dan sistematika penulisan.

BAB II GAMBARAN UMUM TEMPAT PENELITIAN

Pada bab ini penulis akan membahas tentang profil tempat penelitian, visi dan misi, struktur organisasi, tugas wewenang.

BAB III TINJAUAN PUSTAKA

Pada bab ini akan diuraikan teori-teori yang mendukung yang terkait dengan penelitian.

BAB IV METODE PENELITIAN

Pada bab ini diuraikan jenis data, teknik pengumpulan data, jenis penelitian, alat dan teknik pengembangan sistem serta alat dan teknik pengujian.

BAB V HASIL DAN PEMBAHASAN

Pada bab ini dilaporkan hasil-hasil yang diperoleh dalam penelitian dan pembahasan terhadap hasil yang telah dicapai dan uji coba teknologi. Hasil dan pembahasan disesuaikan dengan teknik pengembangan sistem yang digunakan.

BAB VI PENUTUP

Pada bab ini memberikan kesimpulan secara umum dari apa yang telah dibahas.

BAB II

GAMBARAN UMUM PERUSAHAAN

2.1 Profil Perusahaan

2.1.1 Sejarah Perusahaan

Dapunta Digital merupakan perusahaan yang bergerak di bidang multimedia dalam bidang jasa desain grafis yang membantu masyarakat yang kesulitan memasuki era digital. Mulai dari pembuatan logo perusahaan, branding, iklan, video komersial, video animasi dan melayani hingga urusan cetak mencetak bahan promosi. Dapunta telah berdiri sejak tahun 2019 dan telah dipercaya oleh lebih dari 700 klien. Dapunta mempunyai prinsip pelayanan prima, respon cepat, akurat, kualitas produk, kemudahan akses, ramah dan efisien.

2.2 Visi dan Misi

2.2.1 Visi

Menjadi Perusahaan Pemasaran Digital multinasional berbasis IT

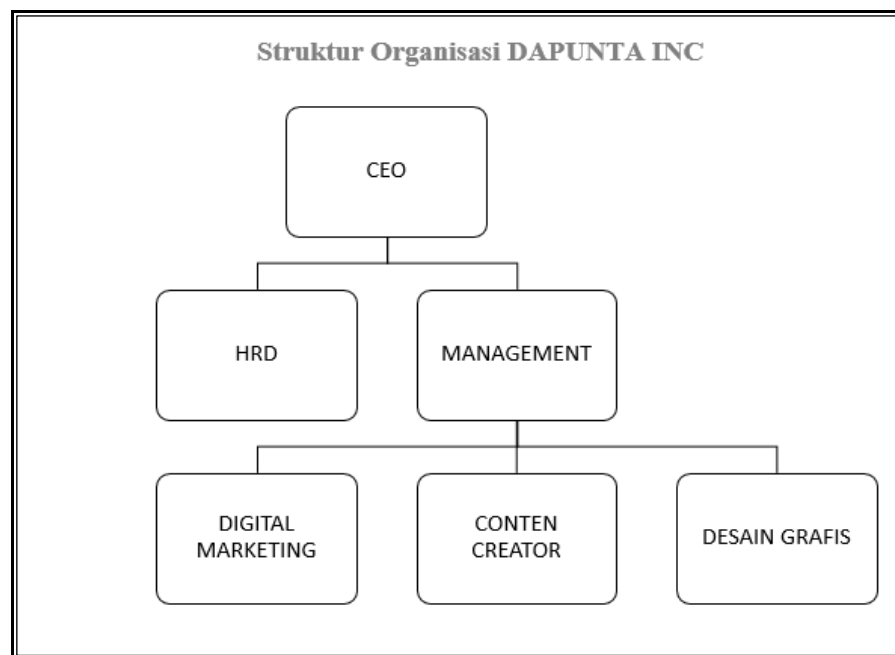
2.2.2 Misi

1. Menyediakan produk digital dengan kualitas terbaik.
2. Memberikan pelayanan terbaik di bidang digital.
3. Menerapkan Prosedur Operasional Standar yang berkualitas.
4. Menjaga keutuhan perusahaan dengan menerapkan prinsip kepemimpinan Fathonah, Amanah, Shiddiq dan Tabligh.

5. Memiliki Sumber Daya Manusia yang berkualitas dengan menjaga integritas baik terhadap pelanggan maupun sesama tim di perusahaan.
6. Menjaga standar kualitas produk terbaik dengan menjaga kualitas sumber daya teknologi yang dimiliki dan penerapan kendali mutu yang baik

2.3 Struktur Organisasi

Dapunta Inc memiliki struktur organisasi yang dapat dilihat pada gambar 2.1.



Gambar 2. 1 Struktur Orgnisasi DAPUNTA INC

Sumber: Dapunta Inc

2.4 Tugas Wewenang

Tugas wewenang Dapunta Inc adalah sebagai berikut:

1. *CEO (Chief Executive Officer)*
 - a.) Menjalankan visi dan misi Percetakan Dapunta Inc.
 - b.) Memimpin dan mengkoordinasikan seluruh karyawan.
 - c.) Mengkoordinasikan program kerja Dapunta Inc baik dalam perencanaan, pelaksanaan, evaluasi maupun pertanggungjawaban.
2. HRD
 - a.) Merencanakan kebutuhan SDM perusahaan.
 - b.) Perekrutan dan seleksi karyawan.
 - c.) Mengevaluasi kinerja karyawan.
3. Management
 - a.) Mengembangkan orang lain melalui penilaian hasil kerja.
 - b.) Membuat target dan sasaran.
 - c.) Menyusun rencana dan anggaran biaya.
 - d.) Memberikan latihan (training).
 - e.) Membuat laporan keuangan.
4. *Digital Marketing*
 - a.) Meningkatkan jumlah penjualan.
 - b.) Mendekati konsumen target.
 - c.) Memberikan informasi terkait jasa layanan.
 - d.) Mengelola social media dapunta inc.

5. *Content Creator*

- a.) Mengedit konten berupa foto dan video yang berkaitan dengan layanan jasa atau layanan produk.
- b.) Membuat feed atau video animasi produk semenarik mungkin untuk konten pemasaran.
- c.) Membantu mengelola social media.

6. Desain Grafis

- a.) Mendesain Logo, animasi, sticker, dan desain lainya sesuai reques pelanggan dengan mendominasi menggunakan *Canva*.
- b.) Mencetak Banner, totebag dan produk lainya yang tersedia di layanan jasa Dapunta Inc .
- c.) Packing barang untuk dikirim ke pelanggan.

BAB III

TINJAUAN PUSTAKA

3.1 E-Catalog

Menurut Harmayani (2020), *e-catalog* adalah penyebaran, penjualan, pemasaran, pembelian barang atau jasa dengan sarana elektronik seperti jaringan komputer, televisi, www, dan jaringan internet lainnya. E-catalog merupakan sebuah platform digital yang berisi kumpulan informasi mengenai produk atau layanan tertentu biasanya disertai dengan gambar, deskripsi, dan spesifikasi. yang memungkinkan pengguna untuk menjelajahi tentang produk atau layanan secara online.

E-catalog merupakan sebuah aplikasi belanja *online* yang dikembangkan oleh lembaga kebijakan pengadaan barang atau jasa pemerintah yang menyediakan berbagai macam produk dari berbagai komunitas yang dibutuhkan. Suatu sistem berupa daftar, jenis, spesifikasi teknis, harga barang, yang dituangkan dalam suatu daftar yang mudah dan dapat diakses secara digital kapanpun dan dimanapun.

3.2 Rational Unified Process

Menurut Rosa dan Shalahuddin (2022), *Rational Unified Process* (RUP) adalah pendekatan pengembangan perangkat lunak yang dilakukan berulang-ulang (*iterative*), fokus pada arsitektur (*architecture centric*), lebih diarahkan berdasarkan penggunaan kasus (*use case driven*). Metode RUP merupakan pendekatan pengembangan perangkat

lunak yang terstruktur dan berbasis iteratif dengan mengedepankan kolaborasi antara tim pengembang, dokumentasi yang komprehensif, dan manajemen risiko yang efektif. yang dimana pendekatan ini terdiri dari serangkaian tahapan yang diulang-ulang secara iteratif yang dimulai dari perencanaan hingga implementasi. RUP yaitu sebuah proses pembangunan sistem meliputi seluruh *lifecycle* pembangunan perangkat lunak yang menyediakan suatu pendekatan untuk membantu tugas dan tanggung jawab yang dikembangkan dengan mengumpulkan berbagai *best practises* yang terdapat dalam industri pengembangan perangkat lunak.

3.3 Unified Modelling Language (UML)

Menurut Sukamto dan Shalahuddin (2020:2), UML (*Unified Modeling Language*) adalah bahasa pemodelan visual yang digunakan untuk menggambarkan sistem perangkat lunak dengan menggunakan notasi dan diagram. UML sendiri juga memberikan standar penulisan sebuah sistem *blueprint*, yang meliputi konsep proses bisnis, penulisan kelas-kelas dalam bahasa program yang spesifik, skema *database*, dan komponen yang diperlukan dalam sistem *software*. Adapun berikut beberapa pemodelan yang digunakan:

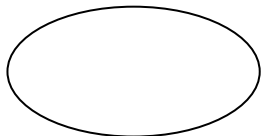
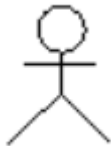
3.4 Use Case Diagram


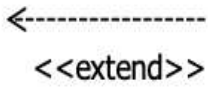
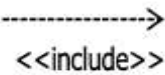
Use case atau diagram *use case* menggambarkan kelakuan (behavior) sistem informasi yang akan dibuat. *Use case* mendeskripsikan sebuah interaksi antara satu atau lebih aktor dengan sistem informasi yang akan

dibuat. Secara kasar, *use case* digunakan untuk mengetahui fungsi apa saja yang ada di dalam sebuah sistem informasi dan siapa saja yang berhak menggunakan fungsi-fungsi itu.

Penamaan pada *use case* didefinisikan sesederhana mungkin dan mudah untuk dipahami. Ada dua hal utama dalam *use case*, yaitu aktor dan *use case*. Aktor merupakan orang, proses, atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat di luar sistem informasi yang akan dibuat itu sendiri. Meskipun simbol dari aktor berbentuk orang, tapi aktor belum tentu merupakan orang. *Use case* merupakan fungsionalitas yang disediakan sistem sebagai unit-unit yang saling bertukar pesan antar unit atau aktor. Menurut Sugiarti (2020: 42), dalam *use case diagram* terdapat beberapa simbol yang digunakan dalam pembuatan *use case diagram* seperti pada tabel 3.1.

Tabel 3.1. Use Case Diagram

No	Simbol	Deskripsi
1		Fungsionalitas yang disediakan sistem sebagai unit yang saling bertukar pesan antar unit atau aktor; sering dinyatakan dengan menggunakan kata kerja, misal input data.
2		Orang, proses, atau sistem lain yang berinteraksi dengan sistem yang akan dikembangkan di luar sistem tersebut, sehingga meskipun simbol aktor adalah gambar orang, tapi aktor belum tentu merupakan orang; sering dinyatakan menggunakan kata benda, misal admin, mahasiswa, dll.



No	Simbol	Deskripsi
3		Menunjukkan komunikasi atau hubungan antara aktor dan <i>use case</i> atau antar <i>use case</i>
4		Hubungan <i>use case</i> tambahan ke <i>use case</i> dimana <i>use case</i> yang ditambahkan dapat berdiri sendiri walau tanpa <i>use case</i> tambahan itu; anak panah menuju <i>use case</i> yang dituju.
5		Hubungan antara <i>use case</i> tambahan ke sebuah <i>use case</i> dimana <i>use case</i> yang ditambahkan memerlukan <i>use case</i> ini untuk menjalankan fungsinya, sehingga <i>use case</i> harus dijalankan terlebih dulu; arah panah menunjuk pada <i>use case</i> yang perlu dijalankan terlebih dahulu





Sumber: Sugiarti, (2020)

3.5 Activity Diagram

Diagram aktivitas menggambarkan aliran kerja atau aktivitas dari sebuah sistem, tetapi bukan aktivitas aktor. Diagram aktivitas juga menggambarkan bagaimana alur sistem berawal, pilihan (*decision*) yang mungkin terjadi, dan bagaimana akhir alur sistem tersebut, adapun simbol-simbol yang digunakan pada *activity diagram* dapat dilihat pada tabel 3.2.

Tabel 3.2. Activity Diagram

No	Simbol	Nama	Keterangan
1		Status Awal	Sebuah diagram aktivitas memiliki sebuah status awal
2		Aktivitas	Aktivitas yang dilakukan sistem, aktivitas biasanya diawali dengan kata kerja

No	Simbol	Nama	Keterangan
3		Percabangan	Percabangan dimana ada pilihan aktivitas yang lebih dari satu
4		Penggabungan	Penggabungan dimana yang mana lebih dari satu aktivitas lalu digabungkan jadi satu
5		Status Akhir	Status akhir yang dilakukan sistem, sebuah diagram aktivitas memiliki status akhir.
6		<i>Swimlane</i>	<i>Swimlane</i> memisahkan organisasi bisnis yang bertanggung jawab terhadap aktivitas yang terjadi


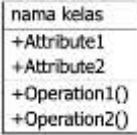






Sumber: Sugiarti, (2020)

3.6 Class Diagram

Class diagram digunakan untuk menggambarkan struktur dalam objek sistem. Diagram ini menunjukkan *class object* yang menyusun sistem dan juga hubungan antara *class object* (Sugiarti, 2020: 37). Kelas memiliki atribut dan metode atau operasi. Atribut adalah variabel-variabel yang mendeskripsikan properti dengan bentuk sebaris teks dalam kelas tersebut, sedangkan metode adalah fungsi yang dimiliki oleh kelas yang dalam *class diagram* dilambangkan menggunakan simbol-simbol yang dapat dilihat pada tabel 3.3.

Tabel 3.3. Class Diagram

No	Simbol	Deskripsi
----	--------	-----------



No	Simbol	Deskripsi
1		<i>Package</i> merupakan sebuah bungkus dari satu atau lebih kelas
2		Kelas pada struktur sistem, setiap kelas memiliki nama, atribut, dan <i>operation</i> atau <i>method</i> .
3		Sama seperti konsep <i>interface</i> dalam pemrograman berorientasi objek.
4		Relasi antar kelas dengan pengertian umum.
5		Relasi antar kelas dengan pengertian kelas yang satu digunakan oleh kelas yang lain.
6		Relasi antar kelas dengan pengertian generalisasi spesialisasi (umum-khusus).
7		Relasi antar kelas dengan pengertian kebergantungan antar kelas.
8		Relasi antar kelas dengan makna semua-sebagian (<i>whole-part</i>).


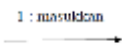
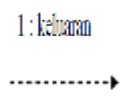
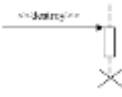
Sumber: Sugiarti, (2020)

3.7 Sequence Diagram

Sequence Diagram merupakan suatu susunan mengenai urutan waktu tertentu. *Sequence Diagram* dapat menggambarkan satu tugas dari beberapa tugas di dalam suatu *use case* secara lebih detail. (Uus Rusmawan, 2019:87). Simbol dari *sequence diagram* dapat dilihat pada tabel 3.4.

Tabel 3.4. Simbol-simbol *Sequence Diagram*

No.	Nama	Simbol	Keterangan
1.	<i>Aktor</i>		Orang, proses, atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat diluar sistem informasi yang akan dibuat itu sendiri, jadi walaupun simbol dari aktor adalah gambar orang, tapi aktor belum tentu merupakan orang, biasanya dinyatakan dalam menggunakan kata benda di awal frase nama aktor.
2.	<i>Garis hidup/lifeline</i>		Menyatakan kehidupan suatu objek.
3.	<i>Objek</i>		Menyatakan objek yang berinteraksi pesan.
4.	<i>Waktu Aktif</i>		Menyatakan objek dalam keadaan aktif dan berinteraksi, semuanya yang terhubung dengan waktu aktif ini adalah sebuah tahapan yang dilakukan di dalamnya.
5.	<i>Pesan tipe Create</i>		Menyatakan suatu objek membuat objek yang lain, arah panah mengarah pada objek yang dibuat.

No.	Nama	Simbol	Keterangan
6.	<i>Pesan tipe Call</i>		Menyatakan suatu objek memanggil operasi/metode yang ada pada objek lain atau dirinya sendiri.
7.	<i>Pesan tipe Send</i>		Menyatakan bahwa suatu objek mengirimkan data/masukkan/informasi ke objek lainnya, arah panah mengarah pada objek yang dikirim.
8.	<i>Pesan tipe Return</i>		Menyatakan bahwa suatu objek yang telah menjalankan suatu operasi atau metode menghasilkan suatu kembalian ke objek tertentu, arah panah mengarah pada objek yang menerima kembalian.
9.	<i>Pesan tipe Destroy</i>		Menyatakan suatu objek mengakhiri hidup objek yang lain, arah panah mengarah pada objek yang diakhiri, sebaliknya jika ada <i>create</i> maka ada <i>destroy</i> .

Sumber: Uus Rusmawan, (2019)

3.8 Hasil Penelitian Terdahulu

Beberapa jurnal terkait dengan penelitian sebagai referensi digunakan penulis dalam mengkaji penelitian yang dilakukan. Berikut merupakan beberapa penelitian terdahulu dapat dilihat pada tabel 3.4.

Tabel 3.5. Penelitian Terdahulu

No	Judul	Penulis dan Tahun	Hasil
1	Penerapan Metode	Fidi	Penelitian ini bertujuan untuk

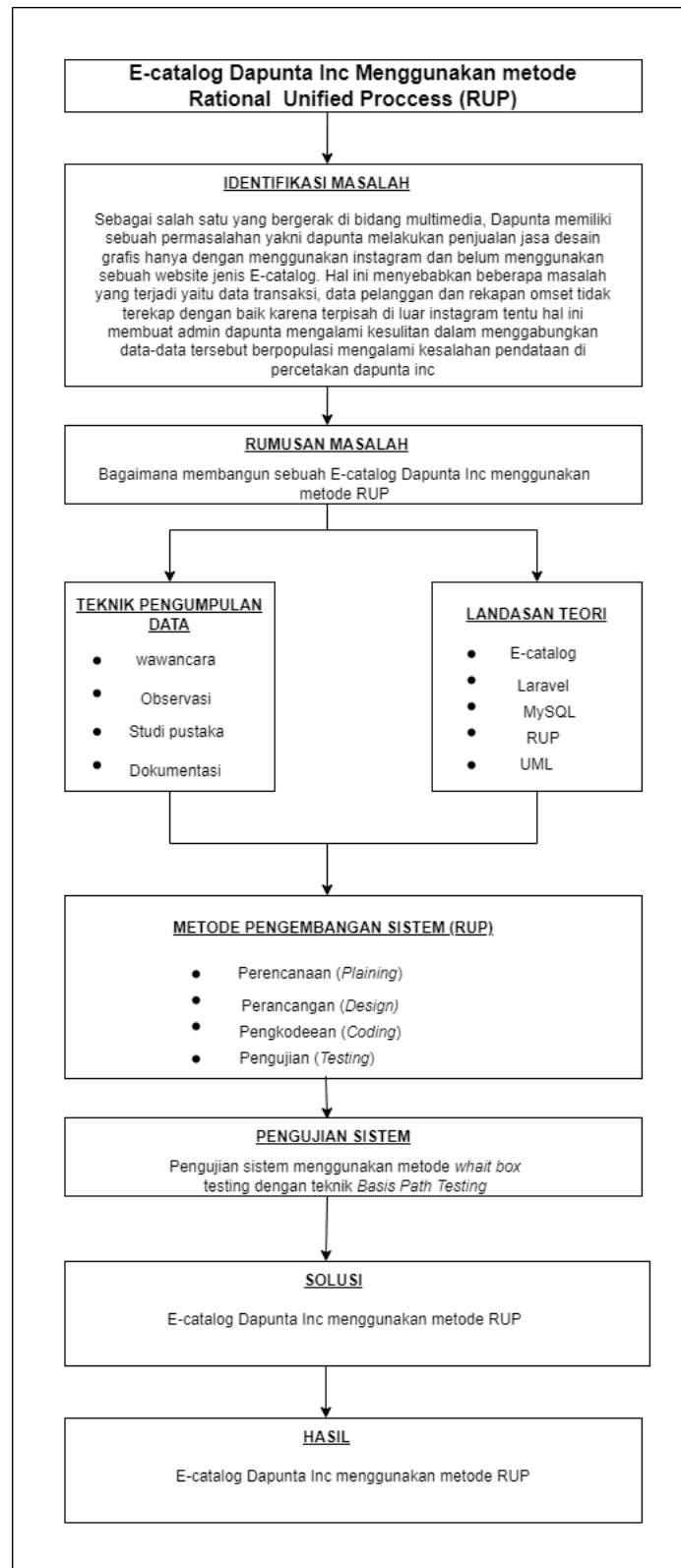
No	Judul	Penulis dan Tahun	Hasil
	Rational Unified Process Pada Perancangan Sistem Pengolah Data Arisankita (Vol.10, No.02, Juli 2019 p-ISSN: 2087-1627, e-ISSN: 2685-9858)	Supriadi, Reza Hardian (2019)	merancang sistem berbasis web untuk program usaha Arisankita dengan mengimplementasikan metode RUP, dalam meningkatkan akurasi pengolahan data pada perusahaan, mengoptimalkan otomasi penjadwalan, serta meningkatkan efisiensi, efektivitas dan produktivitas.
2	Analisis Implementasi E-Katalog Terhadap Perkembangan UMKM di Kota Medan dalam Mendukung Kemajuan Ekonomi Syariah (Fitri Diani, Fauzi Arif Lubis (2022)	Program Wali Kota Medan yang menginginkan bahwa pelaku usaha UMKM masuk ke dalam Katalog Elektronik (E-Katalog) sudah terlaksanakan. Tetapi belum maksimal dikarenakan pencairan dana tidak sesuai dengan perjanjian kontrak. Dan hanya sedikit pelaku usaha UMKM di Kecamatan Medan Marelan yang dapat bergabung di E-Katalog Lokal Medan.
3	Pengaruh Pelaksanaan E Katalog Dalam Pengadaan Barang/jasa Pemerintah Terhadap Umkm (Jurnal USM Law Review Vol 3 No 1 Tahun 2020)	Muhammad Iqbal (2022)	Pelaksanaan e katalog dalam pengadaan barang/jasa pemerintah berpengaruh pada sektor UMKM, masih adanya pelaku industri kecil yang selama ini menjadi rantai distribusi barang/jasa. Melalui e katalog yang memutus rantai distribusi tersebut menyebabkan terhapusnya peran penyedia jasa kecil yang selama ini menjadi bagian dari rantai distribusi.

Berdasarkan penelitian terdahulu, dapat disimpulkan bahwa *e-catalog* dapat membantu mempermudah sistem jual beli dan dapat membantu meningkatkan penjualan secara luas, selain itu juga dapat mempermudah proses pencatatan transaksi, pencarian barang, dan monitoring stok barang, selain itu dengan menggunakan metode RUP dapat meningkatkan akurasi

pengolahan data pada perusahaan, mengoptimalkan otomatisasi penjadwalan, serta meningkatkan efisiensi, efektivitas dan produktivitas. Untuk penelitian terdahulu fokus penelitiannya seperti aspek *E-catalog* yang diteliti (misalnya fungsionalitas, usability, implementasi, tujuan penelitian dan pertanyaan penelitian yang diajukan). Sedangkan penelitian penulis saat ini memiliki tujuan membangun E-catalog untuk mengatasi masalah yang terjadi di Dapunta Inc seperti data transaksi, data pelanggan, dan rekapan omset yang terpisah di luar Instagram.

3. Kerangka Pemikiran

Berikut merupakan hasil dari kerangka pemikiran yang dapat dilihat pada gambar 3.2.



Gambar 3. 1 Kerangka Pemikiran

BAB IV METODE PENELITIAN

4.1 Lokasi dan Waktu Penelitian

4.1.1 Lokasi Penelitian

Penelitian ini dilaksanakan di Dapunta Inc berlokasi di di Jl. Lembang No.14 Kel Blok F14, Lorok Pakjo, Ilir Barat I, Palembang City, South Sumatra 30137

4.1.2 Waktu Penelitian

Peneliti menyusun kegiatan dalam bentuk jadwal dari bulan Februari 2024 hingga bulan Juni 2024. Bentuk jadwal penelitian dapat dilihat pada tabel 4.1.

Tabel 4.1. Jadwal Penelitian

No	Nama Kegiatan	Bulan-ke																			
		Februari				Maret				April				Mei				Juni			
		1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4
1	Wawancara																				
2	Observasi																				
3	Studi Pustaka																				
4	<i>Inception</i>																				
5	<i>Elaboration</i>																				
6	<i>Construct</i>																				

No	Nama Kegiatan	Bulan-ke																			
		Februari				Maret				April				Mei				Juni			
		1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4
	<i>ion</i>																				
7	<i>Transitio n</i>																				

4.2 Jenis Data

Penulis dalam melakukan penelitian ini menggunakan 2 sumber data, yaitu:

4.2.1 Data Primer

Data primer, merupakan data yang diperoleh dari sumber data pertama di lapangan, dalam penelitian ini data primernya yaitu data-data yang didapatkan dari Dapunta Inc.

4.2.2 Data Sekunder

Data Sekunder, merupakan data yang diperoleh dari sumber kedua. Data sekunder ini untuk melengkapi data primer, dan biasanya data sekunder ini sangat membantu peneliti bila data primer terbatas atau sulit diperoleh, adapun data sekunder diambil dari bukti pemesanan, daftar harga dan data pendukung lainnya yang bersumber dari data-data dokumentasi Dapunta Inc.

4.3 Teknik Pengumpulan Data

Teknik pengumpulan data dalam penelitian ini, penulis menggunakan 4 metode, yaitu:

4.3.1 Wawancara

Wawancara Menurut Hutagalung (2020:119) Wawancara merupakan salah satu teknik dalam mengumpulkan data dapat dilaksanakan melalui interaksi lisan antara peneliti dengan subjek penelitian atau responden atau sumber data, pengumpulan data yang dilakukan dengan wawancara, mengajukan pertanyaan langsung kepada pihak Dapunta. Data yang didapat dari hasil wawancara menyatakan upaya untuk terus berkembang dan bersaing di pasar yang semakin kompetitif, Dapunta menyadari pentingnya memanfaatkan teknologi dalam operasional bisnisnya.

4.3.2 Observasi

Observasi Menurut Manoppo (2023:11) yaitu teknik penelitian dengan melakukan pengamatan langsung pada perusahaan yang menjadi objek penelitian Dalam hal ini penulis melakukan penelitian dan pengamatan secara langsung pada Dapunta. Penulis melakukan observasi langsung ke Dapunta inc untuk mengidentifikasi masalah di dapunta tersebut, Berdasarkan hasil observasi selama penelitian, penulis mendapatkan informasi mengenai transaksi terpisah di luar Instagram, yang dimana hal

tersebut dapunta inc kesulitan dalam merekap data transaksi di karenakan transaksinya terpisah.

4.3.3 Studi Pustaka

Studi Pustaka Menurut Ansori (2023:111) Studi pustaka (*library research*) dimana penulis mengumpulkan data dengan cara membaca, mempelajari, dan menganalisis jurnal-jurnal, buku, artikel dari peneliti terdahulu yang ada. Penulis mencari referensi melalui jurnal yang berkaitan dengan permasalahan yang penulis yang diangkat.

4.3.4 Dokumentasi

Dokumentasi Menurut Merentek, (2023:648) dokumentasi merupakan metode yang bersumber pada benda-benda yang tertulis berupa buku- buku, majalah, dokumen, peraturan-peraturan, notulen rapat, catatan harian dan sebagainya. Dalam penelitian ini memperoleh dokumentasi dari admin berupa struktur organisasi, profil dapunta, sejarah dapunta, data produk, visi dan misi Dapunta inc.

4.4 Alat dan Teknik Pengembangan Sistem

4.4.1 Alat Pengembangan Sistem

Pemodelan sistem yang digunakan penulis adalah pemodelan *Unified Modelling Language* yang terdiri dari *Use Case Diagram*, *Activity Diagram*, *Sequence Diagram*, dan *Class Diagram*.

4.4.2 Teknik Pengembangan Sistem

Teknik pengembangan sistem yang digunakan penulis adalah metode *Rational Unified Process* (RUP).

4.5 Alat dan Teknik Pengujian

Teknik pengujian sistem yang digunakan penulis adalah metode *white box testing*.

4.5.1 White Box Testing

Menurut Roni Subagia (2020) *White box* disebut juga sebagai pengujian kotak kaca atau pengujian struktural dimana pengujian yang dikembangkan berdasarkan pada kode program. pengujian dalam *white box testing* memiliki pengetahuan tentang kode dan penulisan kasus uji dengan parameter yang sesuai.

Dalam pengujian sistem ini menggunakan *white box testing* dengan menggunakan teknik *Basis path testing*.

BAB V

HASIL DAN PEMBAHASAN

5.1 Hasil

Selama melakukan penelitian pada Dapunta Inc, penulis menemukan beberapa masalah, yaitu Dapunta melakukan penjualan jasa desain grafis hanya dengan menggunakan instagram, dan belum menggunakan sebuah website jenis *e-catalog*, menyebabkan beberapa masalah yang terjadi yaitu data transaksi, data pelanggan dan rekapan omset tidak terekap dengan baik karena terpisah diluar instagram tentu hal ini membuat admin dapunta mengalami kesulitan dalam menggabungkan data-data tersebut sehingga mengalami kesalahan pendataan di percetakan Dapunta Inc. Maka dari itu dibutuhkan sebuah *e-catalog* Dapunta Inc untuk membantu proses penjualan, adapun penulis menggunakan metode *Rational Unified Process* (RUP), metode RUP sendiri terdiri dari 4 tahap.

5.1.1 Inception

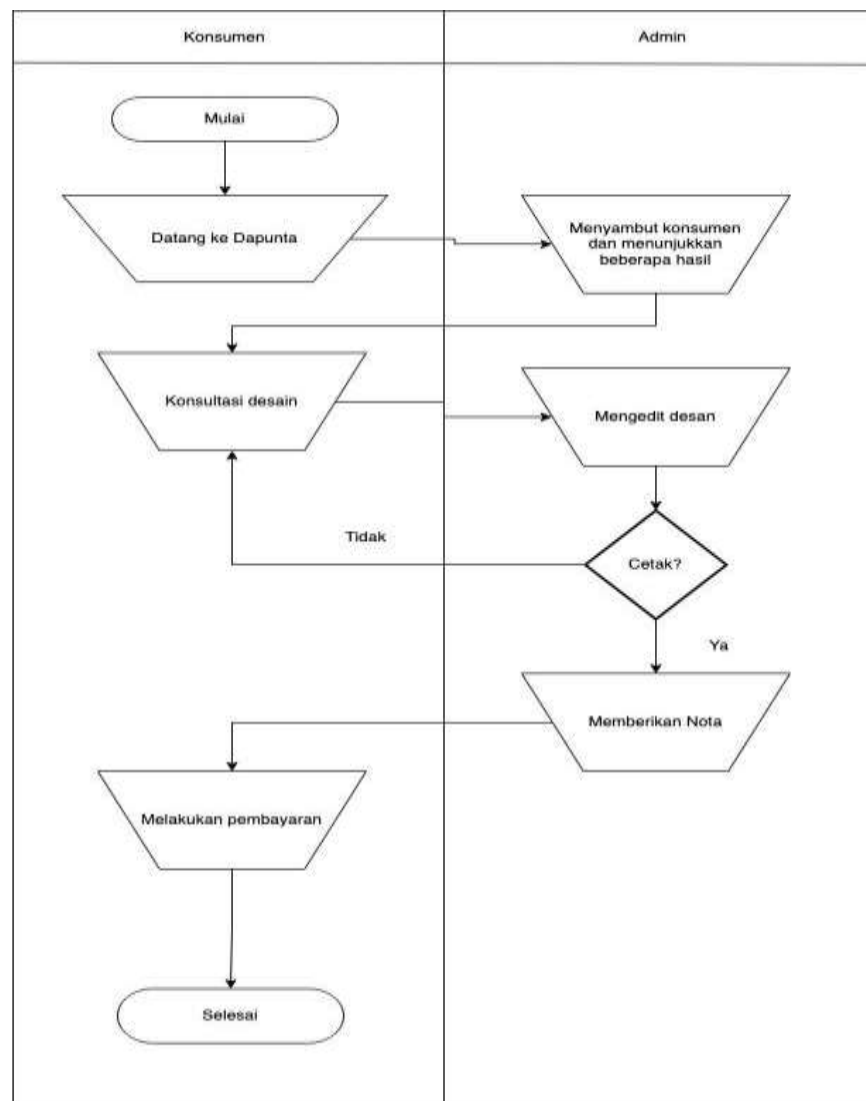
Tahap ini merupakan tahapan awal dalam perencanaan pembuatan perangkat lunak baru, tahap ini memodelkan proses bisnis yang dibutuhkan. Pada tahap ini penulis mengumpulkan kebutuhan dalam pembuatan *e-catalog* Dapunta Inc. Berdasarkan proses pengumpulan data yang penulis lakukan, penulis mendefinisikan data-data tersebut yang dibutuhkan sebagai berikut:

- 1. Data produk.**

2. Data harga.
3. Data *user*.
4. Data kategori.

5.1.1.1 Alur yang berjalan

Alur penjualan yang berjalan pada Dapunta Inc dapat dilihat pada gambar 5.1.



Gambar 5. 1 Alur yang berjalan pada Dapunta Inc

Penjelasan dari alur yang berjalan adalah sebagai berikut:

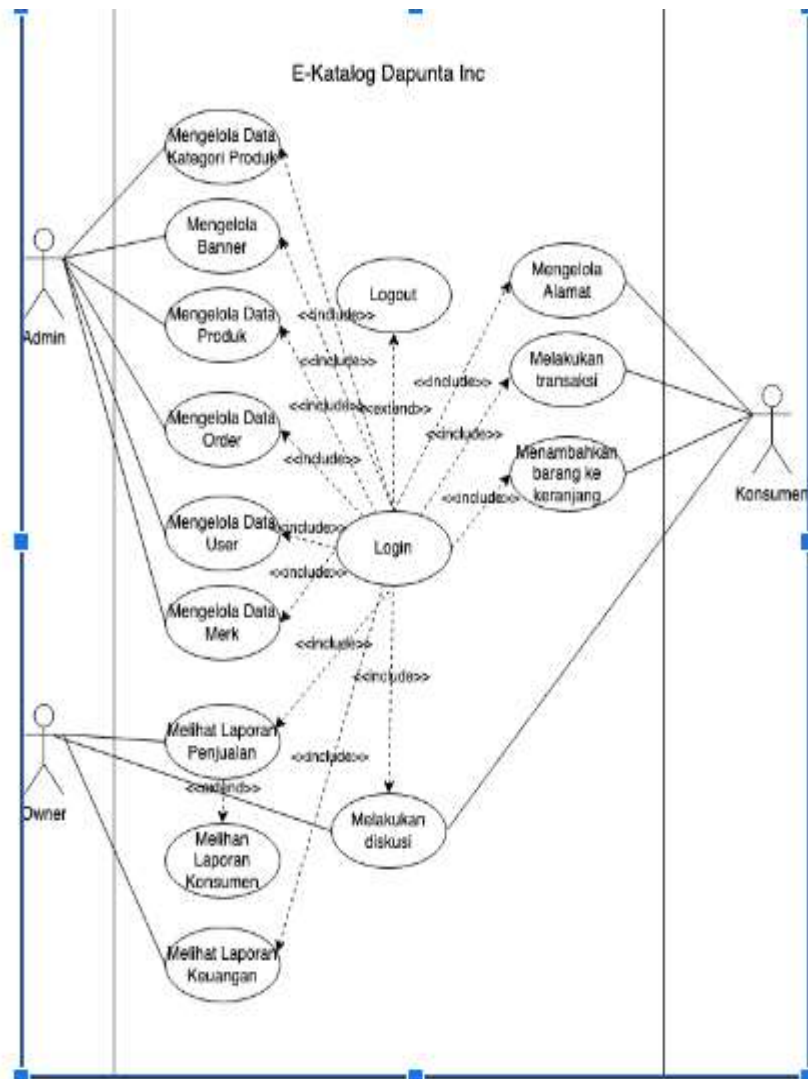
1. Mulai.
2. Datang ke dapunta
3. Menyambut konsumen dan menunjukkan beberapa hasil
4. Konsultasi Desain
5. Mengedit desain
6. Jika ya lanjut memberikan nota
7. Masukan pembayaran
8. Selesai.

5.1.2 Elaboration

Tahap ini merupakan tahap untuk melakukan desain secara lengkap berdasarkan hasil analisis pada tahap inception. Aktivitas yang dilakukan pada tahap ini antara lain mencakup pembuatan desain arsitektur subsistem (*architecture pattern*), desain *database*, desain *user interface*, pemodelan diagram UML (*use case diagram*, *activity diagram*, *sequence diagram* dan *class diagram*).

5.1.2.1 Use Case Diagram yang diusulkan

Adapun *use case diagram* yang diusulkan pada *E-Catalog Dapunta Inc* ini dapat dilihat pada gambar 5.2



Gambar 5. 2 Use Case Diagram yang diusulkan

Use case diagram menggambarkan alur proses yang diusulkan, pada gambar 5.2. *use case diagram* digambarkan memiliki 3 aktor, yaitu admin, pimpinan dan konsumen, dimana admin memiliki akses berupa mengolah data user, data kategori barang, dan data barang, sedangkan konsumen

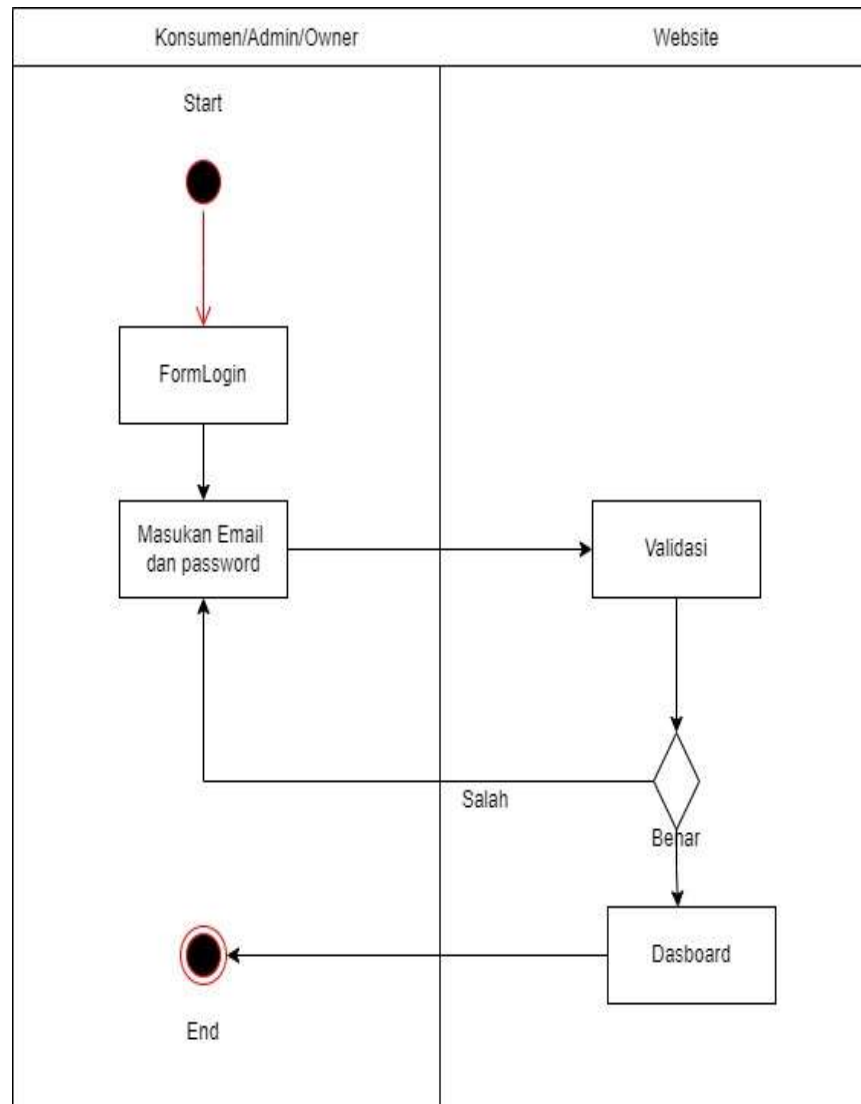
memiliki akses untuk registrasi, melihat dan mencari produk dan melakukan transaksi/*checkout* produk.

5.1.2.2 Activity Diagram Admin

Adapun *activity diagram* yang diusulkan pada pembuatan *E-Catalog* Dapunta Inc ini dapat dilihat pada gambar 5.3.

1. Activity Diagram login

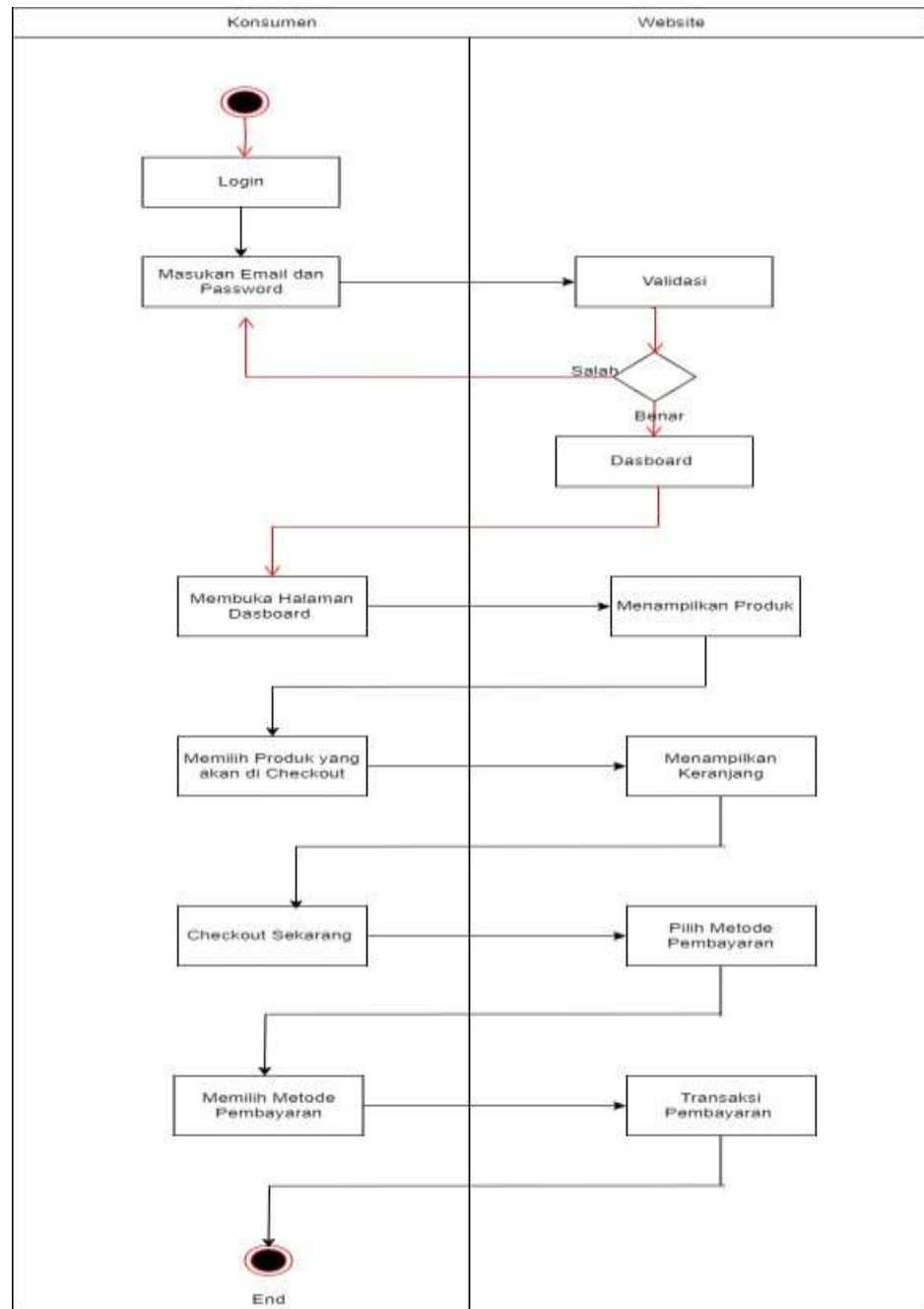
Gambar *activity diagram login* yang diusulkan dapat dilihat sebagai berikut.



Gambar 5.3 Activity Diagram Login

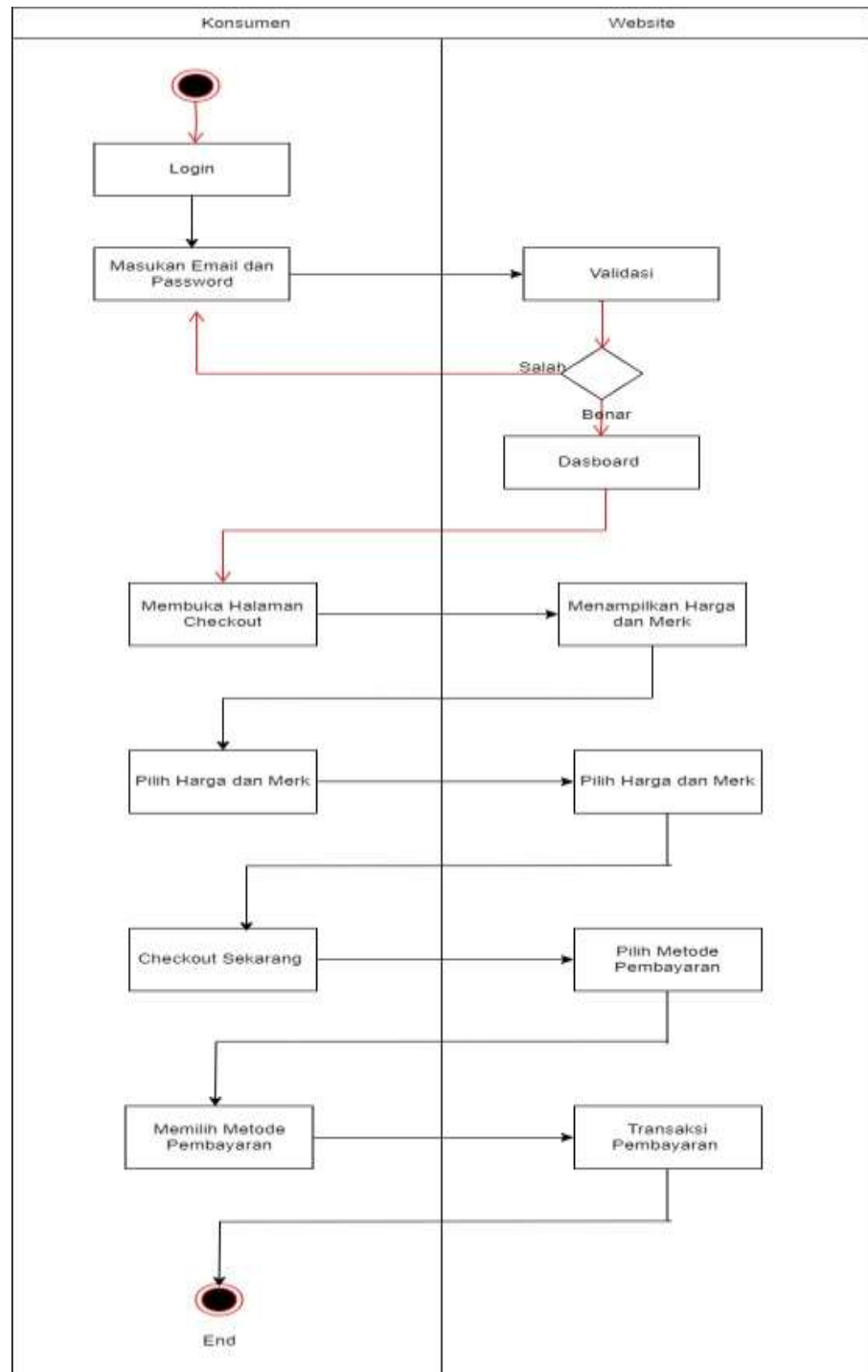
2. *Activity Diagram* Menambahkan barang ke keranjang

menambahkan barang ke keranjang yang diusulkan dapat dilihat pada gambar berikut.



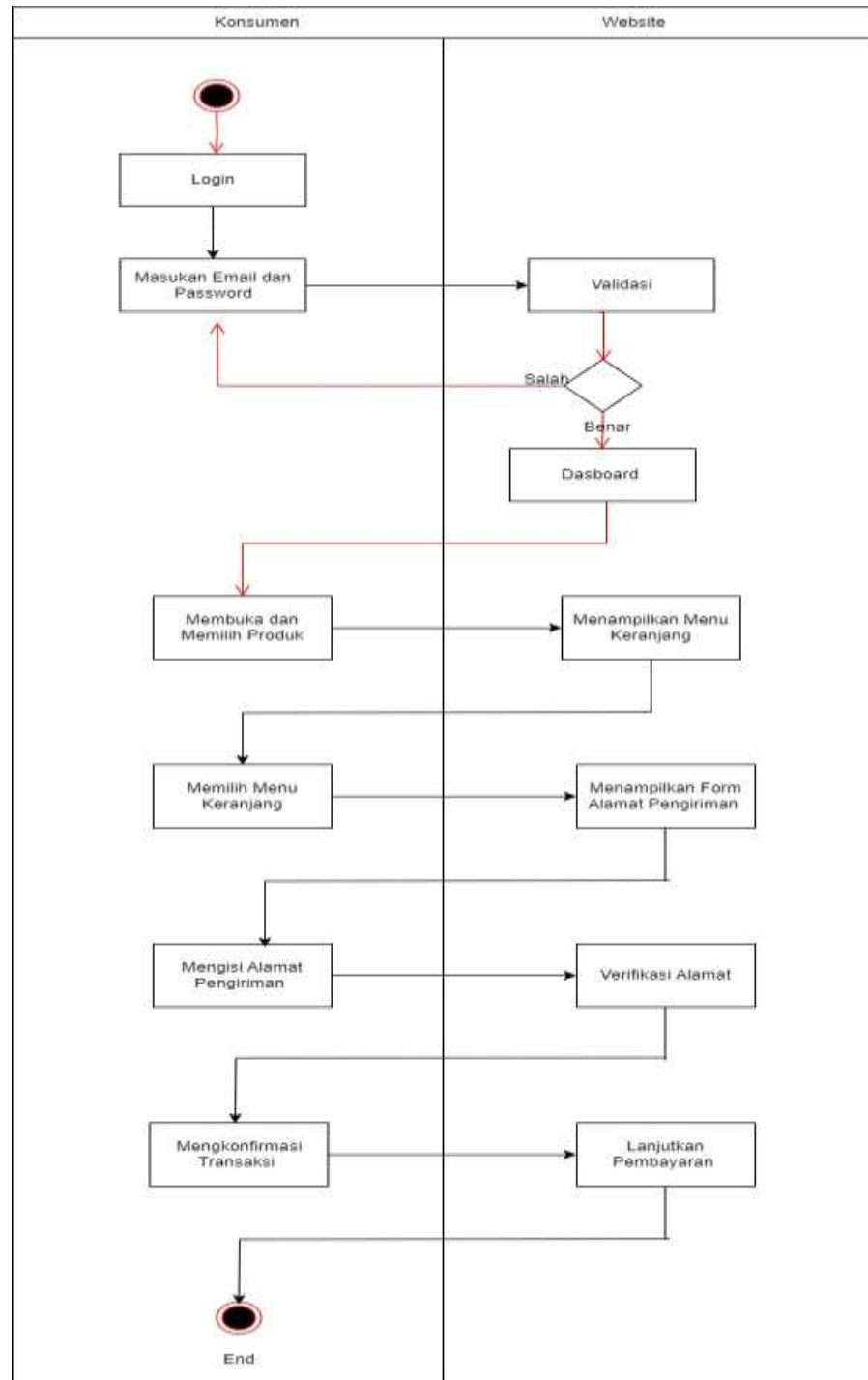
Gambar 5. 4Activity Diagram Konsumen

3. *Activity Diagram* Transaksi melakukan transaksi yang diusulkan dapat dilihat sebagai berikut.



Gambar 5. 5Activity Diagram Konsumen

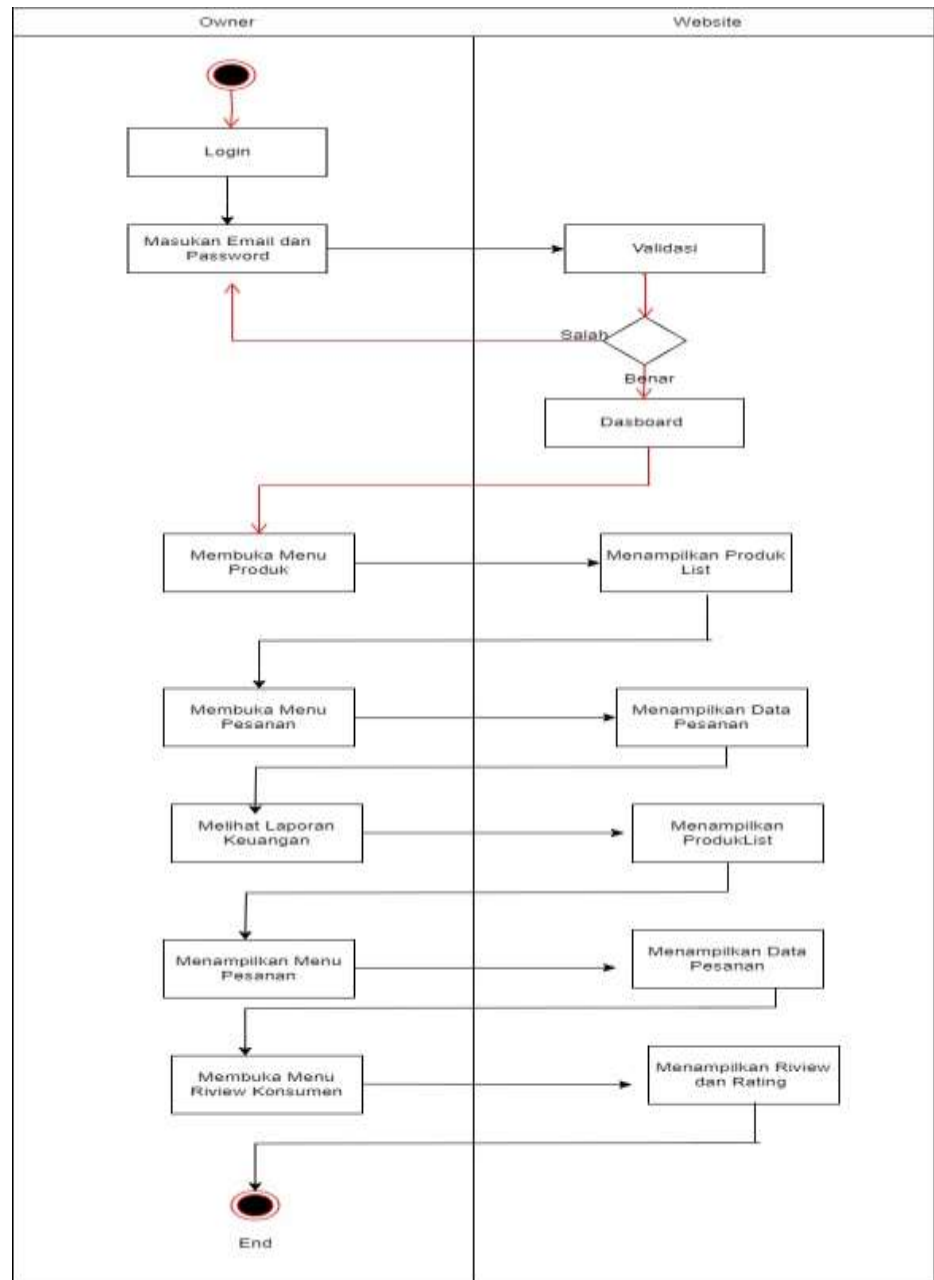
4. *Activity Diagram* konsumen mengelola alamat yang diusulkan dapat dilihat pada gambar berikut.



Gambar 5. 6Activity Diagram Konsumen

5. *Activity diagram* Laporan Penjualan

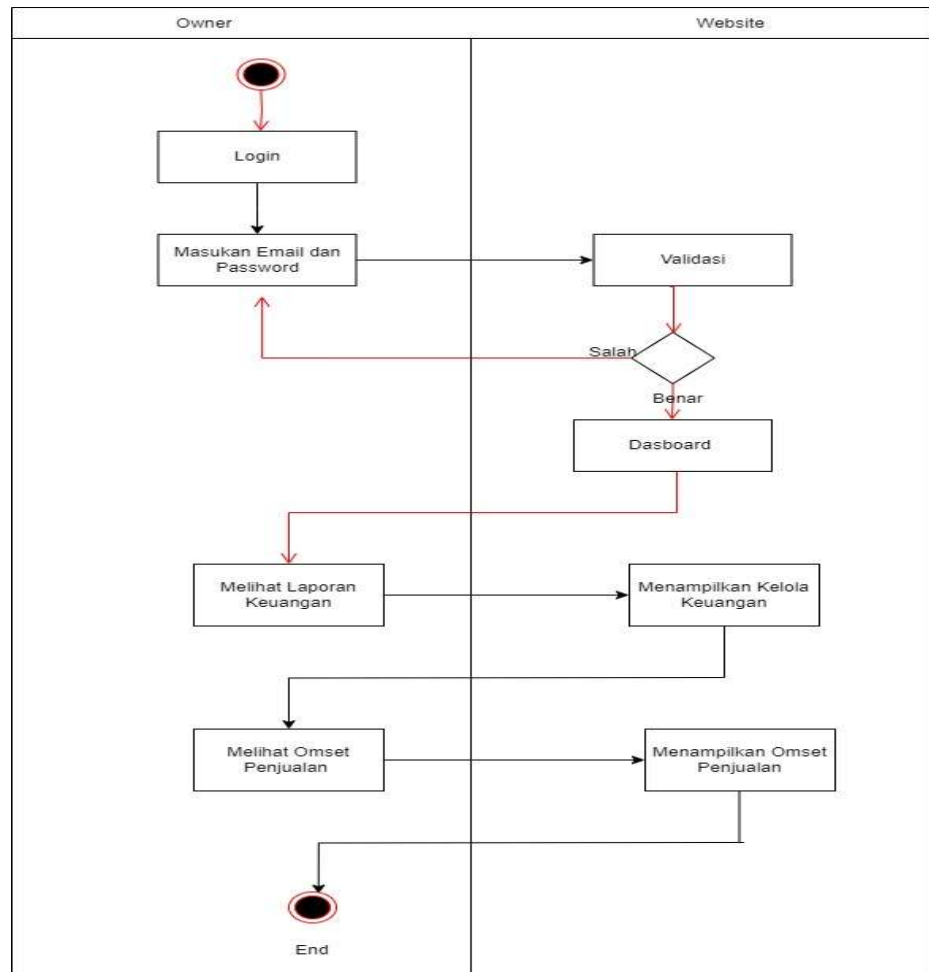
Gambar *activity diagram owner* melihat laporan penjualan yang diusulkan dapat dilihat pada gambar berikut.



Gambar 5. 7Activity Diagram Owner

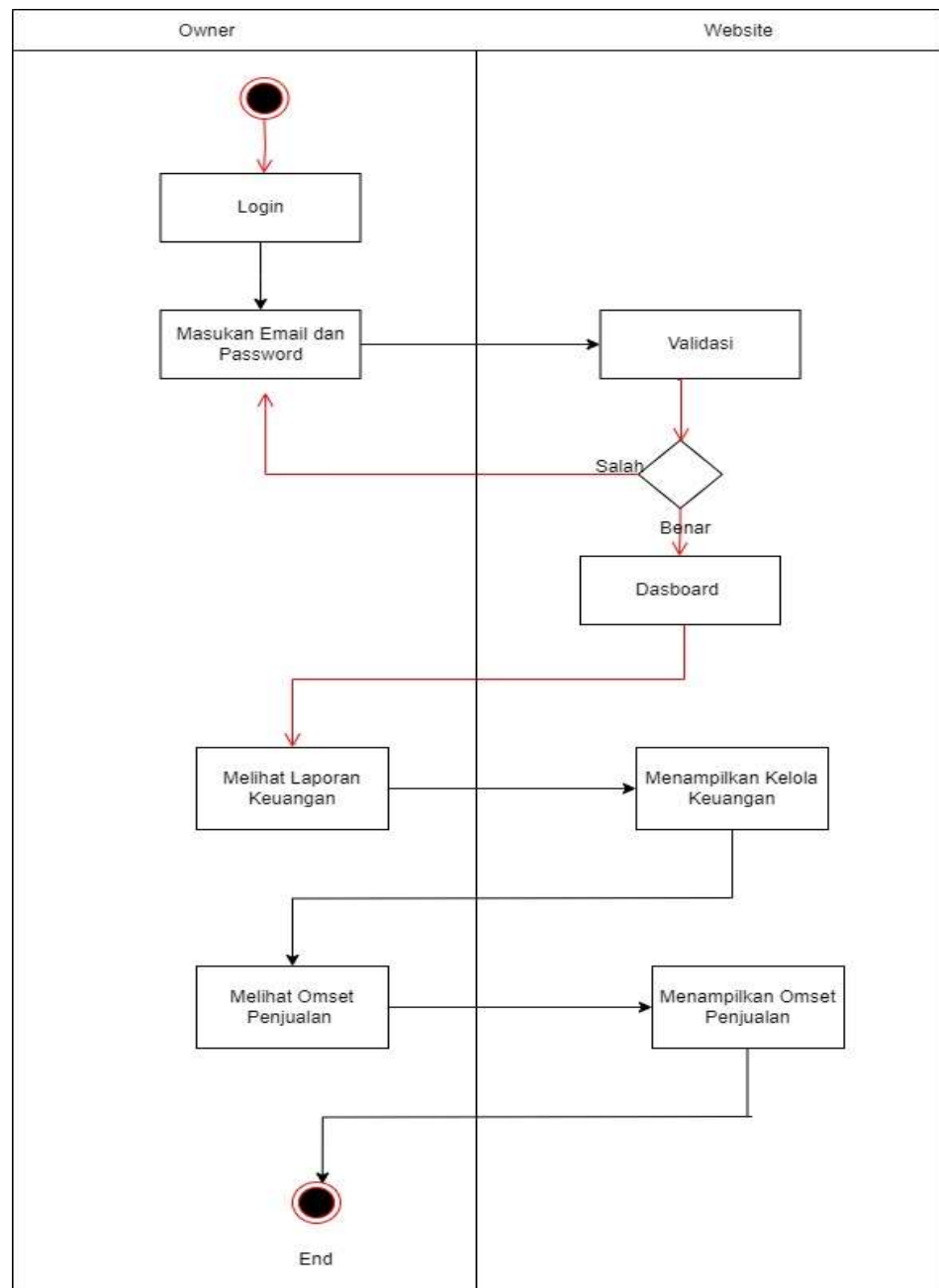
6. Activity Diagram Laporan Konsumen

Gambar *activity diagram* owner melihat laporan konsumen yang diusulkan dapat dilihat sebagai berikut.



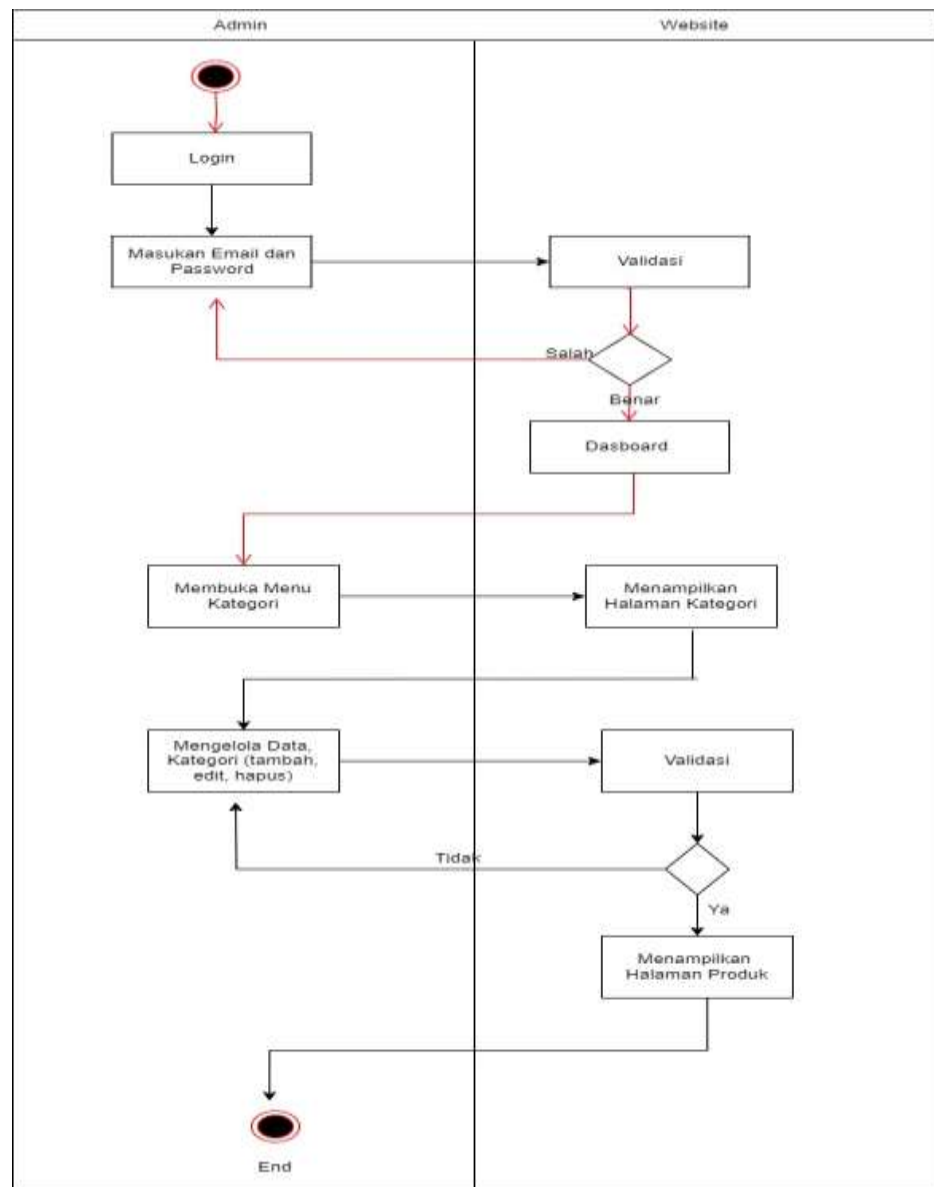
Gambar 5. 8 Activity Diagram Owner

7. *Activity Diagram* Laporan Keuangan melihat laporan keuangan yang diusulkan dapat dilihat sebagai berikut.



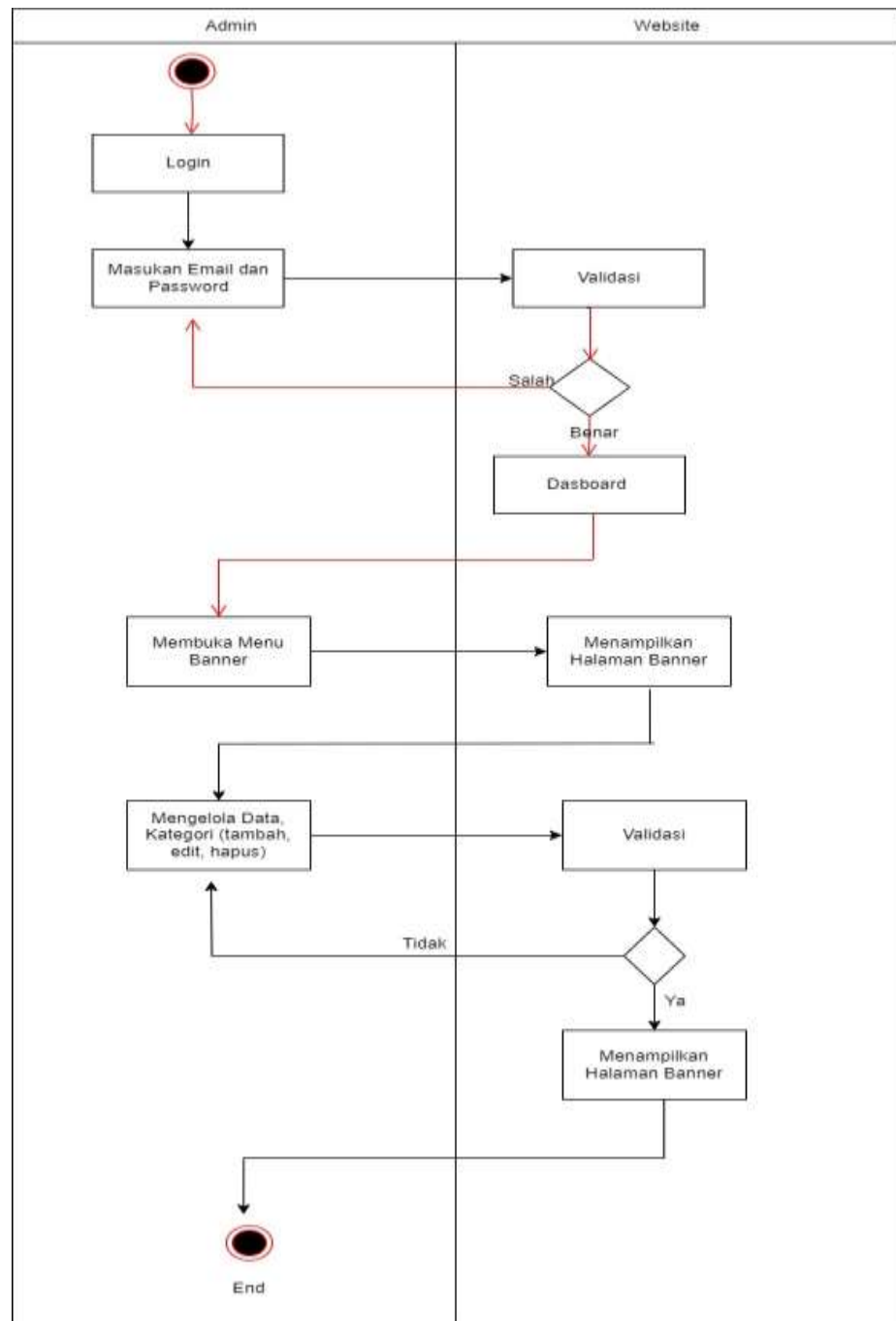
Gambar 5. 9Activity Diagram Owner

8. *Activity Diagram* kategori produk admin mengelola data kategori produk yang diusulkan dapat dilihat sebagai berikut.



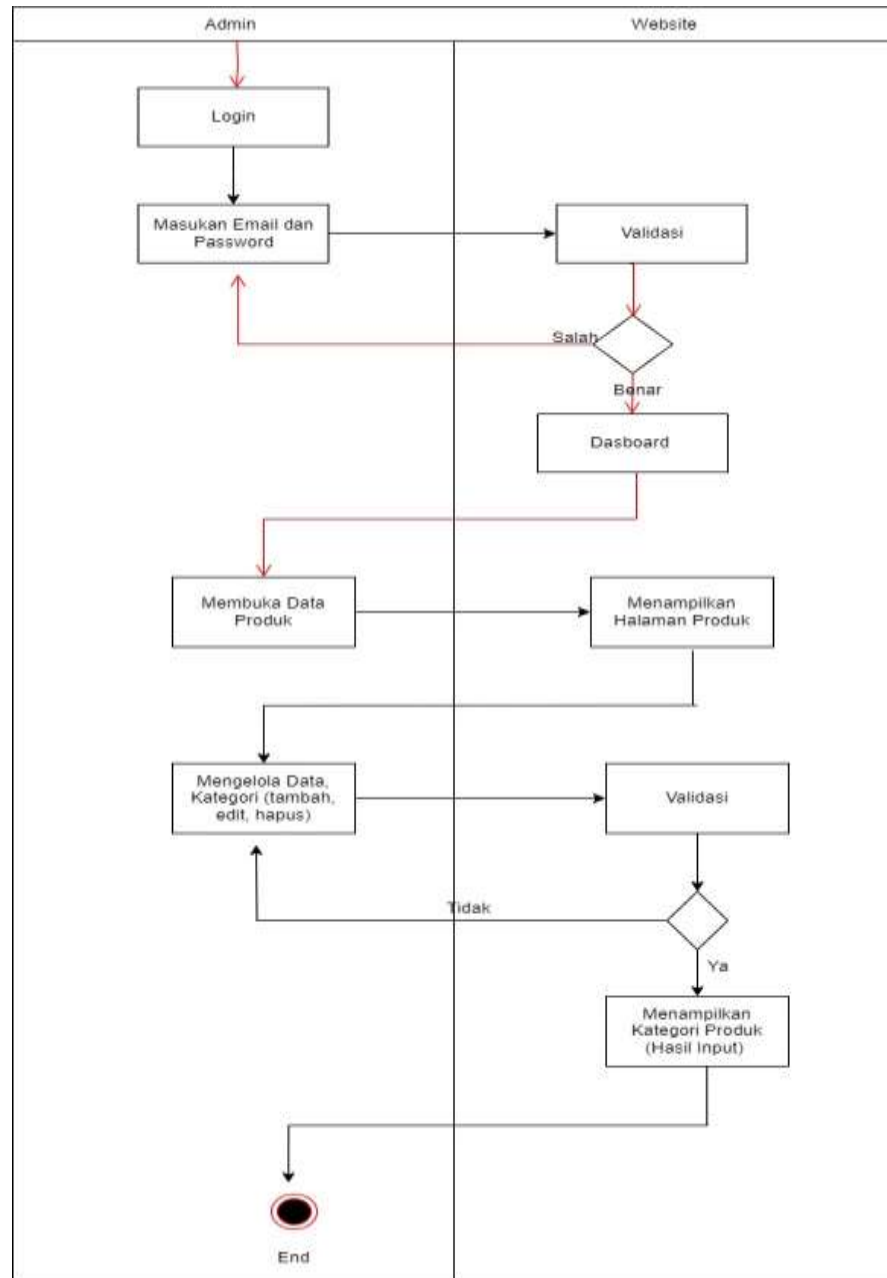
Gambar 5. 10Activity Diagram Admin

9. *Activity Diagram* mengelola banner yang diusulkan dapat dilihat pada gambar sebagai berikut.



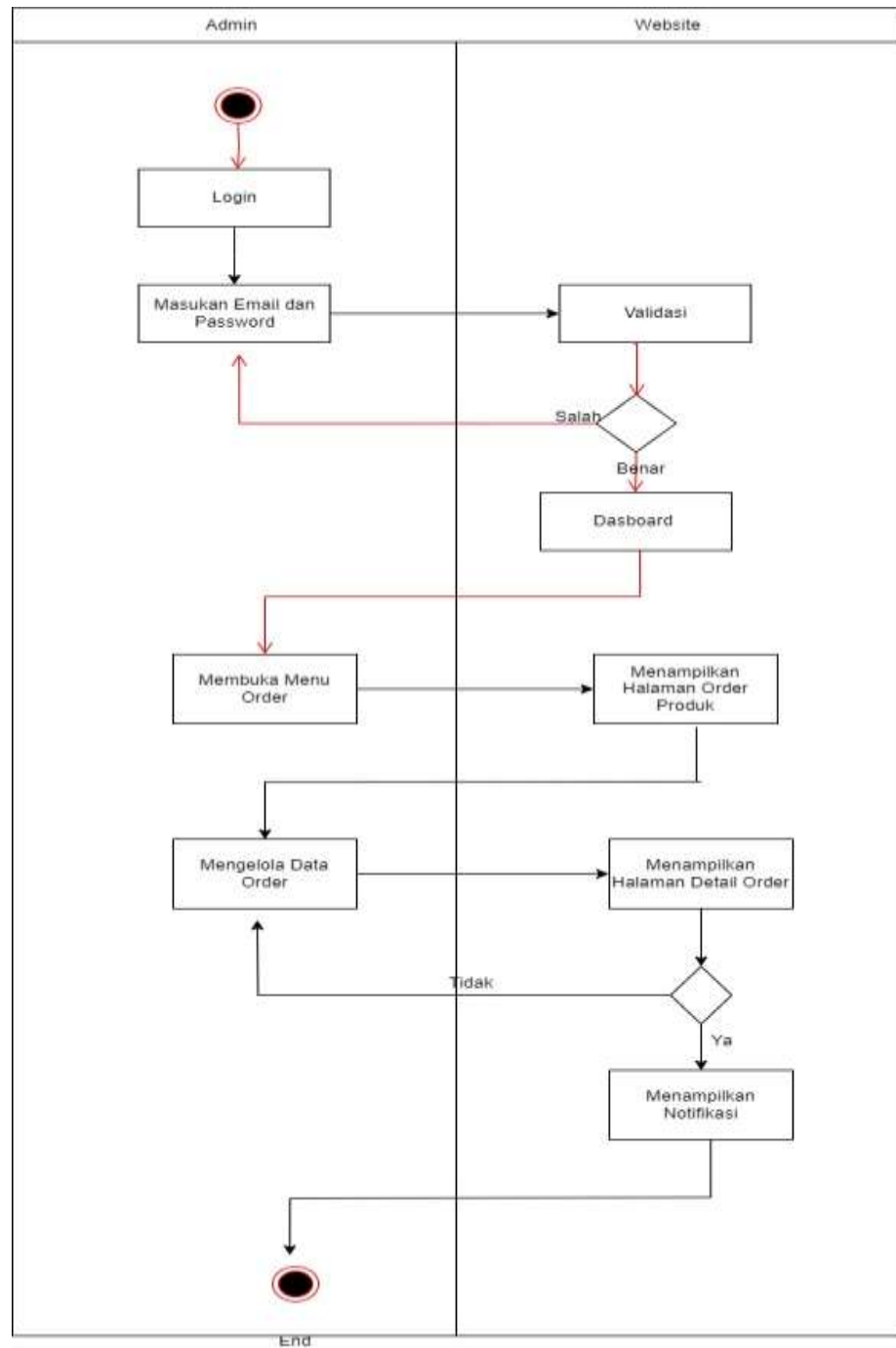
Gambar 5. 11Activity Diagram Admin

10. *Activity Diagram* data produk admin mengelola data produk yang diusulkan dapat dilihat sebagai berikut.



Gambar 5. 12Activity Diagram Admin

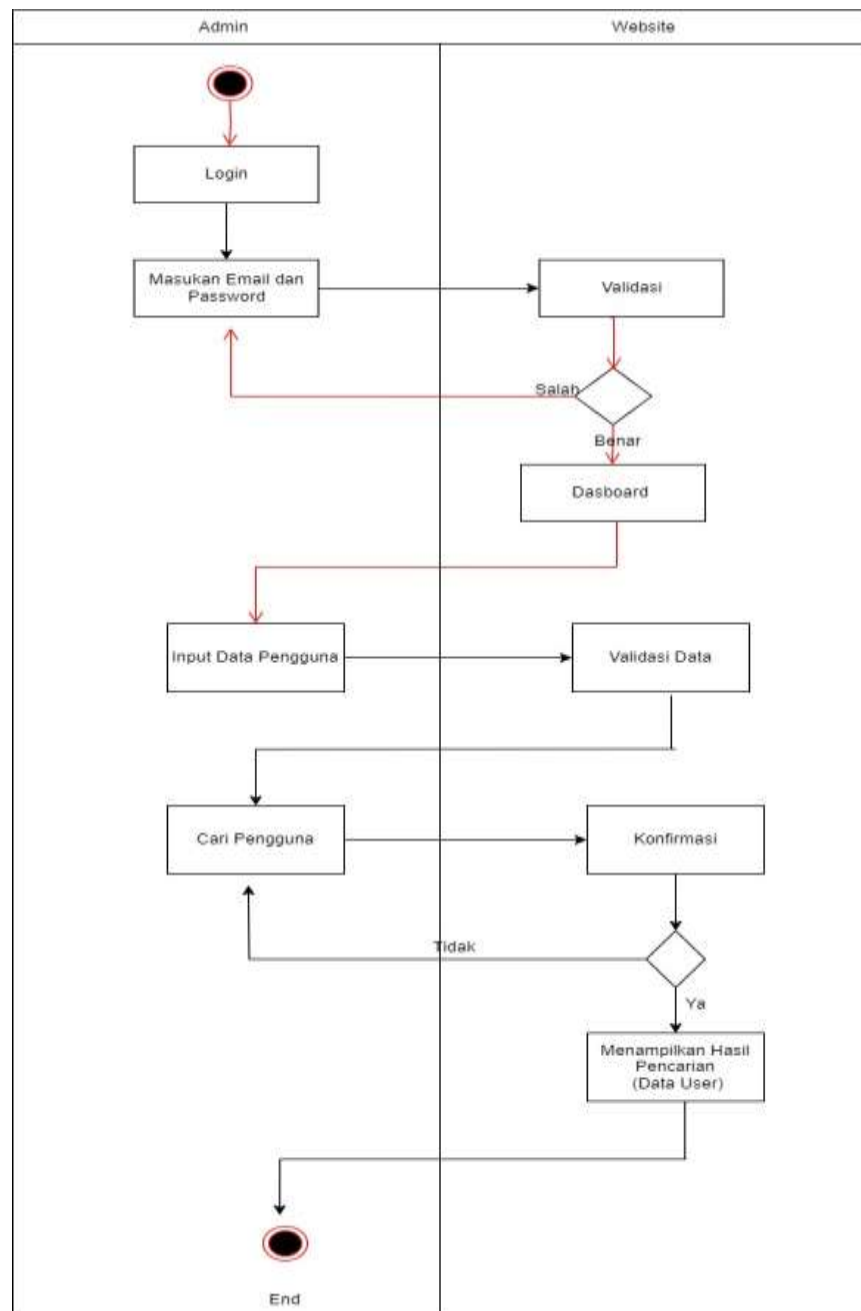
11. Activity Diagram data order admin mengelola data order yang diusulkan dapat dilihat sebagai berikut.



Gambar 5. 13Activity Diagram Admin

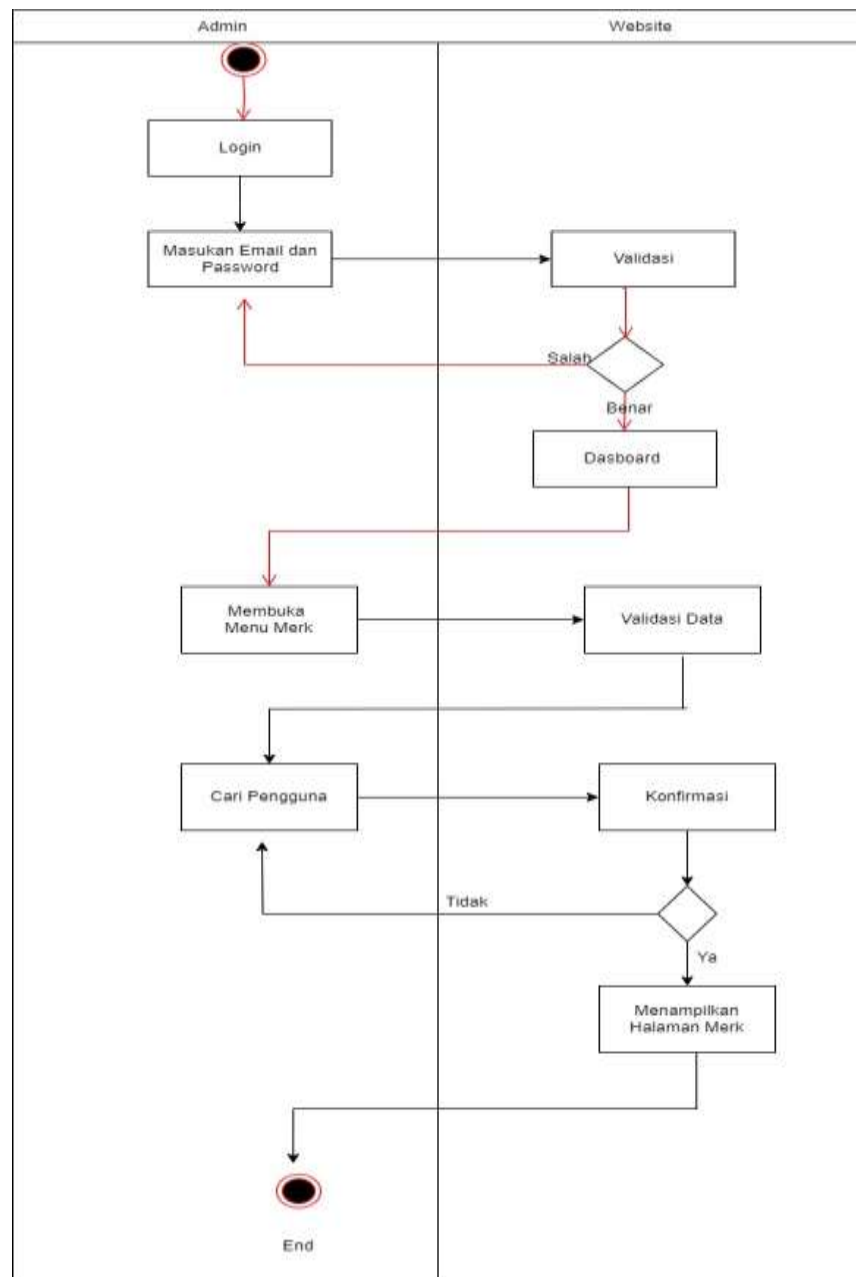
12. Activity Diagram Data User

Gambar *activity diagram* admin mengelola data user yang diusulkan dapat dilihat sebagai berikut.



Gambar 5. 14Activity Diagram Admin

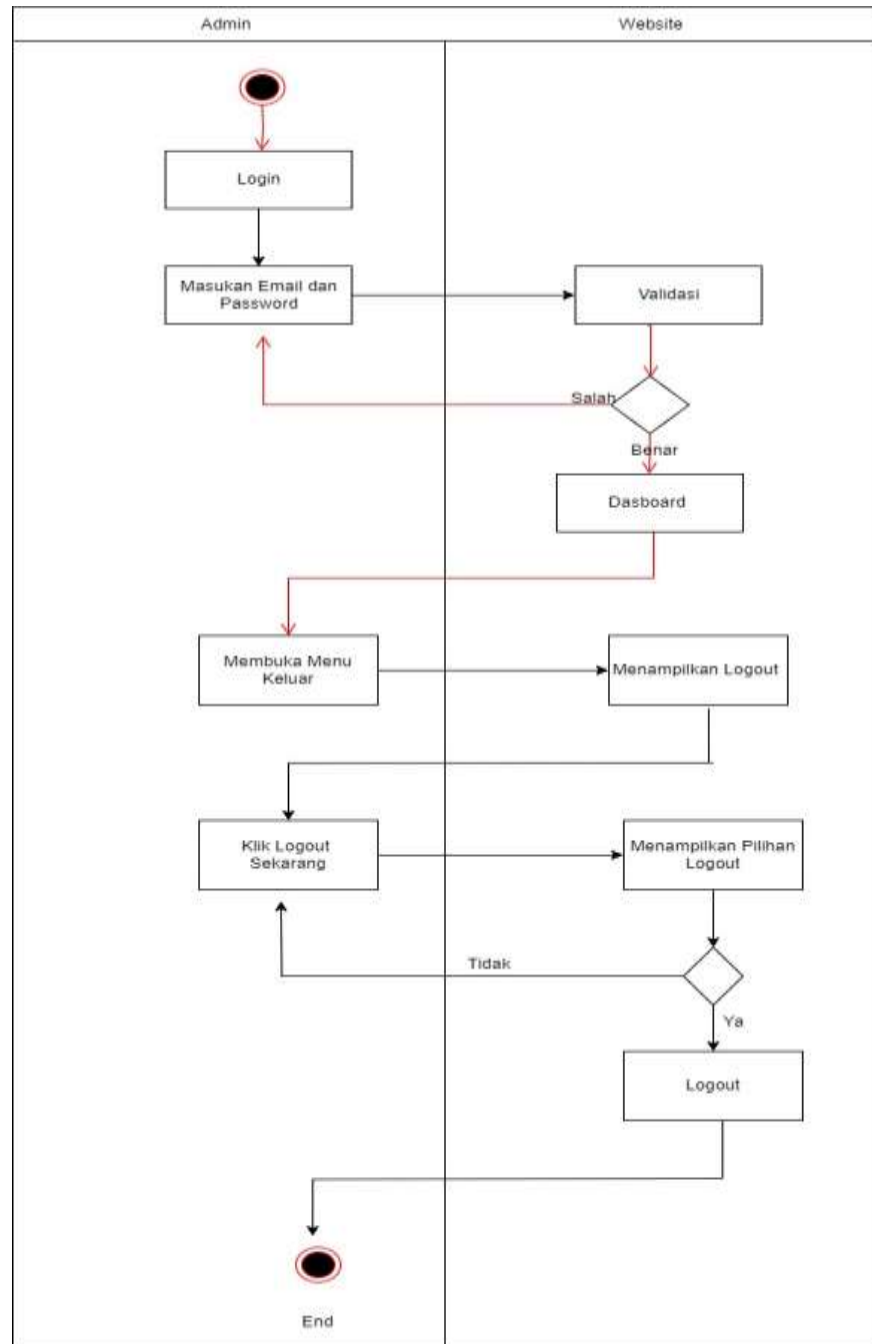
13. *Activity Diagram* Data Merk admin mengelola data merk yang diusulkan dapat dilihat sebagai berikut.



Gambar 5. 15Activity Diagram Admin

14. Activity Diagram

Gambar *activity diagram* logout yang diusulkan dapat dilihat sebagai berikut.

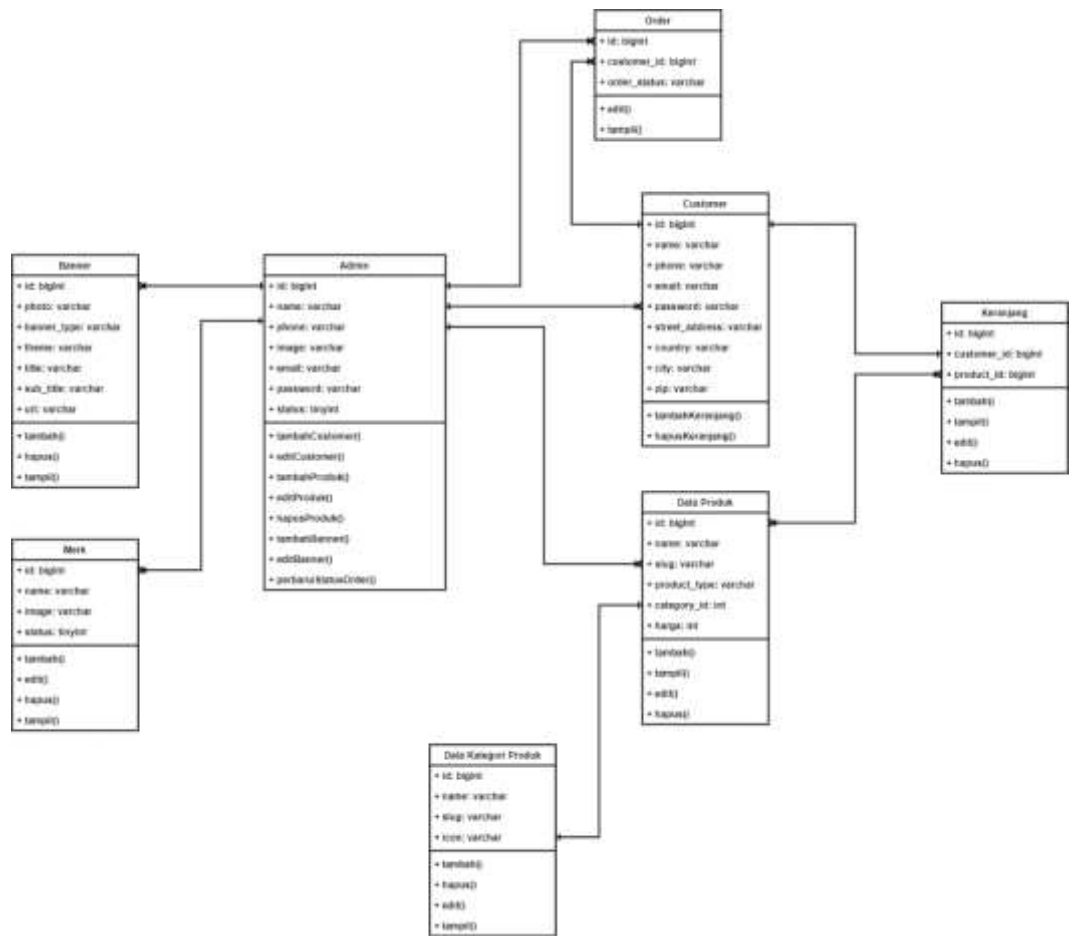


Gambar 5. 16Activity Diagram Logout

Activity Diagram menggambarkan *workflow* (aliran kerja) atau aktivitas dari sebuah sistem atau proses bisnis. Gambar 5.3. menjelaskan aktivitas mengelola data barang. Dimulai dengan klik menu data barang, mengelola data barang, yang dimana dapat melakukan tambah barang, mengubah barang, dan menghapus barang. Kemudian sistem akan melakukan validasi data. Jika data valid maka sistem akan menyimpan perubahan data barang ke database dan selesai. Jika tidak valid, sistem akan menampilkan notifikasi data gagal dikelola dan akan kembali ke halaman kelola data barang.

5.1.2.3 Class Diagram

Class diagram yang diusulkan untuk pembuatan *E-Catalog* Dapunta Inc ini dapat dilihat pada gambar 5.4.



Gambar 5. 17Class Diagram yang Diusulkan

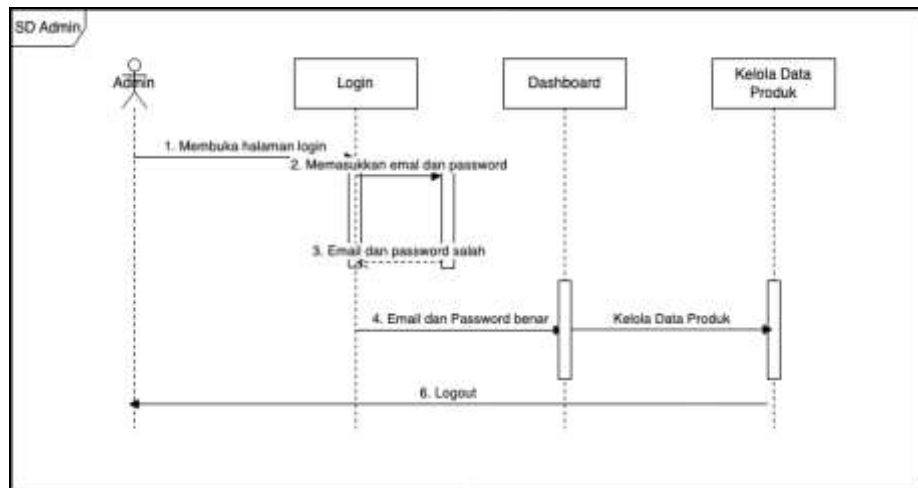
Class diagram atau diagram kelas menggambarkan struktur sistem dari segi pendefinisian dari *class-class* yang akan dibuat untuk membangun sistem. Adapun kelas-kelas yang ada pada *E-Catalog* Dapunta Inc adalah terdapat beberapa kelas, yaitu seperti kelas *user*, kategori produk, produk, transaksi dan *chat*.

5.1.2.4 Sequence Diagram

Sequence diagram yang diusulkan untuk pembuatan *E-Catalog* Dapunta Inc ini dapat dilihat pada gambar 5.4

1. Sequence Diagram

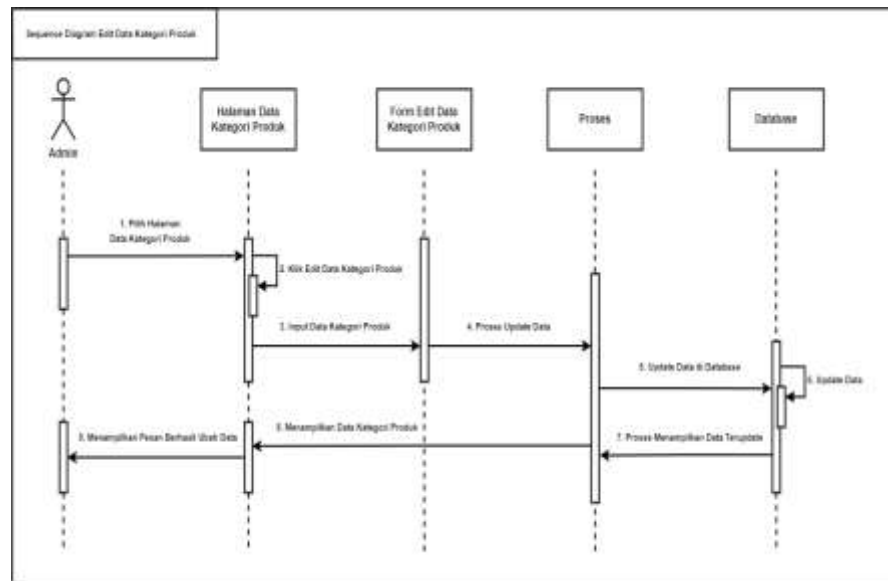
Gambar *sequence diagram* kelola data produk yang diusulkan dapat dilihat sebagai berikut.



Gambar 5. 18Sequence Diagram Kelola Data Produk

2. Sequence Diagram Edit Data Kategori Produk

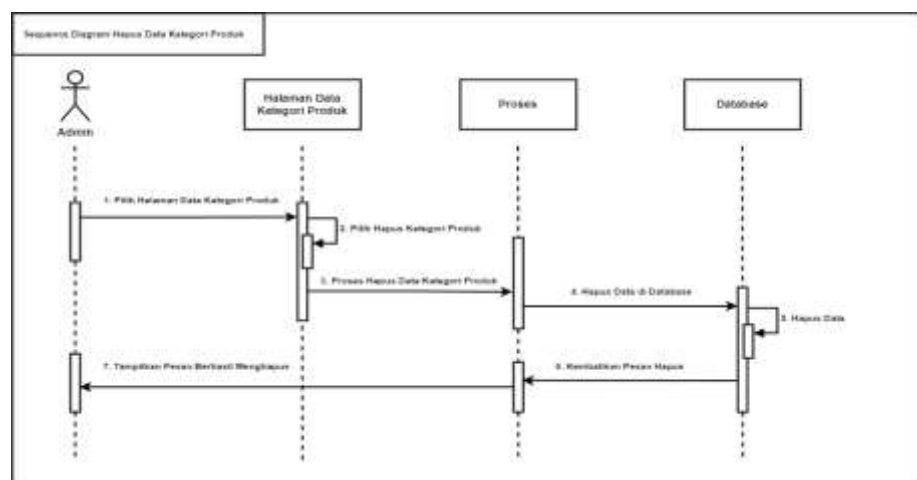
Gambar *sequence diagram* admin edit kategori produk yang diusulkan dapat dilihat pada gambar sebagai berikut.



Gambar 5. 19 Sequence Diagram Edit Data Kategori Produk

3. *Squence Diagram* Hapus Data Kategori Produk

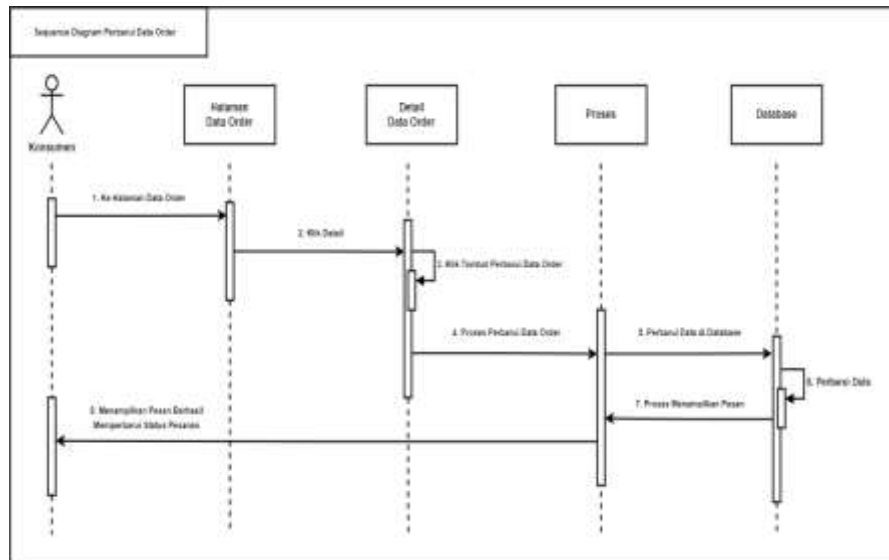
Gambar *squence diagram* admin hapus data kategori produk yang diusulkan dapat dilihat pada gambar sebagai berikut.



Gambar 5. 20Sequence Diagram Hapus Data Kategori Produk

4. *Sequence Diagram* Tambah Data Kategori Produk

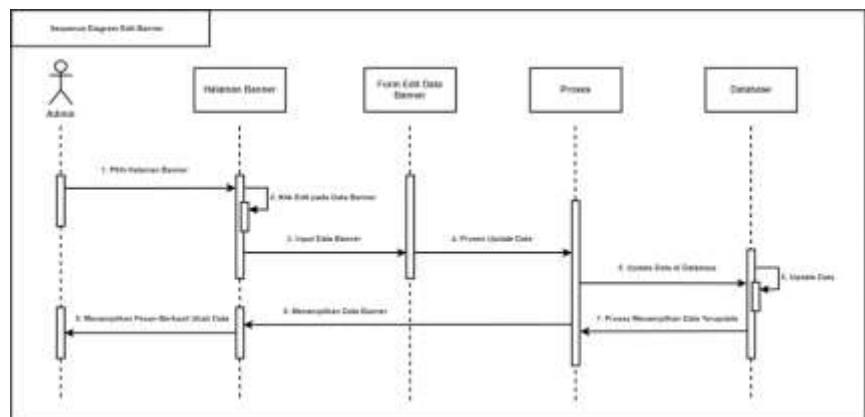
Gambar *sequence diagram* admin tambah data kategori produk yang diusulkan dapat dilihat pada gambar sebagai berikut.



Gambar 5. 21 *Sequence Diagram* Tambah Data Kategori Produk

5. *Sequence Diagram* Edit Banner

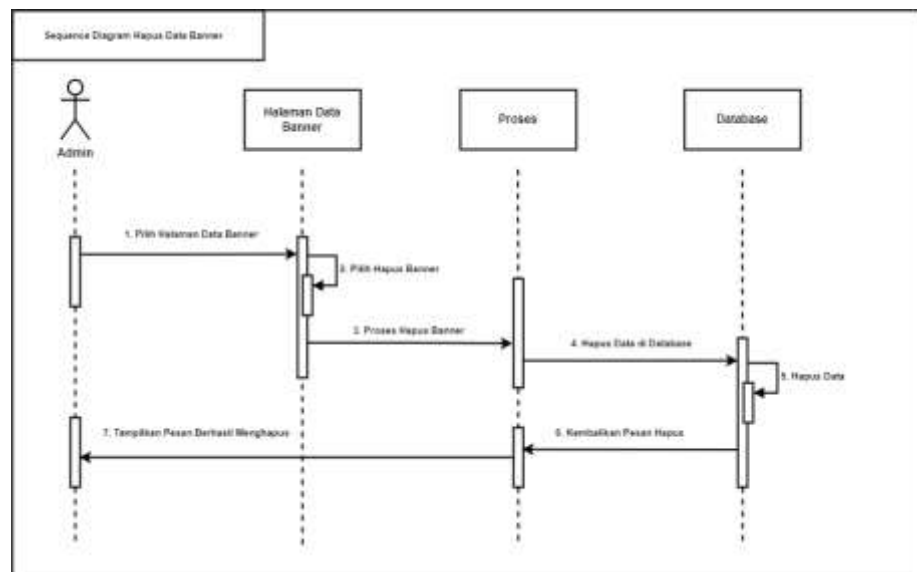
Gambar *sequence diagram* admin mengedit banner produk yang diusulkan dapat dilihat pada gambar sebagai berikut.



Gambar 5. 22 Gambar Sequence Diagram Edit Banner Produk

6. Squence Diagram Hapus Banner

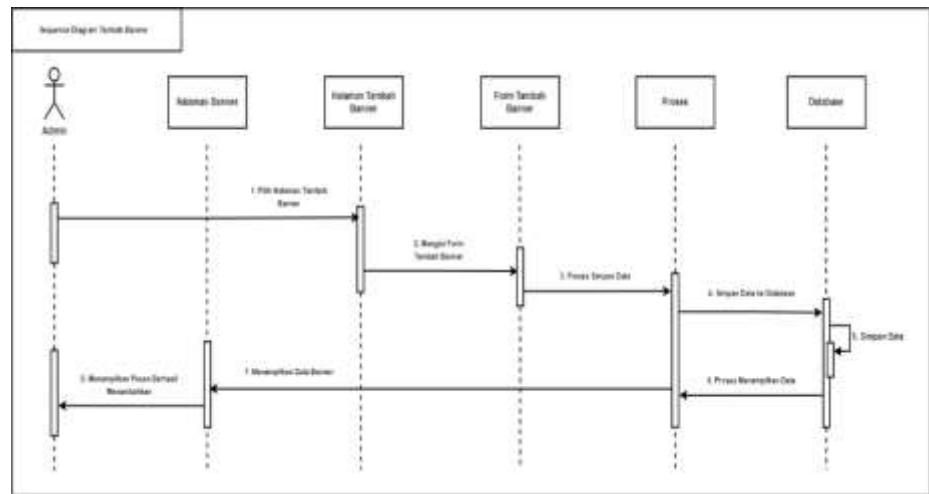
Gambar *squence diagram* admin hapus banner produk yang diusulkan dapat dilihat pada gambar sebagai berikut.



Gambar 5. 23 Sequence Diagram Edit Banner Produk

7. Squence Diagram Tambah Banner

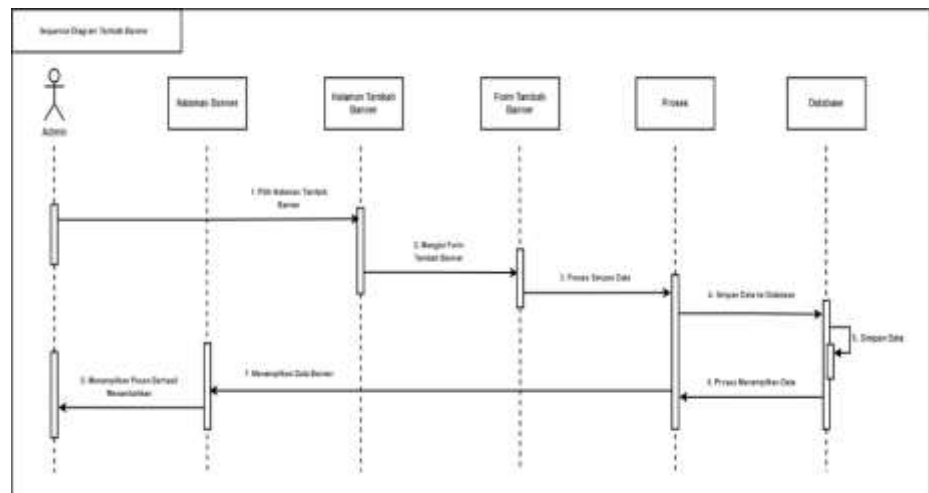
Gambar *squence diagram* admin tambah banner produk yang diusulkan dapat dilihat pada gambar sebagai berikut.



Gambar 5. 24 Sequence Diagram Tambah Banner

8. Sequence Diagram Edit Data Produk

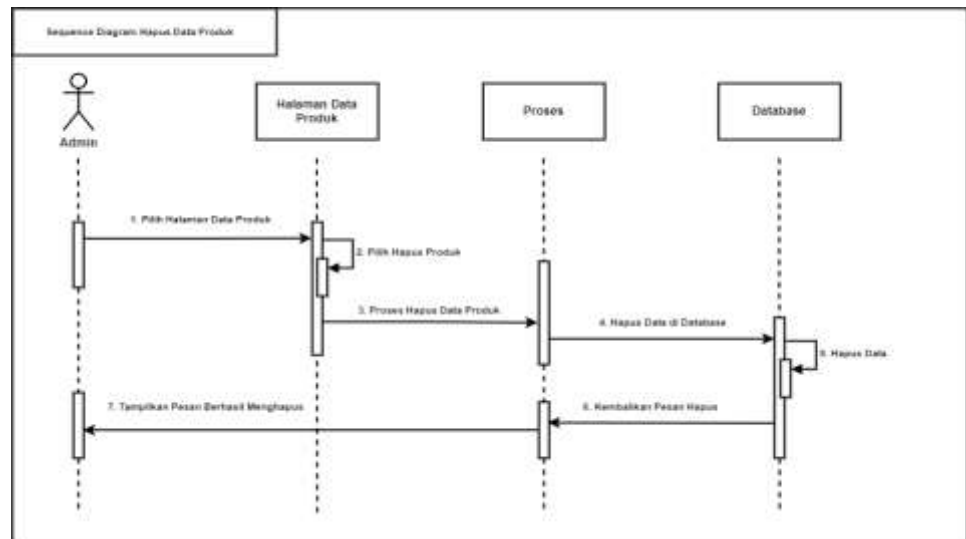
Gambar *sequence diagram* admin edit data produk yang diusulkan dapat dilihat pada gambar sebagai berikut.



Gambar 5. 25 Sequence Diagram Edit Data Produk

9. Sequence Diagram Hapus Data Produk

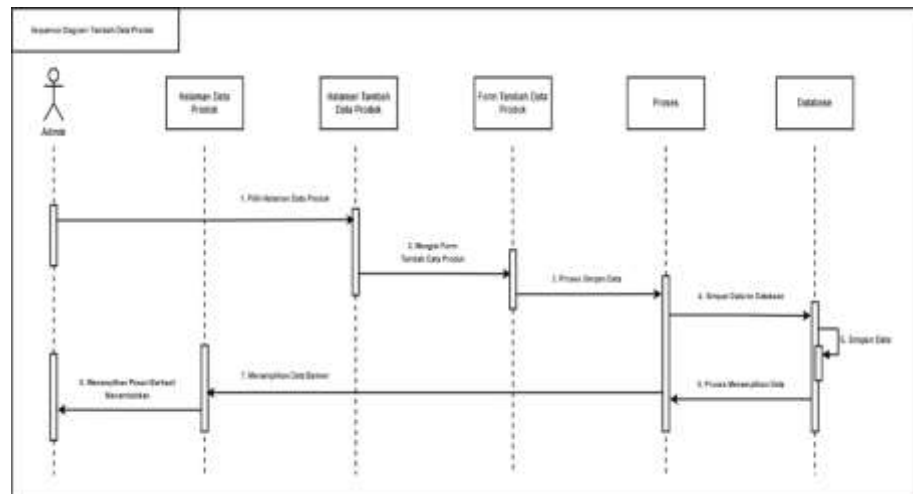
Gambar *sequence diagram* admin hapus data produk yang diusulkan dapat dilihat pada gambar sebagai berikut.



Gambar 5. 26Sequence Diagram Hapus Data Produk

10. Squence Diagram

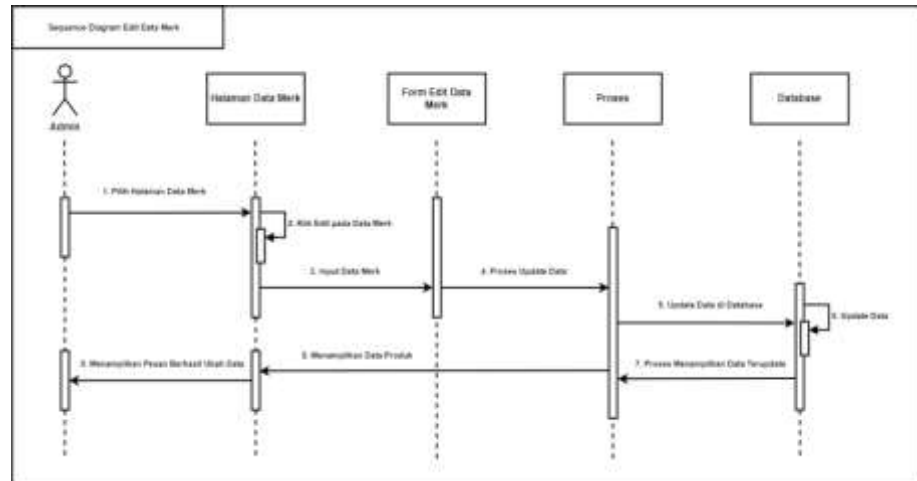
Gambar *squence diagram* admin tambah data produk yang diusulkan dapat dilihat pada gambar sebagai berikut.



Gambar 5. 27Sequence Diagram Tambah Data Produk

11. Squence Diagram Edit Data Merk

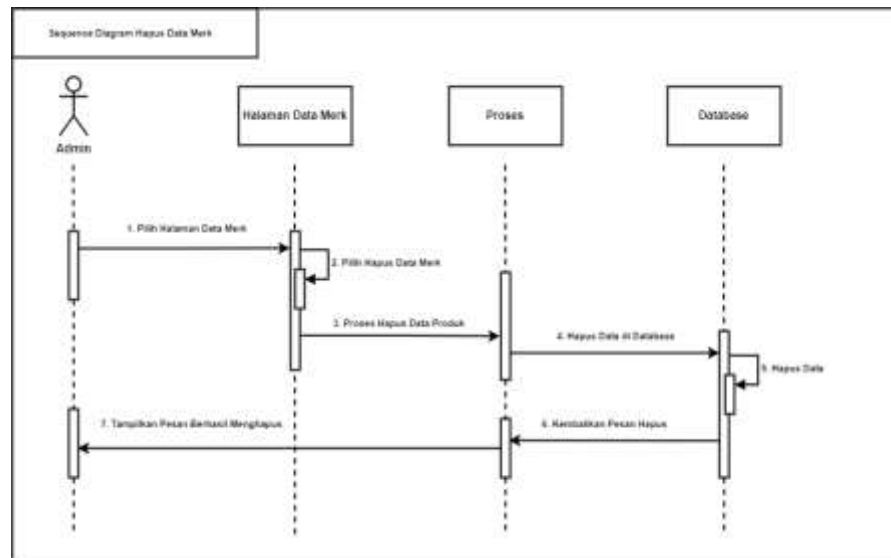
Gambar *sequence diagram* admin edit data merk yang diusulkan dapat dilihat pada gambar sebagai berikut.



Gambar 5. 28.Sequence Diagram Edit Data Merk

12. *Sequence Diagram* Hapus Data Merk

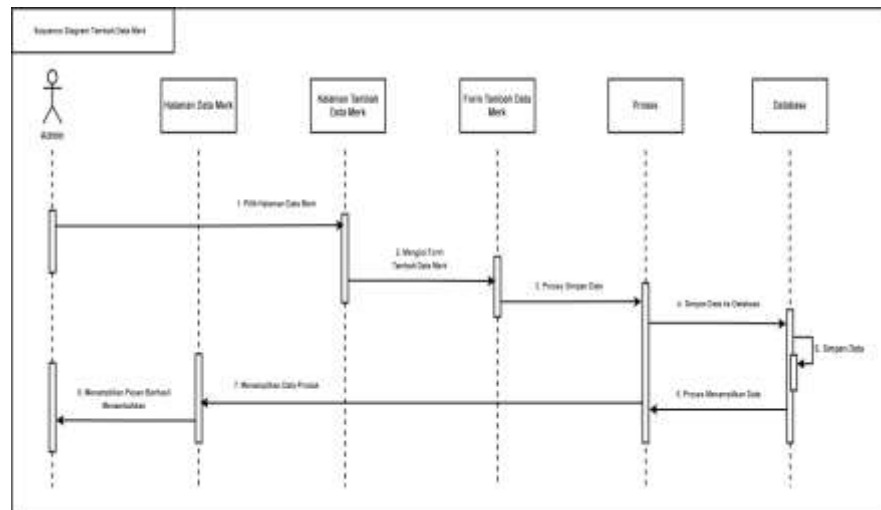
Gambar *sequence diagram* admin hapus data merk yang diusulkan dapat dilihat pada gambar sebagai berikut.



Gambar 5. 29.Sequence Diagram Hapus Data Merk

13. Sequence Diagram Tambah Data Merk

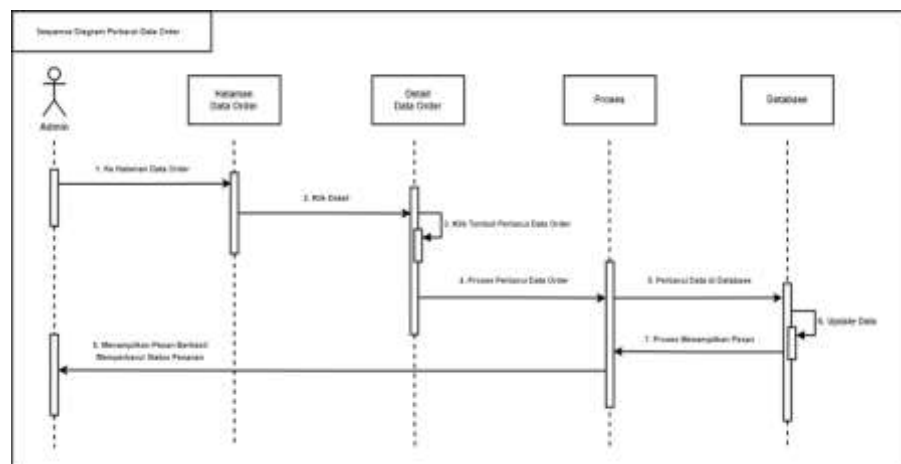
Gambar *sequence diagram* admin tambah data merk yang diusulkan dapat dilihat pada gambar sebagai berikut.



Gambar 5. 30. Sequence Diagram Tambah Data Merk

14. Sequence Diagram Perbarui Status Data Order

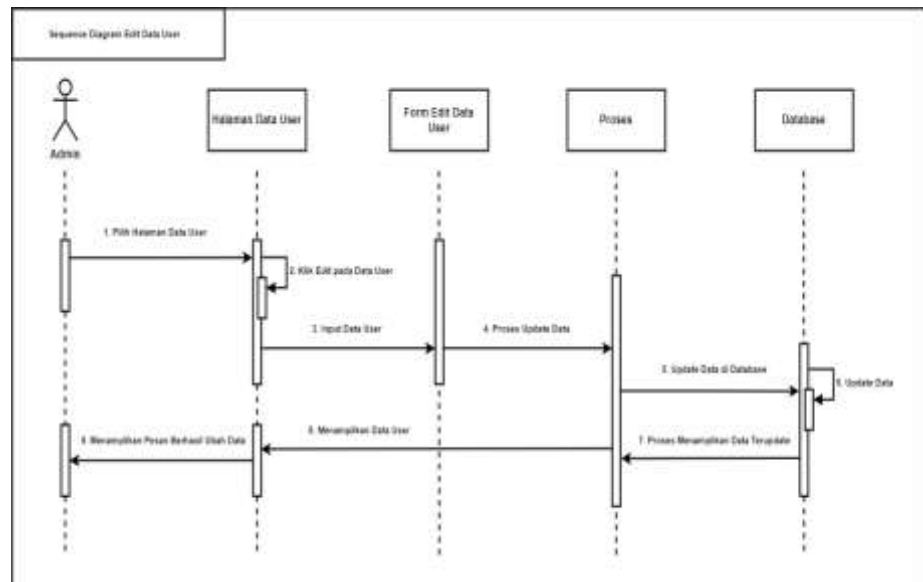
Gambar *sequence diagram* admin perbarui status data order produk yang diusulkan dapat dilihat pada gambar sebagai berikut.



Gambar 5. 31. Sequence Diagram Perbarui Status Data Order

15. Sequence Diagram Edit Data User

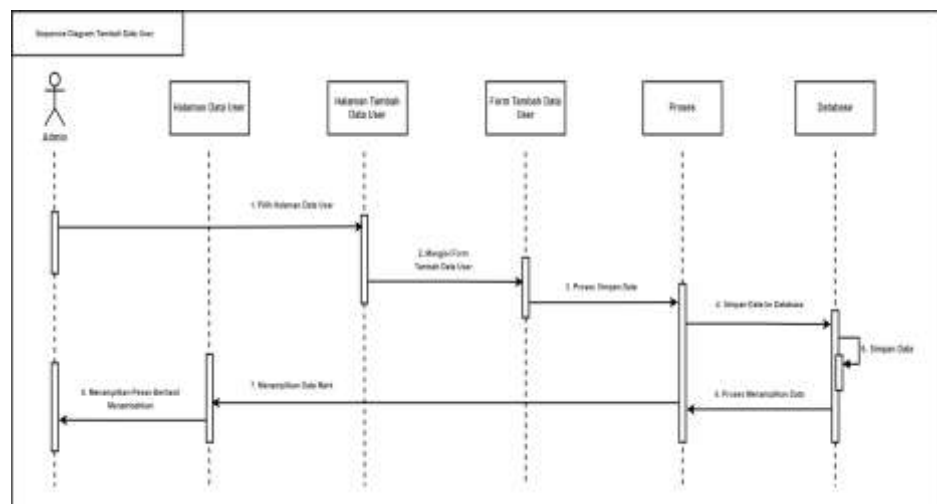
Gambar *sequence diagram* admin edit data user yang diusulkan dapat dilihat pada gambar sebagai berikut.



Gambar 5. 32. Sequence Diagram Edit Data User

16. Sequence Diagram Tambah Data User

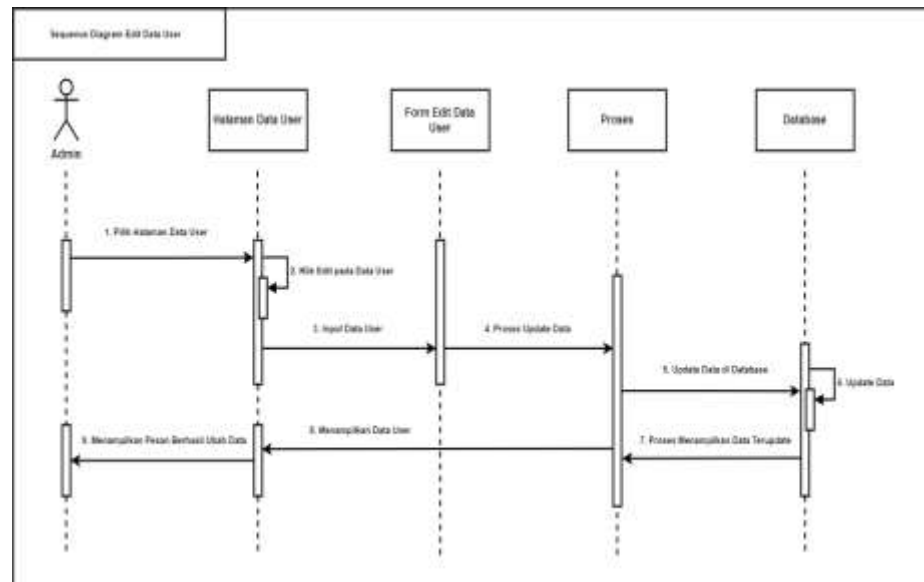
Gambar *sequence diagram* admin tambah data user yang diusulkan dapat dilihat pada gambar sebagai berikut.



Gambar 5. 33. Sequence Diagram Tambah Data User

17. *Sequence Diagram* Konsumen Checkout

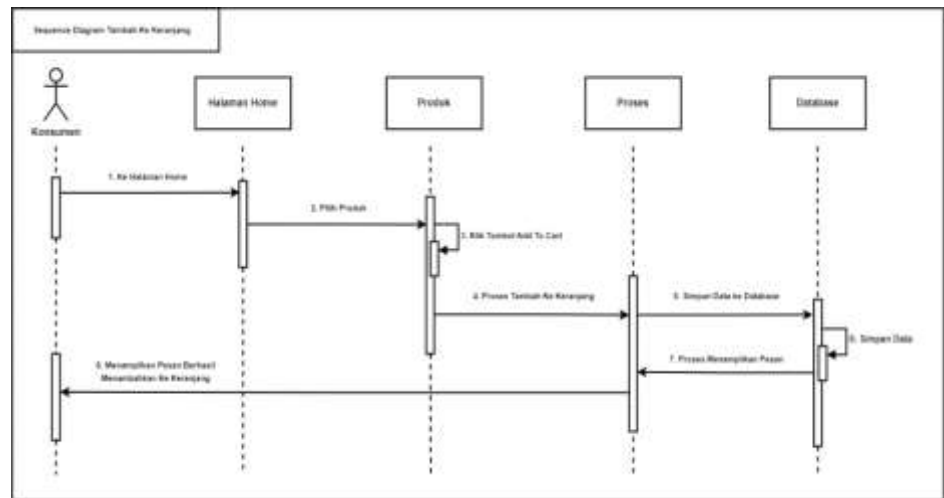
Gambar *sequence diagram* konsumen *checkout* yang diusulkan dapat dilihat pada gambar sebagai berikut.



Gambar 5. 34. Sequence Diagram Konsumen Checkout

18. *Sequence Diagram* Tambah Keranjang

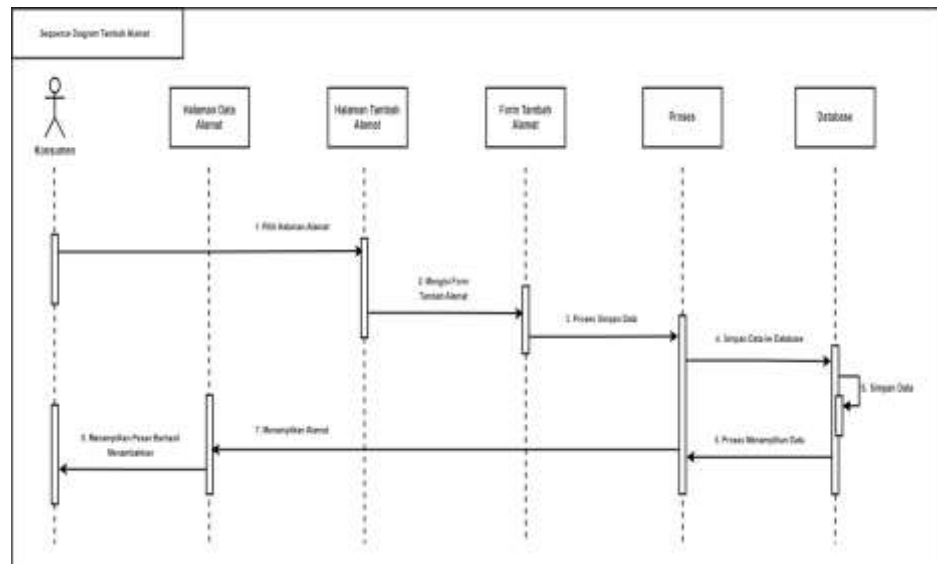
Gambar *sequence diagram* konsumen tambah ke keranjang yang diusulkan dapat dilihat pada gambar sebagai berikut.



Gambar 5. 35. Sequence Diagram Tambah Keranjang

19. Sequence Diagram Tambah Alamat

Gambar *sequence diagram* konsumen tambah alamat yang diusulkan dapat dilihat pada gambar sebagai berikut.



Gambar 5. 36. Sequence Diagram Konsumen Tambah Alamat

Sequence Diagram mendeskripsikan waktu hidup objek dan *message* yang dikirimkan dan diterima antar objek.

Gambar 5.4. menunjukkan alur dimana konsumen melakukan pencarian di katalog, lalu diproses oleh server, kemudian server mengolah data di *database* dan mengirimkan hasil pencarian dari *database* ke antarmuka/halaman katalog yang akan ditampilkan ke konsumen.

5.1.2.5 Desain Database

Desain *database* dilakukan oleh penulis dalam pembuatan aplikasi pelayanan terpadu satu pintu dalam rangka mempermudah proses pengembangan aplikasi. Beberapa tabel tersebut adalah sebagai berikut ini:

1. Tabel Admin

Tabel admin digunakan untuk menampung data admin dan dapat dilihat pada tabel 5.1.

Nama tabel : admin

Primary key : id

Foreign key : -

Tabel 5.1. Tabel admin

No	Nama <i>Field</i>	Tipe	Ukuran	Keterangan
----	-------------------	------	--------	------------

No	Nama Field	Tipe	Ukuran	Keterangan
1	<i>Id</i>	<i>BigInt</i>	20	id admin, <i>primary key</i>
2	<i>Name</i>	<i>Varchar</i>	80	nama admin
3	<i>Phone</i>	<i>Varchar</i>	25	nomor <i>handphone</i>
4	<i>admin_role_id</i>	<i>BigInt</i>	20	<i>id role admin</i>
5	<i>Image</i>	<i>Varchar</i>	30	Gambar
6	<i>identify_image</i>	<i>Text</i>	-	gambar identifikasi
7	<i>identify_type</i>	<i>Varchar</i>	255	tipe identifikasi
8	<i>identify_number</i>	<i>Int</i>	11	nomor identifikasi
9	<i>Email</i>	<i>Varchar</i>	80	<i>Email</i>
10	<i>email_verified_at</i>	<i>Timestamp</i>	-	waktu verifikasi <i>email</i>
11	<i>Password</i>	<i>Varchar</i>	80	kata sandi
12	<i>remember_token</i>	<i>Varchar</i>	100	<i>token</i> pengingat
13	<i>created_at</i>	<i>Timestamp</i>	-	waktu dibuat
14	<i>updated_at</i>	<i>Timestamp</i>	-	waktu diperbaharui
15	<i>Status</i>	<i>Tinyint</i>	1	Status

2. Tabel Admin Role

Tabel *admin role* digunakan untuk menampung data *admin role* dan dapat dilihat pada tabel 5.2.

Nama tabel : *admin_roles*

Primary key : *id*

Foreign key : -

Tabel 5.2. Tabel *admin_roles*

No	Nama <i>Field</i>	Tipe	Ukuran	Keterangan
1	<i>id</i>	<i>BigInt</i>	20	id admin roles, <i>primary key</i>
2	<i>name</i>	<i>Varchar</i>	30	nama role admin
3	<i>modules_access</i>	<i>Varchar</i>	250	akses modul
4	<i>status</i>	<i>TinyInt</i>	1	Status
5	<i>created_at</i>	<i>Timestamp</i>	-	waktu dibuat
6	<i>updated_at</i>	<i>Timestamp</i>	-	waktu diperbaharui

3. Tabel Atribut

Tabel atribut digunakan untuk menampung data admin dan dapat dilihat pada tabel 5.3.

Nama tabel : atribut

Primary key : id

Foreign key : -

Tabel 5.3. Tabel atribut

No	Nama <i>Field</i>	Tipe	Ukuran	Keterangan
1	<i>id</i>	<i>BigInt</i>	20	id atribut, <i>primary key</i>
2	<i>name</i>	<i>Varchar</i>	30	nama atribut
3	<i>created_at</i>	<i>Timestamp</i>	-	waktu dibuat
4	<i>updated_at</i>	<i>Timestamp</i>	-	waktu diperbaharui

4. Tabel Banner

Tabel *banner* digunakan untuk menampung data *banner* dan dapat dilihat pada tabel 5.4.

Nama tabel : *banner*

Primary key : id

Foreign key : -

Tabel 5.4. Tabel *banner*

No	Nama <i>Field</i>	Tipe	Ukuran	Keterangan
1	<i>Id</i>	<i>BigInt</i>	20	id banner, <i>primary key</i>
2	<i>Photo</i>	<i>Varchar</i>	255	foto <i>banner</i>
3	<i>banner_type</i>	<i>Varchar</i>	250	akses modul
4	<i>Theme</i>	<i>TinyInt</i>	1	Status
5	<i>Published</i>	<i>Int</i>	11	Dipublikasi
6	<i>created_at</i>	<i>Timestamp</i>	-	waktu dibuat
7	<i>updated_at</i>	<i>Timestamp</i>	-	waktu diperbaharui
8	<i>url</i>	<i>Varchar</i>	255	url
9	<i>resource_type</i>	<i>Varchar</i>	191	tipe resource
10	<i>resource_id</i>	<i>BigInt</i>	20	id resource
11	<i>Title</i>	<i>Varchar</i>	191	judul banner
12	<i>sub_title</i>	<i>Varchar</i>	191	sub title
13	<i>button_text</i>	<i>Varchar</i>	191	teks tombol
14	<i>background_color</i>	<i>Varchar</i>	191	warna latar belakang

5. Tabel Cart

Tabel *cart* digunakan untuk menampung data produk di dalam keranjang dan dapat dilihat pada tabel 5.5.

Nama tabel : *cart*

Primary key : id

Foreign key : *customer_id*

Tabel 5.5. Tabel *cart*

No	Nama <i>Field</i>	Tipe	Ukuran	Keterangan
1	<i>Id</i>	<i>BigInt</i>	20	id keranjang, <i>primary key</i>
2	<i>customer_id</i>	<i>BigInt</i>	20	id konsumen, <i>foreign key</i>
3	<i>cart_group_id</i>	<i>Varchar</i>	191	id grup keranjang
4	<i>product_id</i>	<i>BigInt</i>	20	
5	<i>product_type</i>	<i>Varchar</i>	20	
6	<i>digital_product_type</i>	<i>Varchar</i>	30	
7	<i>Color</i>	<i>Varchar</i>	191	
8	<i>Choices</i>	<i>Text</i>		
9	<i>Variations</i>	<i>Text</i>		
10	<i>Variant</i>	<i>Text</i>		
11	<i>Quantity</i>	<i>Int</i>	11	Kuantitas
12	<i>Price</i>	<i>double</i>	-	Harga
13	<i>Tax</i>	<i>double</i>	-	Pajak
14	<i>Discount</i>	<i>double</i>	-	Diskon
15	<i>tax_model</i>	<i>Varchar</i>	20	model pajak
16	<i>is_checked</i>	<i>Tinyint</i>	1	Cek
17	<i>Slug</i>	<i>Varchar</i>	191	Slug
18	<i>Name</i>	<i>Varchar</i>	191	Nama
19	<i>Thumbnail</i>	<i>Varchar</i>	191	Gambar
20	<i>seller_id</i>	<i>BigInt</i>	20	Id penjual
21	<i>seller_is</i>	<i>Varchar</i>	191	Penjual
22	<i>created_at</i>	<i>Timestamp</i>	-	Waktu dibuat

No	Nama <i>Field</i>	Tipe	Ukuran	Keterangan
23	<i>updated_at</i>	<i>Timestamp</i>	-	Waktu pembaharuan
24	<i>shop_info</i>	<i>Varchar</i>	191	info toko
25	<i>shipping_cost</i>	<i>Double</i>	8,2	biaya pengiriman
26	<i>shipping_type</i>	<i>Varchar</i>	191	tipe pengiriman
27	<i>is_guest</i>	<i>Tinyint</i>	4	Tamu

6. Tabel Order

Tabel *order* digunakan untuk menampung data pembelian dan dapat dilihat pada tabel 5.6.

Nama tabel : *order*

Primary key : *id*

Foreign key : *customer_id*

Tabel 5.6. Tabel *order*

No	Nama <i>Field</i>	Tipe	Ukuran	Keterangan
1	<i>id</i>	<i>BigInt</i>	20	id order, <i>primary key</i>
2	<i>customer_id</i>	<i>BigInt</i>	20	id konsumen, <i>foreign key</i>
3	<i>is_guest</i>	<i>TinyInt</i>	4	Tamu
4	<i>customer_type</i>	<i>Varchar</i>	10	tipe konsumen
5	<i>payment_status</i>	<i>Varchar</i>	15	status pembayaran
6	<i>order_status</i>	<i>Varchar</i>	50	status order
7	<i>payment_method</i>	<i>Varchar</i>	100	metode pembayaran
8	<i>transaction_ref</i>	<i>Varchar</i>	30	referensi transaksi
9	<i>payment_by</i>	<i>Varchar</i>	191	pembayaran dengan

No	Nama <i>Field</i>	Tipe	Ukuran	Keterangan
10	<i>payment_note</i>	<i>Text</i>	-	catatan pembayaran
11	<i>order_amount</i>	<i>Double</i>	-	jumlah order
12	<i>created_at</i>	<i>Timestamp</i>	-	waktu dibuat
13	<i>updated_at</i>	<i>Timestamp</i>	-	waktu diperbaharui

7. Tabel Order Detail

Tabel *order detail* digunakan untuk menampung data detail pembelian dan dapat dilihat pada tabel 5.7.

Nama tabel : *order detail*

Primary key : *id*

Foreign key : *customer_id, product_id*

Tabel 5.7. Tabel *order detail*

No	Nama <i>Field</i>	Tipe	Ukuran	Keterangan
1	<i>id</i>	<i>BigInt</i>	20	<i>id order detail, primary key</i>
2	<i>order_id</i>	<i>BigInt</i>	20	<i>id order, foreign key</i>
3	<i>product_id</i>	<i>TinyInt</i>	4	<i>id produk, foreign key</i>
4	<i>product_details</i>	<i>Text</i>	-	detail produk
5	<i>qty</i>	<i>Int</i>	11	Kuantitas
6	<i>price</i>	<i>double</i>	-	Harga
7	<i>tax</i>	<i>double</i>	-	Pajak
8	<i>discount</i>	<i>double</i>	-	Diskon
9	<i>delivery_status</i>	<i>Varchar</i>	15	status pengiriman
10	<i>payment_status</i>	<i>Varchar</i>	15	status pembayaran

No	Nama <i>Field</i>	Tipe	Ukuran	Keterangan
11	<i>created_at</i>	<i>Timestamp</i>	-	waktu dibuat
12	<i>updated_at</i>	<i>Timestamp</i>	-	waktu diperbaharui

8. Tabel Produk

Tabel produk digunakan untuk menampung data produk dan dapat dilihat pada tabel 5.8.

Nama tabel : produk

Primary key : id

Foreign key : *user_id, category_id, brand_id*

Tabel 5.8. Tabel produk

No	Nama <i>Field</i>	Tipe	Ukuran	Keterangan
1	<i>id</i>	<i>BigInt</i>	20	id produk, <i>primary key</i>
2	<i>added_by</i>	<i>Varchar</i>	191	Ditambahkan
3	<i>user_id</i>	<i>BigInt</i>	20	id user, <i>foreign key</i>
4	<i>name</i>	<i>Text</i>	-	Nama
5	<i>slug</i>	<i>Int</i>	11	Slug
6	<i>product_type</i>	<i>double</i>	-	tipe produk
7	<i>category_ids</i>	<i>BigInt</i>	20	id kategori
8	<i>category_id</i>	<i>BigInt</i>	20	id kategori, <i>foreign key</i>
9	<i>sub_category_id</i>	<i>BigInt</i>	20	id sub kategori
10	<i>sub_sub_category_id</i>	<i>BigInt</i>	20	id sub sub kategori
11	<i>brand_id</i>	<i>BigInt</i>	20	id merk, <i>foreign key</i>
12	<i>unit</i>	<i>Varchar</i>	191	

No	Nama <i>Field</i>	Tipe	Ukuran	Keterangan
13	<i>min_qty</i>	<i>Int</i>	11	minimal kuantitas
14	<i>images</i>	<i>Varchar</i>	191	Gambar
15	<i>thumbnail</i>	<i>Varchar</i>	191	Gambar
16	<i>created_at</i>	<i>Timestamp</i>	-	waktu dibuat
17	<i>updated_at</i>	<i>Timestamp</i>	-	waktu diperbaharui

9. Tabel Product Stock

Tabel stok produk digunakan untuk menampung data stok produk dan dapat dilihat pada tabel 5.9.

Nama tabel : stok produk

Primary key : id

Foreign key : *product_id*

Tabel 5.9. Tabel stok produk

No	Nama <i>Field</i>	Tipe	Ukuran	Keterangan
1	<i>id</i>	<i>BigInt</i>	20	id stok produk, <i>primary key</i>
2	<i>product_id</i>	<i>BigInt</i>	20	id produk, <i>foreign key</i>
3	<i>variant</i>	<i>Varchar</i>	255	Variasi
4	<i>sku</i>	<i>Varchar</i>	255	SKU
5	<i>price</i>	<i>Decimal</i>	8,2	Harga
6	<i>qty</i>	<i>Int</i>	11	Kuantitas
7	<i>created_at</i>	<i>Timestamp</i>	-	waktu dibuat
8	<i>updated_at</i>	<i>Timestamp</i>	-	waktu diperbaharui

10. Tabel Product Tag

Tabel tag produk digunakan untuk menampung data tag produk dan dapat dilihat pada tabel 5.10.

Nama tabel : tag produk

Primary key : id

Foreign key : *product_id, tag_id*

Tabel 5.10. Tabel tag produk

No	Nama <i>Field</i>	Tipe	Ukuran	Keterangan
1	<i>id</i>	<i>BigInt</i>	20	id tag produk, <i>primary key</i>
2	<i>product_id</i>	<i>BigInt</i>	20	id produk, <i>foreign key</i>
3	<i>tag_id</i>	<i>Varchar</i>	255	id tag, <i>foreign key</i>
4	<i>created_at</i>	<i>Timestamp</i>	-	waktu dibuat
5	<i>updated_at</i>	<i>Timestamp</i>	-	waktu diperbaharui

11. Tabel Reviews

Tabel *review* digunakan untuk menampung data *review* produk dan dapat dilihat pada tabel 5.11.

Nama tabel : *review* produk

Primary key : id

Foreign key : *product_id, customer_id, order_id*

Tabel 5.11. Tabel review produk

No	Nama <i>Field</i>	Tipe	Ukuran	Keterangan
----	-------------------	------	--------	------------

No	Nama <i>Field</i>	Tipe	Ukuran	Keterangan
1	<i>id</i>	<i>BigInt</i>	20	id tag produk, <i>primary key</i>
2	<i>product_id</i>	<i>BigInt</i>	20	id produk, <i>foreign key</i>
3	<i>customer_id</i>	<i>BigInt</i>	20	id konsumen, <i>foreign key</i>
4	<i>order_id</i>	<i>BigInt</i>	20	id order, <i>foreign key</i>
5	<i>comment</i>	<i>Text</i>	-	komentar
6	<i>attachment</i>	<i>Text</i>	-	attachment
7	<i>rating</i>	<i>Int</i>	11	rating
8	<i>status</i>	<i>Int</i>	11	status
9	<i>created_at</i>	<i>Timestamp</i>	-	waktu dibuat
10	<i>updated_at</i>	<i>Timestamp</i>	-	waktu diperbaharui

12. Tabel Tags

Tabel *tags* digunakan untuk menampung data *tags* dan dapat dilihat pada tabel 5.12.

Nama tabel : *tags*

Primary key : *id*

Foreign key : -

Tabel 5.12. Tabel tag

No	Nama <i>Field</i>	Tipe	Ukuran	Keterangan
1	<i>id</i>	<i>BigInt</i>	20	id tag produk, <i>primary key</i>
2	<i>tag</i>	<i>Varchar</i>	191	tag
3	<i>visit_count</i>	<i>BigInt</i>	20	hitungan kunjungan
4	<i>created_at</i>	<i>Timestamp</i>	-	waktu dibuat

No	Nama <i>Field</i>	Tipe	Ukuran	Keterangan
5	<i>updated_at</i>	<i>Timestamp</i>	-	waktu diperbaharui

13. Tabel *Transaction*

Tabel *transaction* digunakan untuk menampung data transaksi dan dapat dilihat pada tabel 5.13.

Nama tabel : transaksi

Primary key : id

Foreign key : *order_id*

Tabel 5.13. Tabel transaksi

No	Nama <i>Field</i>	Tipe	Ukuran	Keterangan
1	<i>id</i>	<i>BigInt</i>	20	id tag produk, <i>primary key</i>
2	<i>order_id</i>	<i>BigInt</i>	20	id order, <i>foreign key</i>
3	<i>payment_for</i>	<i>BigInt</i>	20	pembayaran
4	<i>amount</i>	<i>Double</i>	8,2	total
5	<i>created_at</i>	<i>Timestamp</i>	-	waktu dibuat
6	<i>updated_at</i>	<i>Timestamp</i>	-	waktu diperbaharui

14. Tabel *Wishlist*

Tabel *wishlist* digunakan untuk menampung data keinginan dan dapat dilihat pada tabel 5.14.

Nama tabel : *wishlist*

Primary key : id

Foreign key : *order_id*

Tabel 5.14. Tabel *wishlist*

No	Nama <i>Field</i>	Tipe	Ukuran	Keterangan
1	<i>id</i>	<i>BigInt</i>	20	id tag produk, <i>primary key</i>
2	<i>customer_id</i>	<i>BigInt</i>	20	id konsumen, <i>foreign key</i>
3	<i>product_id</i>	<i>BigInt</i>	20	id produk, <i>foreign key</i>
4	<i>created_at</i>	<i>Timestamp</i>	-	waktu dibuat
5	<i>updated_at</i>	<i>Timestamp</i>	-	waktu diperbaharui

5.1.2.6 Desain Interface

1. Login Admin

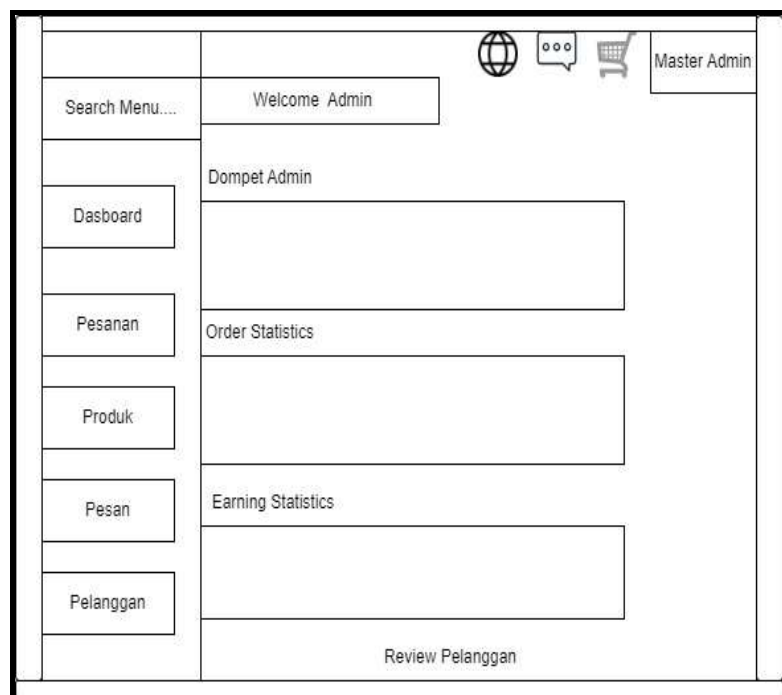
Berikut desain *Interface* admin yang digunakan untuk masuk ke halaman login. Desain Form login dapat dilihat pada desain login dapat dilihat pada gambar dibawah ini.

The image shows a wireframe of an admin login page. At the top center, the word "Masuk" is displayed in a bold font. Below it, the text "(Admin Login)" is centered. The form consists of three vertically stacked input fields. The first field is labeled "Your Email" and is a wide rectangular box. The second field is labeled "Kata sandi" and is also a wide rectangular box. The third field is a wide rectangular box containing the text "Masuk" centered within it, serving as the login button.

Gambar 5. 37. Desain Halaman Login Admin

2. Dashboard Admin

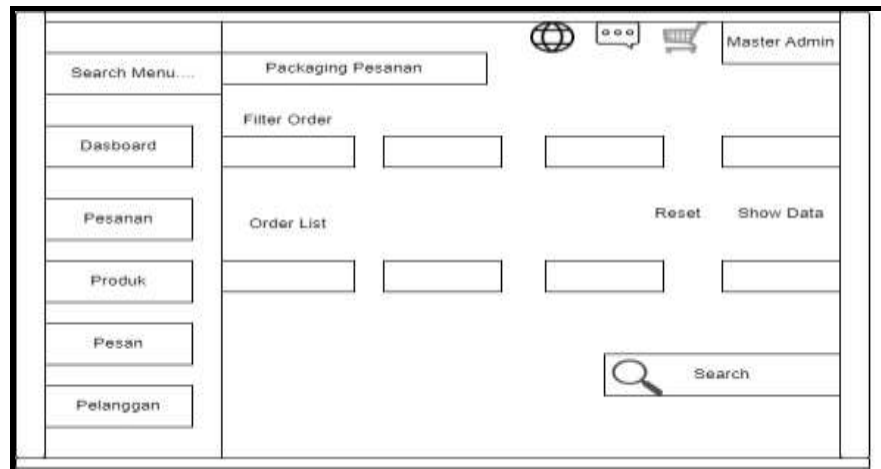
Berikut desain *Interface* admin untuk halaman pertama kali admin setelah melakukan proses login ke dalam sistem atau aplikasi.



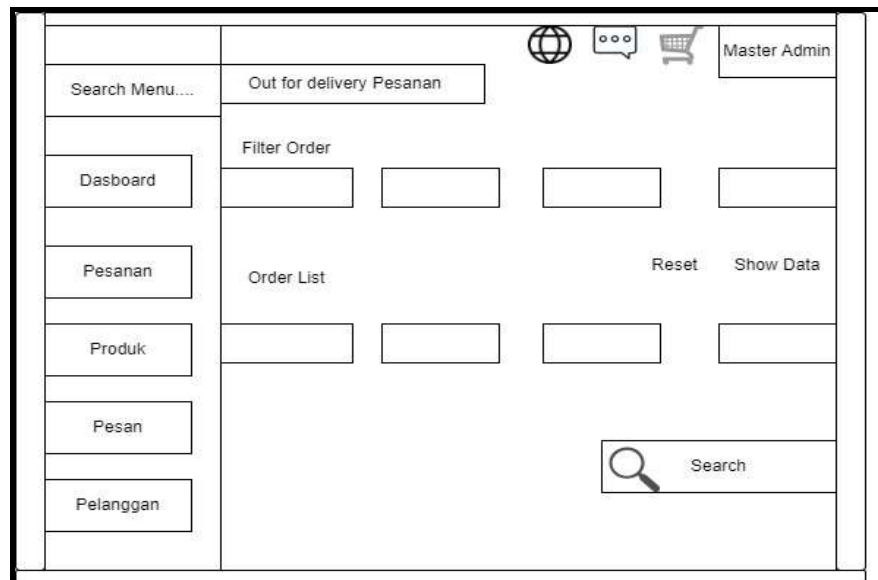
Gambar 5. 38. Desain Halaman Dashboard Admin

3. Pesanan

Berikut desain *Interface* bagian pesanan yang terdapat kategori Packaging dan pesanan dikirim dapat dilihat pada gambar dibawah ini.



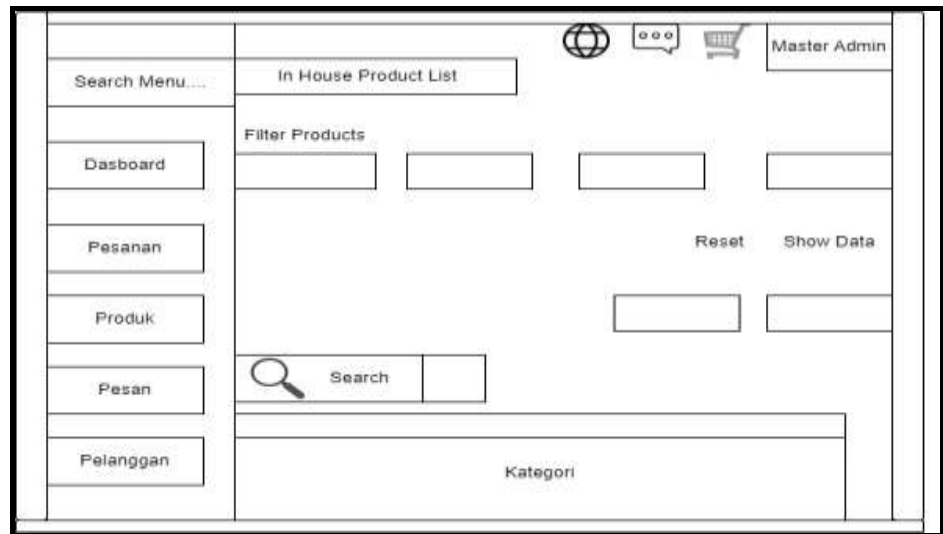
Gambar 5. 39. Desain Halaman Packaging



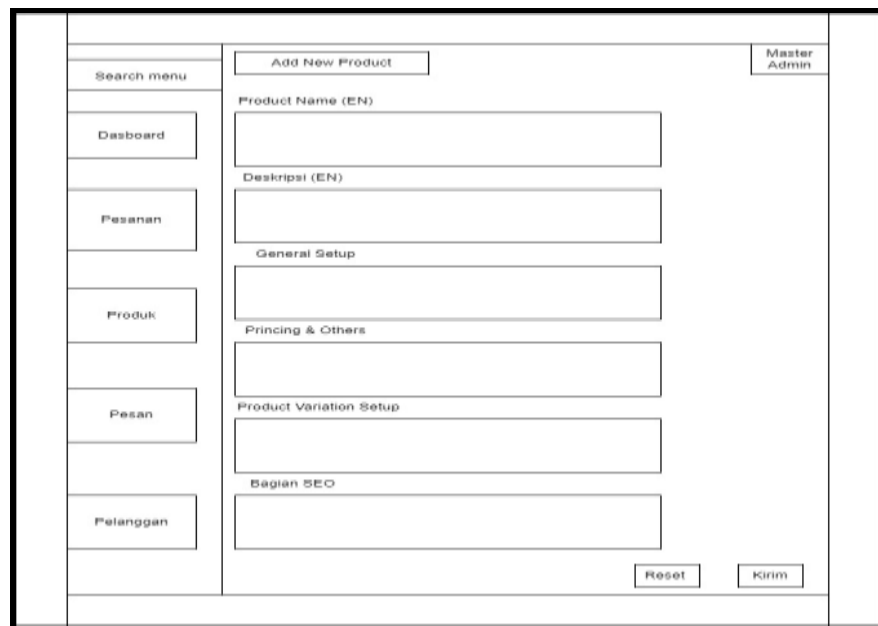
Gambar 5. 40. Desain Halaman Pesanan Dikirim

4. Produk Admin

Berikut desain *Interface* bagian produk yang terdapat kategori tambah produk dan tambah data produk. Dapat dilihat pada gambar dibawah ini.



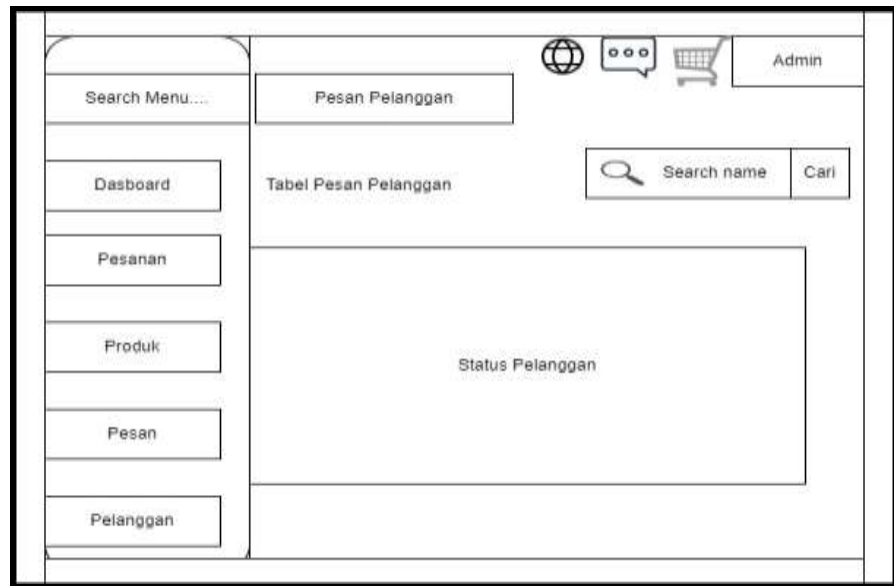
Gambar 5. 41. Desain Interface Data Produk



Gambar 5. 42. Desain Interface Tambah Produk

5. Pesan Pelanggan

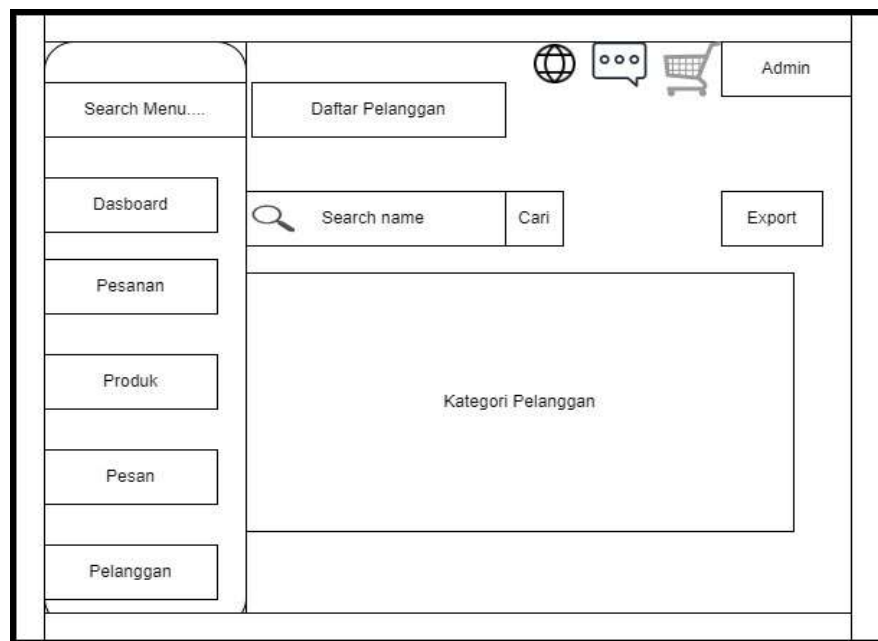
Berikut desain *Interface* bagian Pesan Pelanggan yang terdapat kategori pesan dari pelanggan. Dapat dilihat pada gambar dibawah ini.



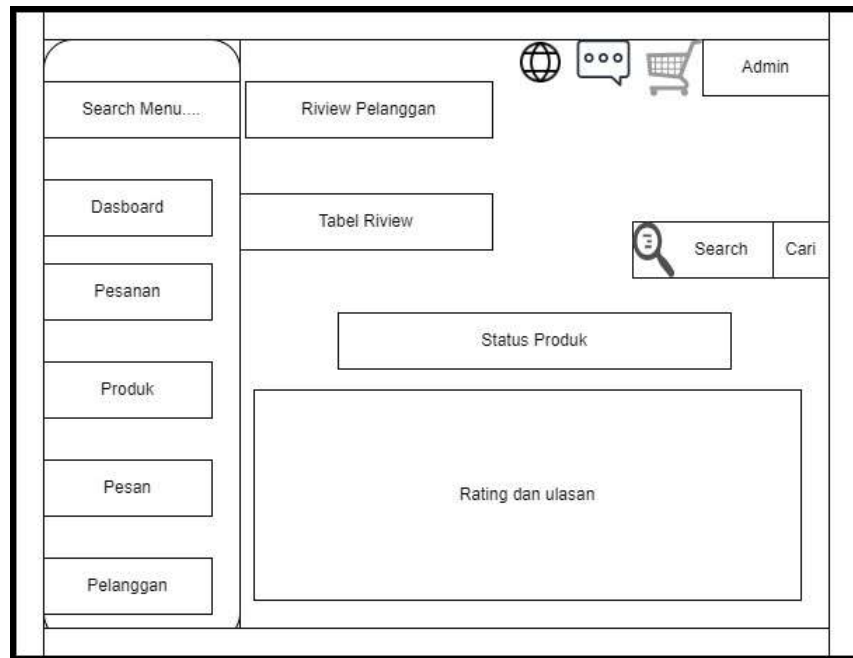
Gambar 5. 43. . Desain Interface Pesan

6. Pelanggan

Berikut desain *Interface* bagian pelanggan yang terdapat kategori daftar pelanggan dan review pelanggan. Dapat dilihat pada gambar dibawah ini.



Gambar 5. 44. Desain Interface Daftar Pelanggan



Gambar 5. 45. Desain Interface Review Pelanggan

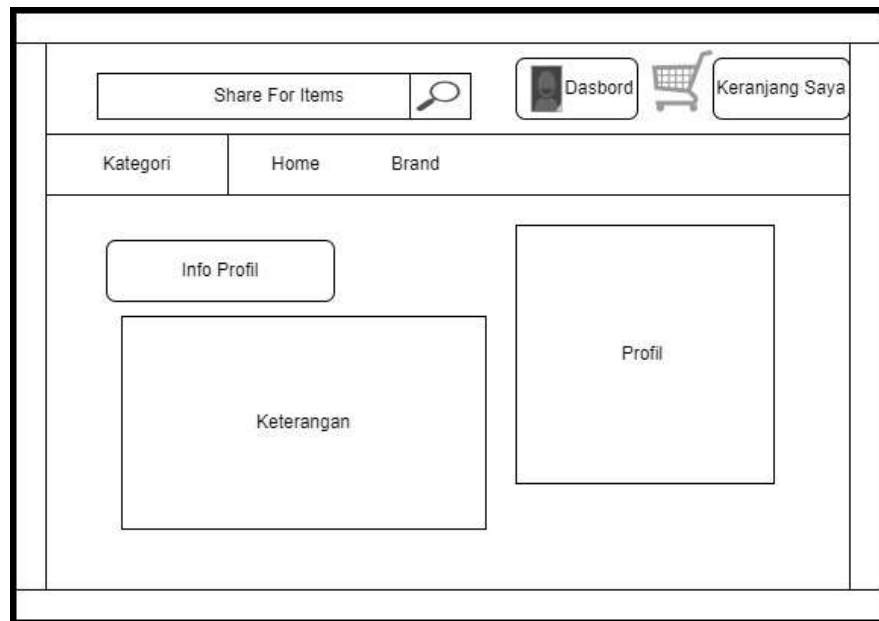
7. Login/ Register Pelanggan

Berikut desain *Interface* bagian login pelanggan yang menampilkan register terlebih dahulu sebelum masuk melalui akun. Dapat dilihat pada gambar dibawah ini.

Gambar 5. 46. Desain Interface Register Pelanggan

8. Profil Pelanggan

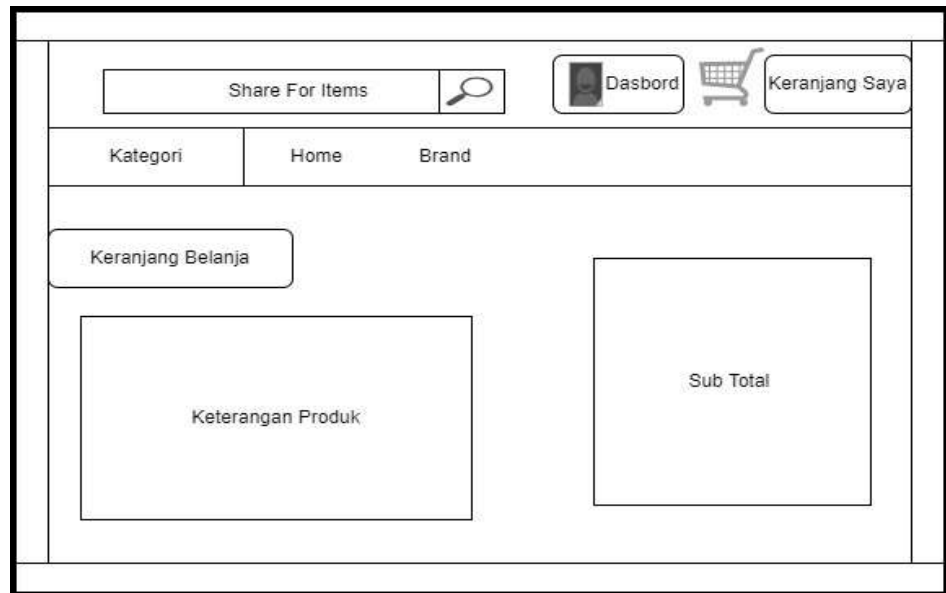
Berikut desain *Interface* bagian profil pelanggan yang terdapat keterangan profil pelanggan seperti foto dan biodata pelanggan. Dapat dilihat pada gambar dibawah ini.



Gambar 5. 47. Desain Interface Profil Pelanggan

9. Keranjang Belanja Pelanggan

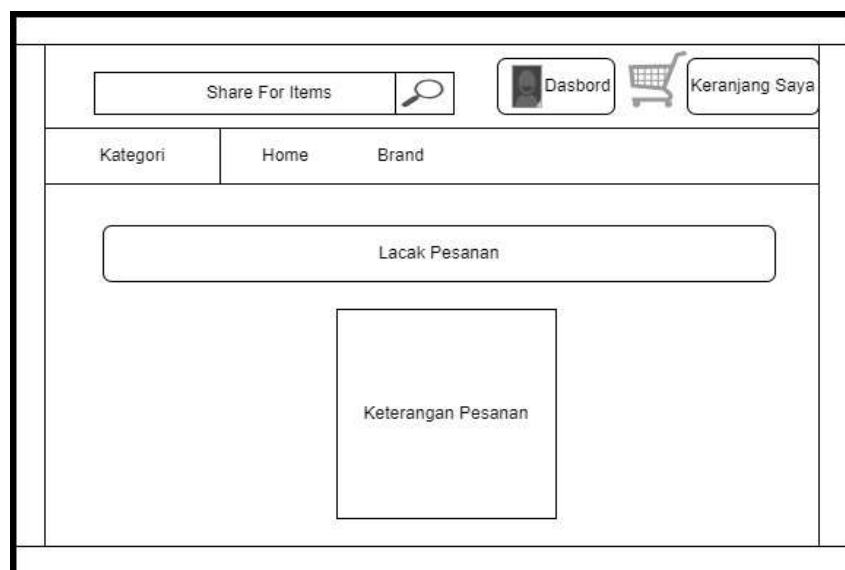
Berikut desain *Interface* bagian keranjang belanja pelanggan dengan melihat informasi beberapa kategori seperti pelanggan dapat *checkout* dan melihat keterangan produk dan subtotal produk yang akan dibeli. Dapat dilihat pada gambar dibawah ini.



Gambar 5. 48. Desain Interface Keranjang Pelanggan

10. Pesanan Dilacak

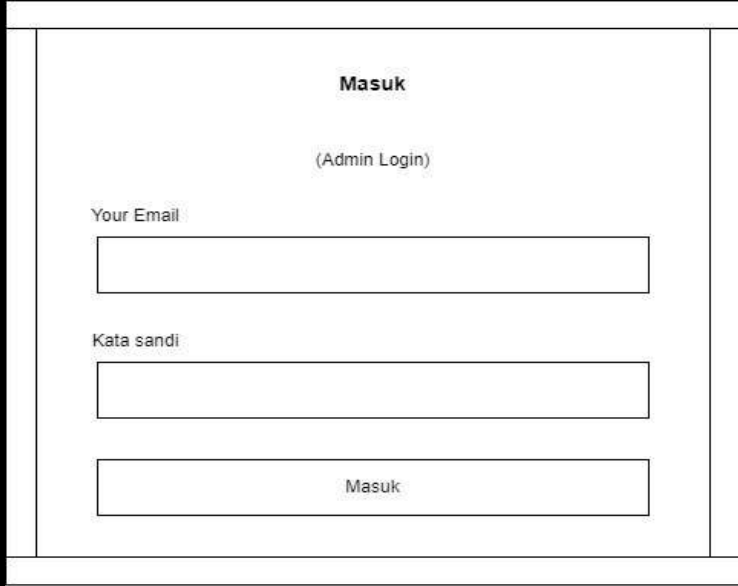
Berikut desain *Interface* bagian pesanan dilacak yang dimana pelanggan dapat melihat proses pengiriman atau melacak pesanan Dapat dilihat pada gambar dibawah ini.



Gambar 5. 49. Desain Interface Pesanan Lacak Pesanan

11.Login Owner

Berikut desain *Interface* owner yang digunakan untuk masuk ke halaman login. Desain Form login dapat dilihat pada desain login dapat dilihat pada gambar dibawah ini.

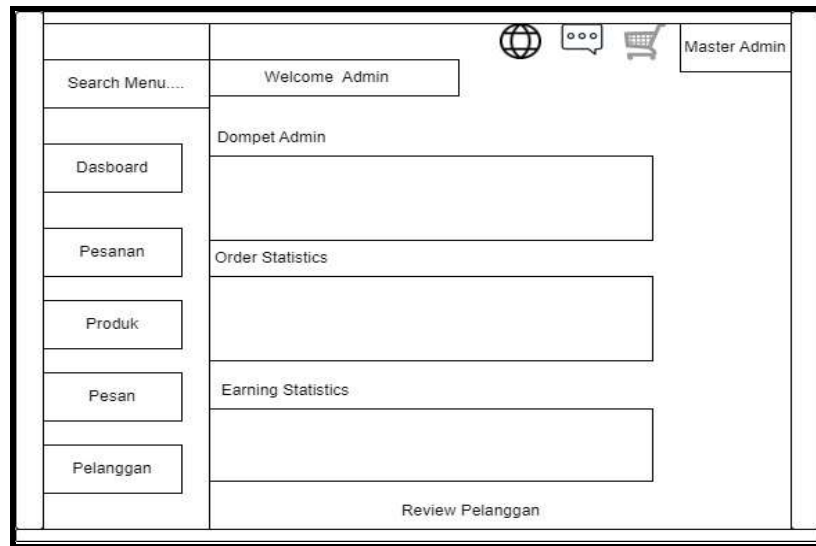


The image shows a login form titled "Masuk" (Login) with the subtitle "(Admin Login)". The form contains three input fields: "Your Email", "Kata sandi" (Password), and a "Masuk" button. The form is enclosed in a rectangular border.

Gambar 5. 50. Desain Interface Login Owner

12.Dashboard Owner

Berikut desain *Interface* Owner untuk halaman pertama kali Owner setelah melakukan proses login ke dalam sistem atau aplikasi. Desain dashboard dapat dilihat pada gambar dibawah ini.



Gambar 5. 51. Desain Interface Dashboard Owner

5.1.3 Construction

Tahapan ini merupakan tahapan implementasi dari tahap sebelumnya, pada tahapan ini penulis melakukan proses pengkodean adapun hasil pengkodean ini dapat dilihat dengan tampilan *interface*.

5.1.3.1 Tampilan Interface

1. Tampilan Login Admin

Halaman Login admin menampilkan halaman untuk login sebagai admin tampilan login admin dapat dilihat pada gambar dibawah ini.

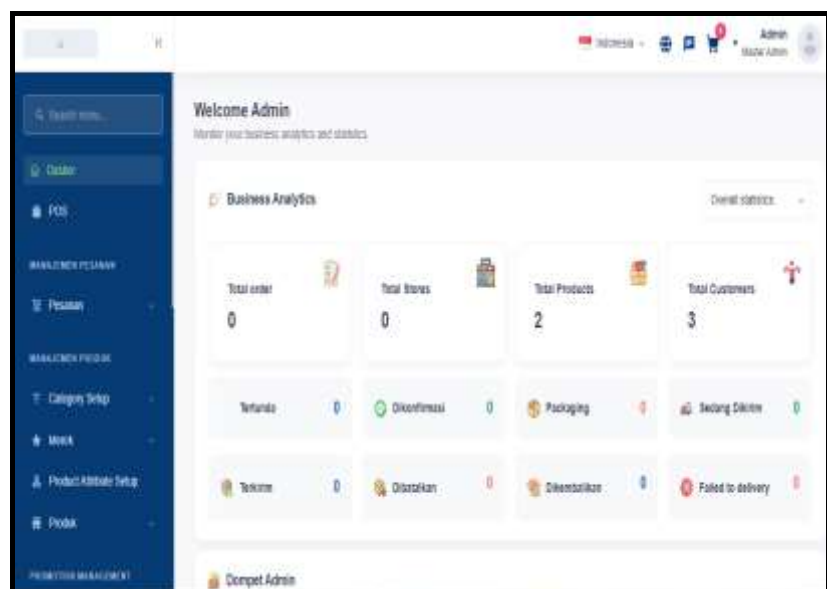
The image shows a login form titled "Masuk (Admin Login)". It contains the following elements:

- Your Email:** A text input field containing "email@address.com".
- Kata Sandi:** A password input field with a strength indicator "8+ characters required" and a toggle for visibility.
- Ingat Saya:** A checkbox labeled "Ingat Saya".
- Enter captcha value:** A text input field next to a captcha image showing the characters "JA QD".
- Masuk:** A large blue button at the bottom.

Gambar 5. 52. Tampilan Interface Halaman Login

2. Tampilan Dashboard Admin

Halaman dashboard admin menampilkan informasi terkait penjualan tampilan dashboard admin dapat dilihat pada gambar dibawah ini.

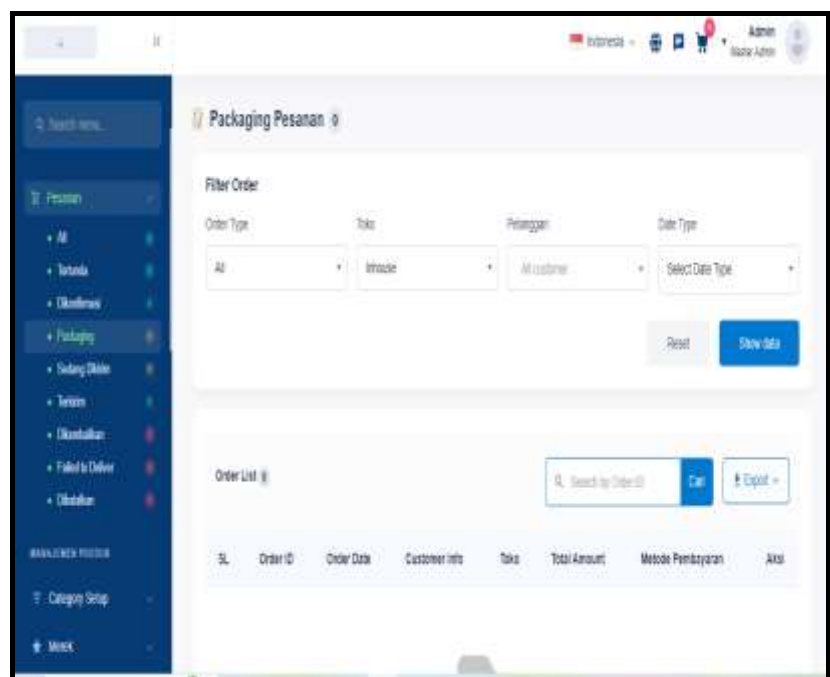


Gambar 5. 53. Tampilan Interface Halaman Dasbord

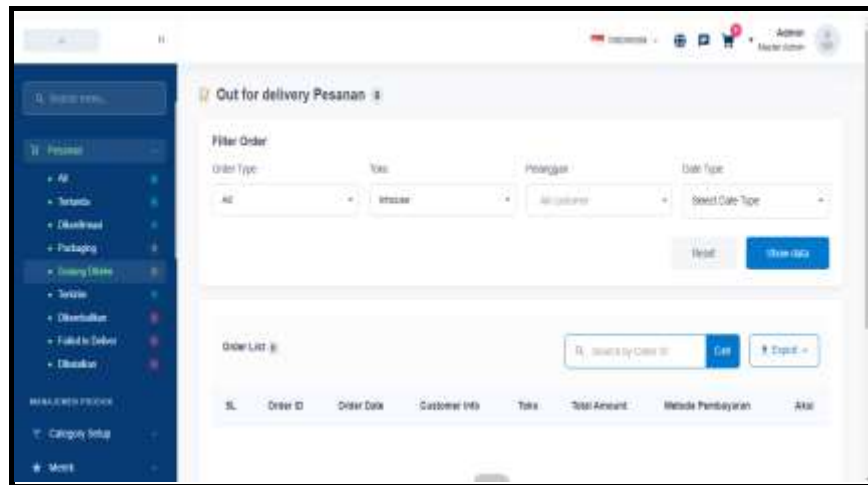
3. Tampilan Pesanan

Halaman pesanan admin menampilkan informasi terkait status pesanan yaitu packaging dan pesanan dikirim.

Dapat dilihat pada gambar dibawah ini.



Gambar 5. 54. Tampilan Interface Halaman Packaging

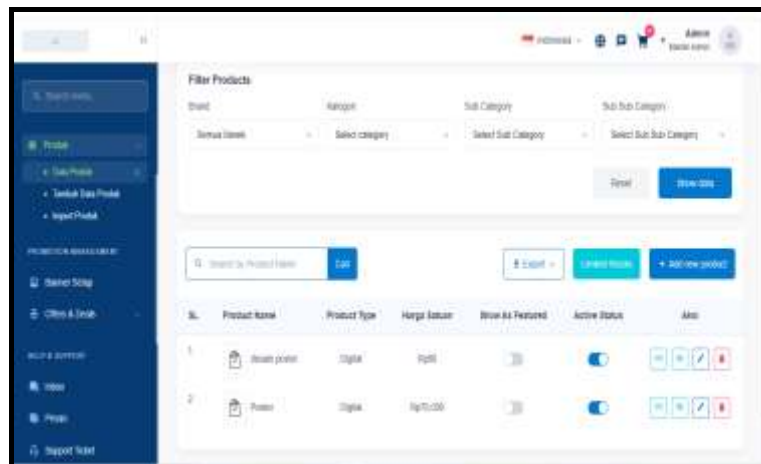


Gambar 5. 55. Tampilan Interface Halaman Pesanan Dikirim

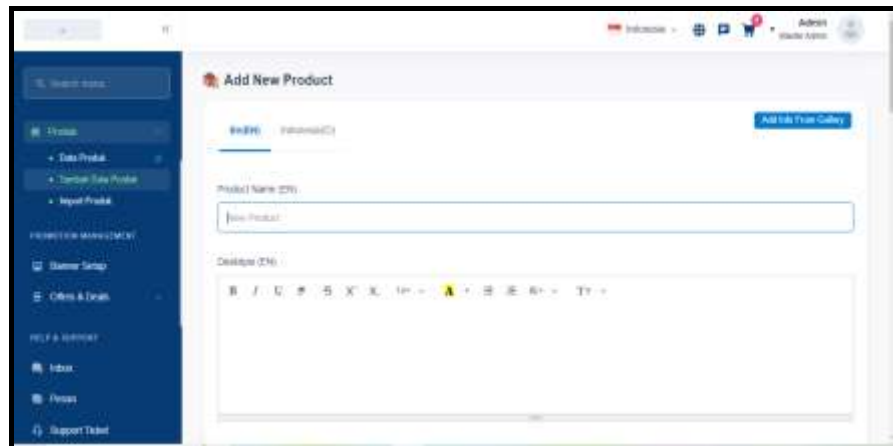
4. Tampilan Produk

Halaman tampilan produk menampilkan informasi terkait kategori produk berupa data produk dan tambah produk.

Dapat dilihat pada gambar dibawah ini.



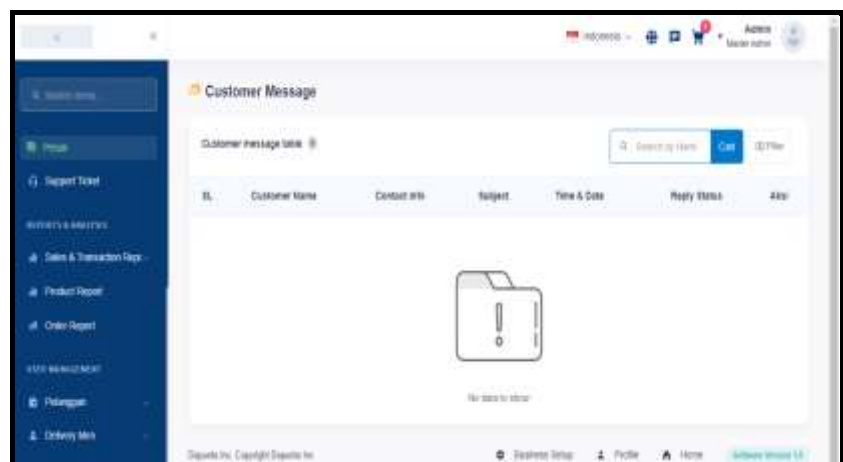
Gambar 5. 56. Tampilan Interface Data Produk



Gambar 5. 57. Tampilan Interface Tambah Produk

5. Pesan

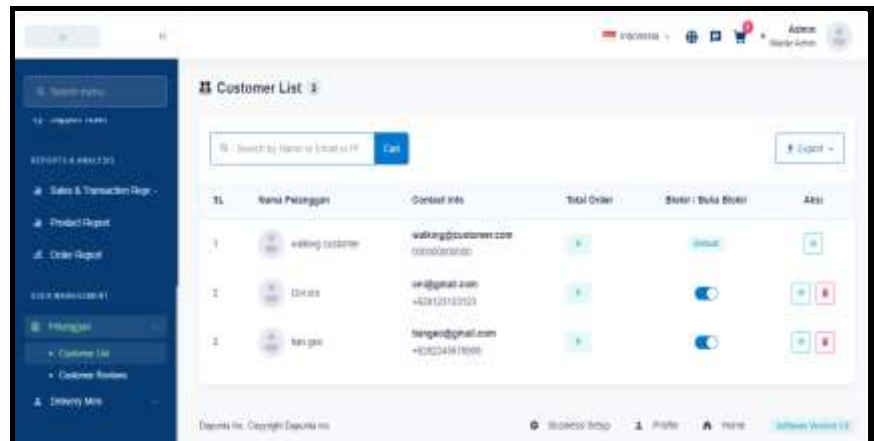
Halaman pesan admin menampilkan status pesan dari pelanggan .Dapat dilihat pada gambar dibawah ini.



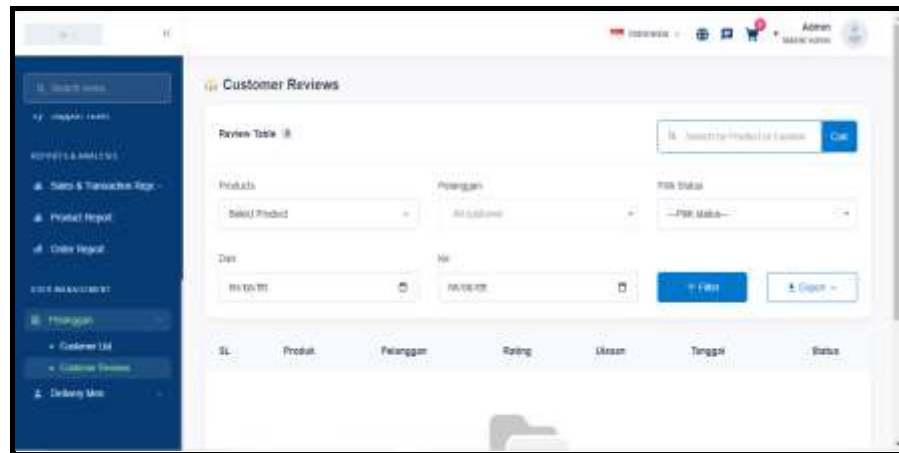
Gambar 5. 58. Tampilan Interface Pesan

6. Pelanggan

Halaman tampilan pelanggan menampilkan daftar pelanggan dan review pelanggan. Dapat dilihat pada gambar dibawah ini.



Gambar 5. 59. Tampilan Interface Daftar Pelanggan



Gambar 5. 60. Tampilan Interface Review Pelanggan

7. Login Pelanggan

Halaman Login pelanggan menampilkan halaman untuk login sebagai tampilan login pelanggan dapat dilihat pada gambar dibawah ini.

Masuk

Email / Phone

Kata Sandi

Ingat saya [Lupa kata sandi?](#)

Enter captcha value

Enjoy New experience [Sign up](#)

Gambar 5. 61. Tampilan Interface Login Pelanggan

8. Profil Pelanggan

Halaman tampilan Profil pelanggan yang terdapat keterangan profil pelanggan seperti foto dan biodata pelanggan. Dapat dilihat pada gambar dibawah ini.

Search for items...

Kategori Home Brand Discounted Products

Info Profil

Info profil

Info Profil

Nama Depan:

Nama Belakang:

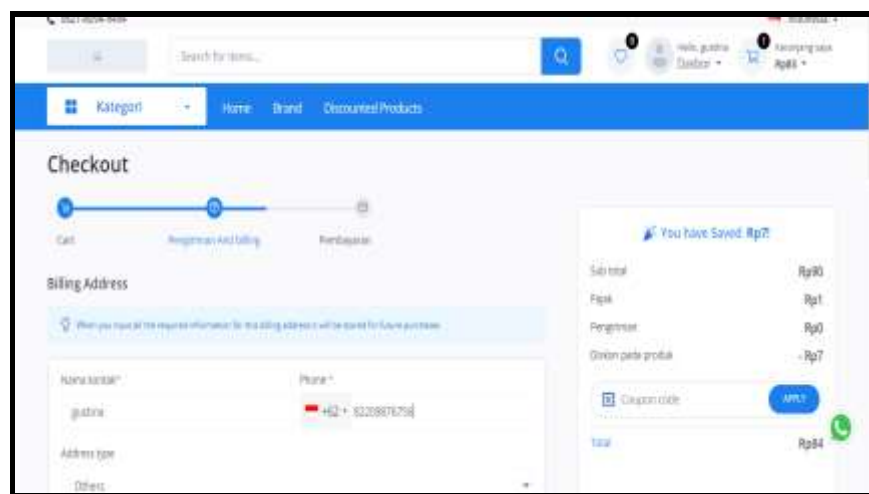
Nomor Telepon:

Email:

Gambar 5. 62. Tampilan Interface Profil Pelanggan

9. Keranjang Pelanggan

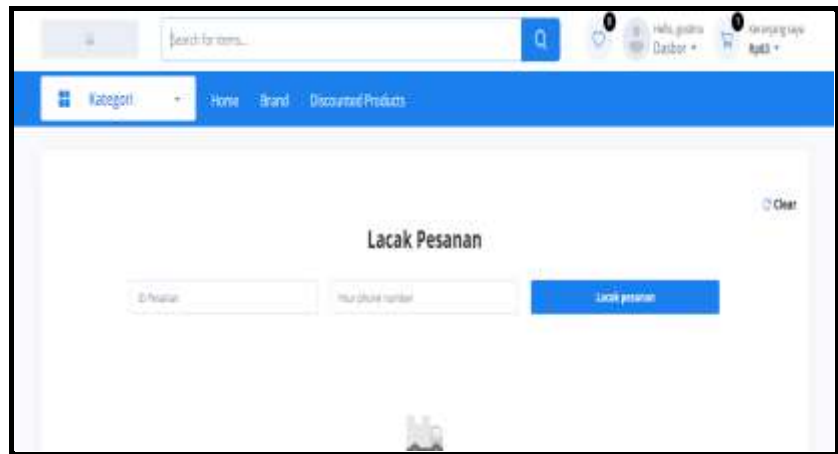
Berikut desain tampilan bagian keranjang belanja pelanggan dengan melihat informasi beberapa kategori seperti pelanggan dapat *checkout* dan melihat keterangan produk dan subtotal produk yang akan dibeli. Dapat dilihat pada gambar dibawah ini.



Gambar 5. 63. Tampilan Interface keranjang Pelanggan

10. Pesanan dilacak

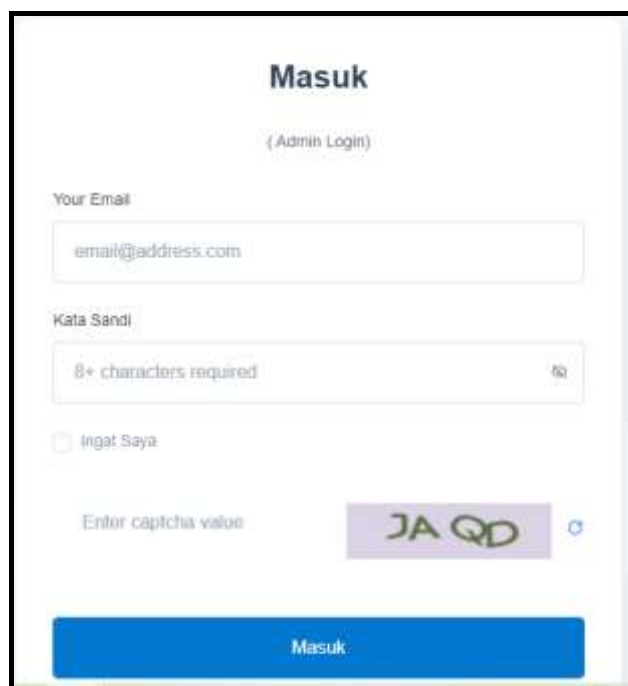
Berikut desain tampilan bagian pesanan dilacak yang dimana pelanggan dapat melihat proses pengiriman atau melacak pesanan. Dapat dilihat pada gambar dibawah ini.



Gambar 5. 64. Tampilan Interface Lacak Pesanan

11. Login Owner

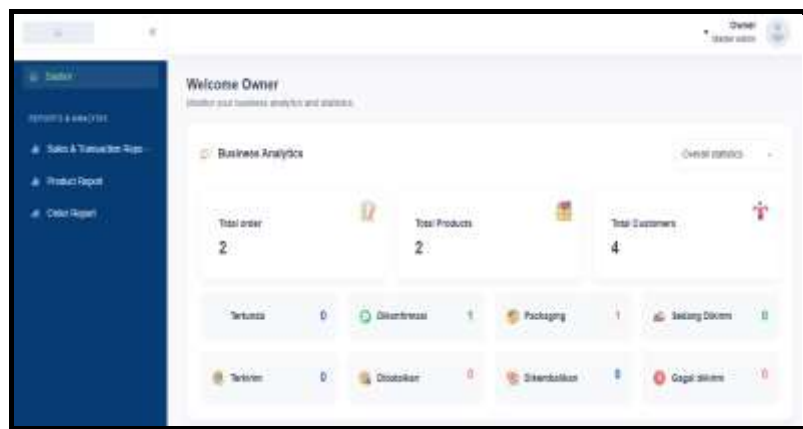
Berikut desain tampilan bagian yang digunakan untuk masuk ke halaman login. Tampilan Form login dapat dilihat pada gambar dibawah ini.



Gambar 5. 65. Tampilan Interface Login Owner

12. Dashboard Owner

Berikut desain tampilan bagian Owner untuk halaman pertama kali Owner setelah melakukan proses login ke dalam sistem atau aplikasi. Desain dashboard dapat dilihat pada gambar dibawah ini.



Gambar 5. 66. . Tampilan Dashboard Owner

5.1.4 Pengujian White Box Testing

Pengujian yang dilakukan oleh penulis pada *e-catalog* Dapunta Inc ini adalah pengujian dengan metode *white box testing* dimana menggunakan teknik *basis path testing*.

Sebelum pengujian dilakukan, berikut adalah *source code* yang akan diuji yang dapat dilihat pada tabel 5.15.

Tabel 5.15. *Source code* yang diuji

```
$user = User::create([ 'name' => $request['f_name'] . ' ' . $request['l_name'],
'f_name' => $request['f_name'], 'l_name' => $request['l_name'], 'email' =>
```

```
$request['email'], 'phone' => $request['phone'], 'is_active' => 1, 'password' =>
bcrypt($request['password']), 'referral_code' =>
Helpers::generate_referer_code(), 'referred_by' => (isset($referUser) &&
$referUser) ? $referUser->id : null, ]);
```

Node 1: Memulai fungsi submit.

Node 2: Memeriksa apakah ada *referral_code* dalam request.

Node 3: Jika ada *referral_code*, mencari pengguna yang merujuk berdasarkan kode tersebut.

Node 4: Membuat pengguna baru dengan informasi dari request.

Node 5: Menyimpan pengguna baru ke dalam database.

Node 6: Mengambil status verifikasi telepon dari konfigurasi web.

Node 7: Mengambil status verifikasi email dari konfigurasi web.

Jumlah Node (N) = 7 Jumlah Edge (E) = 8

Cyclomatic Complexity (V(G)) = E - N + 2 = 8 - 7 + 2 = 3

Dengan 7 node, berikut adalah basis path yang harus diuji:

Path 1: 1 → 2 → 3 → 4 → 5 → 6

Path 2: 1 → 2 → 3 → 4 → 5 → 7

Path 3: 1 → 2 → 4 → 5 → 6

Path 4: 1 → 2 → 4 → 5 → 7

Adapun hasil dari pengujian *white box testing* dapat dilihat pada tabel 5.15.

Tabel 5.15. Pengujian *White Box Testing*

No	Kasus Uji	Alur yang diuji	Hasil yang diharapkan
1	Pertama	1 → 2 → 3 → 4 → 5 → 6	Mengarahkan ke URL verifikasi telepon dengan kode referal valid
2	Kedua	1 → 2 → 3 → 4 → 5 → 7	Mengarahkan ke URL verifikasi email tanpa kode referal
3	Ketiga	1 → 2 → 4 → 5 → 6	Mengarahkan ke URL verifikasi telepon tanpa kode referal
4	Keempat	1 → 2 → 4 → 5 → 7	Mengarahkan ke URL verifikasi email tanpa kode referal

Kasus Uji 1: Valid karena pengguna baru berhasil terdaftar dengan kode referal yang valid dan proses verifikasi telepon dimulai.

Kasus Uji 2: Valid karena pengguna baru berhasil terdaftar tanpa menggunakan kode referal dan proses verifikasi email dimulai.

Kasus Uji 3: Tidak valid karena proses verifikasi telepon dimulai tanpa menggunakan kode referal.

Kasus Uji 4: Tidak valid karena proses verifikasi email dimulai tanpa menggunakan kode referal.

Tabel 5.16. Pengujian *White Box Testing*

No	Kasus Uji	Alur yang diuji	Hasil yang diharapkan
5	Kode referral tidak valid	1 → 2 → 3 (kode tidak ditemukan) → 4 → 5 → 6	Mengarahkan ke URL verifikasi telepon tanpa kode referal, atau menampilkan pesan error
6	Email tidak valid	1 → 2 → 3 → 4 (email tidak valid) → 5 (error)	Menampilkan pesan error bahwa email tidak valid
7	Password terlalu pendek	1 → 2 → 3 → 4 (password tidak memenuhi syarat) → 5 (error)	Menampilkan pesan error bahwa password terlalu pendek
8	Error saat menyimpan database	1 → 2 → 3 → 4 → 5 (error)	Menampilkan pesan error umum atau spesifik (jika memungkinkan)
9	Konfigurasi verifikasi	1 → 2 → 3 → 4 → 5 → 6 (konfigurasi verifikasi email tidak aktif)	Mengarahkan langsung ke dashboard atau halaman lain yang sesuai

Kasus Uji 5 : Kode referal tidak valid, untuk Memastikan sistem dapat menangani situasi di mana kode referral yang dimasukkan tidak ditemukan dalam database.

Kasus Uji 6 : Email tidak valid, untuk Memastikan sistem dapat memvalidasi format email pengguna.

Kasus Uji 7 : Password terlalu pendek, untuk Memastikan sistem menerapkan kebijakan keamanan password.

Kasus Uji 8 : Error saat menyimpan ke database Memastikan sistem dapat menangani kesalahan yang mungkin terjadi saat menyimpan data pengguna ke database.

Kasus Uji 9 : Konfigurasi verifikasi email diubah menjadi tidak aktif, Memastikan sistem dapat beradaptasi dengan perubahan konfigurasi.

Tabel 5.17. Pengujian *White Box Testing*

No	Kasus Uji	Alur yang diuji	Hasil yang diharapkan
1	<i>Login</i> dengan kredensial yang valid	<i>Username</i> dan <i>password</i> benar	Pengguna berhasil <i>login</i>
2	<i>Login</i> dengan <i>username</i> yang salah	<i>Username</i> salah, <i>password</i> benar	Gagal <i>login</i> , tampilkan pesan <i>error</i>
3	<i>Login</i> dengan <i>password</i> yang salah	<i>Username</i> benar, <i>password</i> salah	Gagal <i>login</i> , tampilkan pesan <i>error</i>
4	<i>Login</i> dengan <i>username</i> dan <i>password</i> kosong	Kedua <i>field</i> kosong	Gagal <i>login</i> , tampilkan pesan <i>error</i> semua <i>field</i> harus diisi
5	<i>Login</i> dengan karakter khusus pada <i>username/password</i>	<i>Username</i> atau <i>password</i> mengandung karakter khusus	Diterima jika sesuai kebijakan, ditolak jika tidak valid
6	<i>Login</i> dengan kombinasi huruf besar kecil yang berbeda	<i>Username</i> atau <i>password</i> dengan <i>case sensitive</i>	Ditolak jika tidak cocok
7	<i>Login</i> berulang kali dengan kredensial yang salah	Beberapa percobaan <i>login</i> gagal berturut-turut	Akun terkunci sementara atau <i>captcha</i> muncul
8	<i>Login</i> menggunakan akun yang tidak aktif	Akun pengguna dinonaktifkan	Gagal <i>login</i> , tampilkan pesan akun tidak aktif
9	<i>Login</i> menggunakan akun yang dihapus	Akun pengguna dihapus	Gagal <i>login</i> , tampilkan pesan akun tidak

			ditemukan
10	Mencoba melakukan brute force attack	Melakukan banyak percobaan login dengan kombinasi username dan password secara acak	Sistem mendeteksi dan memblokir IP atau akun

Kasus Uji 1 : Valid karena menguji apakah pengguna dapat masuk dengan username dan password yang benar.

Kasus Uji 2 : Login dengan kombinasi huruf besar kecil yang berbeda menguji sensitivitas case pada username dan password.

Kasus Uji 3 : Menguji apakah sistem memblokir akses setelah beberapa kali percobaan gagal.

Kasus Uji 4 : Secara eksplisit memeriksa apakah username atau password kosong menampilkan pesan error, mengalihkan ke halaman lain.

Kasus Uji 5 : Memastikan sistem tidak mengalami crash atau memberikan output yang tidak terduga ketika pengguna tidak mengisi username atau password.

Kasus Uji 6 : Memastikan uji dengan berbagai kombinasi huruf besar kecil untuk memastikan sistem konsisten.

Kasus Uji 7 : Uji dengan berbagai kombinasi username dan password yang salah, termasuk karakter khusus dan panjang yang berbeda.

Kasus Uji 8 : Mencegah akses yang tidak sah sistem harus menolak akses bagi pengguna yang akunnya tidak aktif.

Kasus Uji 9 : Mencegah akses tidak sah, sistem harus sepenuhnya memblokir akses ke akun yang telah dihapus.

Kasus Uji 10 : Memastikan sistem memiliki mekanisme yang efektif untuk mencegah serangan *brute force*.

Tabel 5.18. *Source code* yang diuji

```
public function add(AttributeRequest $request):
    JsonResponse|RedirectResponse { $dataArray = [ 'name' =>
    $request['name'][array_search('en', $request['lang'])], ]; $savedAttributes =
    $this->attributeRepo->add(data:$dataArray); $this->translationRepo-
    >add(request:$request, model:'App\Models\Attribute', id:$savedAttributes->id);
    Toastr::success(translate('attribute_added_successfully')); return back(); }
```

Node 1: Memulai fungsi add.

Node 2: Membuat array dataArray.

Node 3: Menyimpan atribut menggunakan attributeRepo.

Node 4: Menambah terjemahan menggunakan translationRepo.

Node 5: Menampilkan pesan sukses menggunakan Toastr.

Node 6: Mengembalikan response menggunakan return back().

Jumlah Node (N) = 6 Jumlah Edge (E) = 6

Kompleksitas siklomatik (V(G)) = E - N + 2 = 6 - 6 + 2 = 2

Dengan 6 node dan kompleksitas siklomatik 2, berikut adalah basis path yang harus diuji:

Path 1: 1 → 2 → 3 → 4 → 5 → 6

Path 2: $1 \rightarrow 2 \rightarrow 3 \rightarrow 4 \rightarrow 5 \rightarrow 6$ (karena semua edge harus dilalui, path-nya sama)

Adapun hasil dari pengujian *white box testing* dapat dilihat pada tabel 5.19.

Tabel 5.19. Pengujian *White Box Testing*

No	Kasus Uji	Alur yang diuji	Hasil yang diharapkan
1	Pertama	$1 \rightarrow 2 \rightarrow 3 \rightarrow 4 \rightarrow 5 \rightarrow 6$	Atribut berhasil ditambahkan dan terjemahan disimpan, menampilkan pesan sukses, kembali ke halaman sebelumnya
2	Kedua	$1 \rightarrow 2 \rightarrow 3 \rightarrow 4 \rightarrow 5 \rightarrow 6$	Menampilkan pesan error bahwa nama atribut tidak boleh kosong, tidak menyimpan data
3	Ketiga	$1 \rightarrow 2 \rightarrow 3$ (error)	Menampilkan pesan error umum atau spesifik terkait kesalahan penyimpanan data
4	Keempat	$1 \rightarrow 2 \rightarrow 3 \rightarrow 4$ (error)	Menampilkan pesan error umum atau spesifik terkait kesalahan penambahan terjemahan

Kasus Uji 1: Nama atribut bahasa Inggris valid Alur yang diuji:

$1 \rightarrow 2 \rightarrow 3 \rightarrow 4 \rightarrow 5 \rightarrow 6$

Hasil yang diharapkan: Atribut berhasil ditambahkan dan terjemahan disimpan, menampilkan pesan sukses, kembali ke halaman sebelumnya.

Kasus Uji 2: Nama atribut bahasa Inggris kosong Alur yang diuji: 1 → 2 → 3 → 4 → 5 → 6

Hasil yang diharapkan: Menampilkan pesan error bahwa nama atribut tidak boleh kosong, tidak menyimpan data.

Kasus Uji 3: Error saat menyimpan atribut Alur yang diuji: 1 → 2 → 3 (error)

Hasil yang diharapkan: Menampilkan pesan error umum atau spesifik terkait kesalahan penyimpanan data.

Kasus Uji 4: Error saat menambah terjemahan Alur yang diuji: 1 → 2 → 3 → 4 (error)

Hasil yang diharapkan: Menampilkan pesan error umum atau spesifik terkait kesalahan penambahan terjemahan.

Tabel 5.20. *Source code* yang diuji

```
public function delete(Request $request): JsonResponse { $this->attributeRepo->delete(params:['id'=>$request['id']]); $this->translationRepo->delete(model:'App\Models\Attribute', id:$request['id']); return response()->json(['message'=> translate('attribute_deleted_successfully')]); }
```

Node 1: Memulai fungsi delete.

Node 2: Menghapus atribut menggunakan attributeRepo.

Node 3: Menghapus terjemahan menggunakan translationRepo.

Node 4: Mengembalikan respons JSON.

Jumlah Node (N) = 4 Jumlah Edge (E) = 4

Kompleksitas siklomatik (V(G)) = E - N + 2 = 4 - 4 + 2 = 2

Dengan 4 node dan kompleksitas siklomatik 2, berikut adalah basis path yang harus diuji:

Path 1: 1 → 2 → 3 → 4

Path 2: 1 → 2 → 3 → 4 (karena semua edge harus dilalui, path-nya sama)

Adapun hasil dari pengujian *white box testing* dapat dilihat pada tabel 5.21.

Tabel 5.21. Pengujian *White Box Testing*

No	Kasus Uji	Alur yang diuji	Hasil yang diharapkan
1	Pertama	1 → 2 → 3 → 4	Atribut dan terjemahan berhasil dihapus, menampilkan pesan sukses dalam respons JSON
2	Kedua	1 → 2 (error)	Menampilkan pesan error bahwa atribut tidak ditemukan, tidak melanjutkan ke langkah berikutnya
3	Ketiga	1 → 2 → 3 (error)	Atribut berhasil dihapus tetapi menampilkan pesan error terkait penghapusan terjemahan
4	Keempat	1 (error)	Menampilkan pesan error bahwa ID tidak boleh kosong, tidak melanjutkan proses penghapusan

Kasus Uji 1: Penghapusan atribut berhasil Alur yang diuji: 1 → 2 → 3 → 4

Hasil yang diharapkan: Atribut dan terjemahan berhasil dihapus, menampilkan pesan sukses dalam respons JSON.

Kasus Uji 2: Atribut tidak ditemukan Alur yang diuji: 1 → 2
(error)

Hasil yang diharapkan: Menampilkan pesan error bahwa atribut tidak ditemukan, tidak melanjutkan ke langkah berikutnya.

Kasus Uji 3: Error saat menghapus terjemahan Alur yang diuji:
1 → 2 → 3 (error)

Hasil yang diharapkan: Atribut berhasil dihapus tetapi menampilkan pesan error terkait penghapusan terjemahan.

Kasus Uji 4: Validasi request ID kosong Alur yang diuji: 1
(error)

Hasil yang diharapkan: Menampilkan pesan error bahwa ID tidak boleh kosong, tidak melanjutkan proses penghapusan.

5.1.5 Transition

Tahapan ini biasanya untuk menyerahkan sistem aplikasi kepada user (*roll-out*), yang umumnya mencakup pelatihan dan *beta testing* aplikasi. Tahapan ini merupakan tahapan dimana penulis melakukan penyerahan aplikasi kepada pihak Dapunta.

5.2 Pembahasan

Setelah melakukan pembuatan *E-catalog* Dapunta Inc, dengan menggunakan tahapan pengembangan sistem RUP, telah dihasilkan

sebuah sistem yang dapat berjalan dengan baik, hal ini ditegaskan dengan pengujian yang dilakukan menggunakan metode *whitebox testing* dengan teknik *basis path testing* yang menyatakan validitas uji kelayakan. Adapun fitur-fitur sistem telah dibuat sesuai dengan kebutuhan pengguna, mulai dari fitur registrasi, *login*, kategori, merk, produk, order, hingga pembayaran.

BAB VI

PENUTUP

6.1 Kesimpulan

Berdasarkan hasil penelitian dan pembahasan telah dihasilkan sebuah aplikasi *E-Catalog* Dapunta Inc menggunakan metode *Rational Unified Process (RUP)*. Sebagai solusi dari permasalahan yang ada, yakni Dapunta melakukan penjualan jasa desain grafis hanya dengan menggunakan instagram, dan belum menggunakan sebuah website jenis *E-catalog*. Aplikasi *E-Catalog* berbasis web ini dibuat dengan tujuan dapat memudahkan pendataan transaksi, pelanggan, pendataan omset serta membantu perekapan agar terekap dengan baik dan tidak terpisah diluar instagram.

6.2 Saran

Berdasarkan kesimpulan diatas, aplikasi yang dibuat tentunya masih terdapat kekurangan yang dimana penulis memberikan saran agar penelitian ini dapat dikembangkan lebih baik lagi, yakni seperti belum fitur ulasan untuk pelanggan memberikan penilaian berupa rating dan komentar terhadap produk yang dibeli serta menampilkan jumlah ulasan untuk produk yang terjual.

DAFTAR PUSTAKA

- A.S., Rosa dan Shalahuddin, M. 2022. *Rekayasa Perangkat Lunak Terstruktur dan Berorientasi Objek*. Bandung: Informatika Bandung
- Amelia, T., Erwin, T. H., & Widiyanti, R. 2022. Perancangan E-Catalogue Sebagai Media Promosi Produk Paket Wisata PT. Ranata Air Network. In *Prosiding Industrial Research Workshop and National Seminar* (Vol. 13, No. 01, pp. 1344-1351).
- Black, R. 2020, *Testing Fundamentals: A Practical Guide For Software Testers*. New York: CEC Press.
- Diani, F., & Lubis, F. A. 2020. Analisis Implementasi E-Katalog Terhadap Perkembangan UMKM di Kota Medan dalam mendukung Kemajuan Ekonomi Syariah. *JIEI: Jurnal Ilmiah Ekonomi Islam*, 8(2), 1970-1981.
- Fietri,A., & Ilham, R. 2021. *Monitoring dan Evaluasi Sebuah Pengantar*, Jakarta: PT Gramedia Pustaka Utama
- Firmansyah, R. 2021. *Aplikasi Reservasi Arung Jeram Dengan Metode Rational Unified Process (Studi Kasus Magelang Explore) (Doctoral dissertation, UNIVERSITAS MUHAMMADIYAH MAGELANG)*.
- Harmayani. 2020. *e-catalog: Suatu Pengantar Bisnis Digital*. Medan: Yayasan Klta Menulis
- Hutagalung, R., & Ramadan, Z. H. 2020. Peran orang tua dalam menanamkan nilai multikultural di keluarga siswa sekolah dasar. *Jurnal Obsesi: Jurnal Pendidikan Anak Usia Dini*, 6(5), 4967-4991.
- Iqbal, M. 2020. PENGARUH PELAKSANA E-KATALOG DALAM PENGADAAN BARANG/JASA PEMERINTAH TERHADAP UMKM. *Jurnal USM Law Review*, 3(1), 77.

- Maimunah, M., & Anshori, M. I. 2023. "Employee Well-Being In The Context Of Uncertainty And Change: Implications For HRD"(2017)-Advances In Developing Human Resources. *Jurnal Publikasi Ilmu Manajemen*, 2(4), 01-13.
- Monopo, O. C., Lengkong, V. P., & Uhing, Y. 2023. Praktek Manajemen Sumber Daya Manusia Pada Masa Pandemi Covid 19 Dalam Meningkatkan Kerja Karyawan Di PT Bank Rakyat Indonesia Kanca Tondano. *Jurnal EMBA: Jurnal Riset Ekonomi, Manajemen, Bisnis dan Akuntansi*, 11(02), 169-180.
- Marentek, T. C., Sumual, T. E., Usuh, E. J., & Kampilong, J. K. 2023. Perencanaan Sumber Daya Manusia Dalam Pendidikan Masa Depan. *El-Idare: Jurnal Manajemen Pendidikan Islam*, 9(1), 29-35.
- Rusmawan, U. 2019. *Analisi dan Perancangan Sistem Informasi*. Bandung: Informatika.
- S. Sulasno dan R. Saleh. 2020. *Desain dan Implementasi Sistem Monitoring Sumber Daya Server Menggunakan Zabbix 4.0*. JUITA J Inform, vol 8, no.2.
- Subagia, R.(2020). PENGUJIAN WHITE BOX PADA SISTEM INFORMASI MONITORING SKRIPSI PROGRAM STUDI INFORMATIKA UNIVERSITAS PEMBANGUNAN NASIONAL VETERAN JAWA TIMUR (Doctoral dissertation, UPN Veteran Jawa Timur).
- Sugiarti, L. 2020. *Dasar-dasar Pemograman Java Neatbeans: Basisdata UML dan Interface*. Bandung: Rosda
- Sugiarti, Y. 2020. *Rancang bangun sistem informasi posyandu guna mendukung kesehatan ibu dan memonitorig tumbuh kembang bayi berbasis web studi kasus: suku dinas kesehatan Jakarta*

Timur Provinsi DKI Jakarta (Bachelor's thesis, Fakultas Sains dan Teknologi UIN Syarif Hidayatullah Jakarta).

Sukamto, A., & Shalahuddin, M. 2020. *Rekayasa Perangkat Lunak Berorientasi Objek: Pendekatan UML dan Java* Jakarta: PT Gramedia Pustaka Utama.

Supardi, F., & Hardian, R. 2019. *Penerapan Metode Rational Unified Process Pada Perancangan sistem Pengolahan Data Arisankita*. *Jurnal Teknologi Informasi dan Konukasi*, 10(2), p-ISSN: 2087-1627, e-ISSN: 2686-9858.

LAMPIRAN

 Kode Formulir : FM-IPCT-BAAK-PSB-043	FORMULIR SURAT PERSETUJUAN TOPIK & JUDUL SKRIPSI Institusi : INSTITUT TEKNOLOGI DAN BISNIS PALCOMTECH
---	---

Kepada Yth.
Ka. Prodi System Informasi
di tempat.

Palembang, 21 Januari 2024

Dengan hormat,
Saya yang Bertanda tangan di bawah ini :

Program Studi : System Informasi

No	NPM	Nama	IPK	Semester	Sesi Belajar*	No.HP
1.	021200071	Gusma Dwi P	3.30	7	Pagi	08217696177
2.	021200044	Oni Lestari	3.52	7	Pagi	082216249580
3.						

* Pilih Salah Satu :Pagi/Siang/Malam

Mengajukan Skripsi dengan topik :

Aplikasi Berbasis Website

Dengan melampirkan deskripsi awal penelitian yang terdiri dari :

1. Objek Penelitian
2. Apa yang akan diteliti dari objek
3. Metode Pengembangan/analisis yang digunakan
4. Tujuan / hasil yang diharapkan dari penelitian

Rekomendasi Nama Pembimbing :

Menyetujui,
Wakil Rektor 1,

Mengetahui,
Ka. Prodi

Judul Skripsi (dalam bahasa Indonesia dan Inggris):

1. E-Commerce Dapunta Inc menggunakan metode RUP (E-Commerce Dapunta Inc uses the RUP Method)

2.

Diusulkan judul nomor :

Pemohon,
Mahasiswa 1,

Gusma Dwi Pratiwi

Menyetujui,
Pembimbing

Mahasiswa 2,

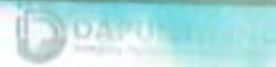
Oni Lestari

Mengetahui,
Ka. Prodi

Mahasiswa 3,

Mengesahkan
Wakil Rektor 1

- Diperbanyak 1 kali : Asli diserahkan ke BAAK dan copy diarsip Mahasiswa
- Form ini wajib dikembalikan ke BAAK pada saat pengumpulan berkas untuk pengajuan ujian komprehensif

**SURAT KETERANGAN**


Nomor: 024/DI/III/2024

Sehubungan dengan surat permohonan riset dari pihak Institut Teknologi dan Bisnis Palcomtech Palembang dengan Nomor : 0296/IPCT/R/II/2024 yang akan melaksanakan riset di perusahaan kami berjumlah 2 orang yaitu:

No	Nama	NPM	Semester	Prodi
1	Oni Lestari	021200044	VIII	Sistem Informasi
2	Gustina Dwi Pratiwi	021200071	VIII	Sistem Informasi

Maka, kami dari Pihak Perusahaan Dapunta Inc mengizinkan mahasiswa tersebut di atas untuk melakukan Riset dari tanggal 26 Februari 2024 di Dapunta Inc. Demikianlah surat keterangan kami buat untuk dipergunakan sebagaimana mestinya.

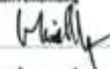


	FORMULIR			
	KONSULTASI LAPORAN SKRIPSI INSTITUT TEKNOLOGI DAN BISNIS PALCOMTECH			
Kode Formulir: FM-ICT-BAK-PSB-049	Institusi: Tahun Akademik	INSTITUT TEKNOLOGI DAN BISNIS PALCOMTECH		
NO	NPM	Nama	Prodi	Semester
1	011100071	Guthrie Febi Diahwa	di sistem informasi	Debaran (8)
2	011200049	Ami Lesian	di sistem informasi	Debaran (8)

Judul Laporan Skripsi: E-Commerce Dapunta Inc Mengukurakan Metode RUP

Pertemuan Ke -	Tanggal Konsultasi	Batas Waktu Perbaikan	Materi yang Dibahas / Catatan Perbaikan	Poin Perbaikan
1	14/5 2024	17/5 2024	Perbaiki latar belakang, rumusan masalah, tujuan dan ruang lingkup	✓
2	20/5 2024	22/5 2024	perbaiki latar belakang, ruang lingkup	✓
3	27/5 2024	26/5 2024	Accu ulang proposal	✓
4	07/5 2024	13/5 2024	Revisi hasil dan Pembatasan. Revisi Struktur organisasi	✓
5	13/5 2024	15/5 2024	Lanjutan Bab V Alur yang sedang berjalan di dapunta	✓
6	15/5 2024	17/5 2024	Perbaiki flowchart yang berjalan dan diuraikan	✓
7	17/5 2024	20/5 2024	Perbaiki Use Case dan Activity Diagram	✓
8	20/5 2024	22/5 2024	Revisi dan lanjut sequence diagram, case di diagram	✓
9	22/5 2024	27/5 2024	Revisi alur class diagram, pengecekan aplikasi	✓
10	29/5 2024	30/5 2024	Revisi aplikasi data browsing di sesuaikan	✓
11	05/6 2024	7/6 2024	Aplikasi	✓
12	10/6 2024	12/6 2024	Aplikasi	✓
13	12/6 2024	13/6 2024	Laporan	✓
14	14/6 2024	19/6 2024	Laporan	✓
15	19/6 2024		Acc laporan dan Aplikasi dan siap ujian.	✓

Palaembang,
Daerah Pembinaan


Mulya T.P.

SURAT PERNYATAAN UJIAN SKRIPSI

Yang bertanda tangan di bawah ini :

Nama : Gustina Dwi Pratiwi
 Tempat/Tanggal Lahir : Sukoharjo, 22 Agustus 2000
 Prodi : SI Sistem Informasi
 NPM : 021200071
 Semester : 8 (Delapan)
 No.Telp/Hp : 082176961177
 Alamat : Sukoharjo

Menyatakan dengan sesungguhnya bahwa :

1. Laporan ini saya buat dengan sebenarnya dan berdasarkan sumber yang benar.
2. Objek tempat saya melaksanakan laporan berbentuk CV/PT/Pemerintahan/SMA sederajat dan dinyatakan masih aktif beroperasi hingga saat ini
3. Data perusahaan dalam laporan skripsi ini benar adanya dan bersifat valid.
4. Laporan ini bukan merupakan hasil plagiat/menjiplak karya ilmiah orang lain
5. Laporan ini merupakan hasil kerja saya sendiri (bukan buatan/dibuatkan orang lain)
6. Buku referensi yang saya gunakan untuk laporan skripsi ini merupakan buku yang terbit dalam 5 (lima) tahun terakhir ini.
7. Semua dokumen baik berupa dokumen asli maupun salinan yang saya serahkan sebagai syarat untuk mengikuti ujian skripsi adalah dokumen yang sah dan benar.
8. Hasil karya saya yang merupakan hasil dari skripsi berupa karya tulis, program, aplikasi atau alat, setelah melalui ujian komprehensif dan revisi, bersedia untuk saya serahkan kepada lembaga melalui Kaprodi untuk dokumentasi dan kepentingan pengembangan ilmu pengetahuan dan teknologi.

Demikian pernyataan ini saya buat dalam keadaan sadar dan tanpa paksaan dari pihak manapun dan apabila di kemudian hari ternyata saya terbukti secara sah melanggar salah satu dari pernyataan ini, saya bersedia untuk menerima sanksi sesuai dengan peraturan dan hukum berlaku di negara Republik Indonesia, dan gelar akademik yang saya peroleh dari Perguruan Tinggi ini dapat dibatalkan.

Pakmbang, 27 Mei 2024

Yang menyatakan,



Gustina Dwi Pratiwi

SURAT PERNYATAAN UJIAN SKRIPSI

Yang bertanda tangan di bawah ini :

Nama : Oni Lestari
 Tempat/Tanggal Lahir : Banjarnegara / 13 Januari 2002
 Prodi : Sistem Informatika
 NPM : 021200044
 Semester : Delapan (8)
 No.Telp/Hp : 0822-1624-9580
 Alamat : Dsn III Pkl Bulian Desa Bentayan

Menyatakan dengan sesungguhnya bahwa :

1. Laporan ini saya buat dengan sebenarnya dan berdasarkan sumber yang benar.
2. Objek tempat saya melaksanakan laporan berbentuk CV/PT/Pemerintahan/SMA sederajat dan dinyatakan masih aktif beroperasi hingga saat ini
3. Data perusahaan dalam laporan skripsi ini benar adanya dan bersifat valid.
4. Laporan ini bukan merupakan hasil plagiat/menjiplak karya ilmiah orang lain
5. Laporan ini merupakan hasil kerja saya sendiri (bukan buatan/dibuatkan orang lain)
6. Buku referensi yang saya gunakan untuk laporan skripsi ini merupakan buku yang terbit dalam 5 (lima) tahun terakhir ini.
7. Semua dokumen baik berupa dokumen asli maupun salinan yang saya serahkan sebagai syarat untuk mengikuti ujian skripsi adalah dokumen yang sah dan benar.
8. Hasil karya saya yang merupakan hasil dari skripsi berupa karya tulis, program, aplikasi atau alat, setelah melalui ujian komprehensif dan revisi, bersedia untuk saya serahkan kepada lembaga melalui Kaprodi untuk dokumentasi dan kepentingan pengembangan ilmu pengetahuan dan teknologi.

Demikian pernyataan ini saya buat dalam keadaan sadar dan tanpa paksaan dari pihak manapun dan apabila di kemudian hari ternyata saya terbukti secara sah melanggar salah satu dari pernyataan ini, saya bersedia untuk menerima sanksi sesuai dengan peraturan dan hukum berlaku di negara Republik Indonesia, dan gelar akademik yang saya peroleh dari Perguruan Tinggi ini dapat dibatalkan.

Banjarnegara, 27 Mei 2024

.....
 .takan,



 METERAI
 TEMPEL
 LOCALX163140128
 Oni Lestari

 PalComTech	FORMULIR REVISI UJIAN PROPOSAL INSTITUT TEKNOLOGI DAN BISNIS PALCOMTECH
	Kode Formulir FM-IPCT-BAAK-PSB-127


**Revisi Ujian Proposal Skripsi
Mahasiswa Institut Teknologi dan Bisnis PalComTech**

Program Studi : Sistem Informasi Program Sarjana
 Tanggal Pelaksanaan : 26 Maret 2024
 Judul Proposal Skripsi : E-Commerce Dapunta Inc Menggunakan Metode RUP


No.	NPM	Nama	Semester
1	021200071	Gustina Dwi Pratiwi	VIII
2	021200044	Oni Lestari	VIII

No	Revisi	Nama Penguji	Tanda Tangan
1	Sesuaikan pendahuluan dengan pendahuluan	Yessi Snyeni	
2	Kerangka Pemikiran		
3	Lampiran ke-1		
4	Perubahan terdahulu		
	latur belakang disesuaikan dengan bentuk ruang lingkup disesuaikan juga. Aplikasi revisi RUP UML White box	Jaka P.	
			

Perubahan Judul Skripsi : E-Catalog Dapunta Inc Menggunakan Metode Rational Unified Process (RUP)
 Dapunta Inc's e-catalog uses the Rational Unified Process (RUP) method
 Palembang, 26 Maret 2024
 Ketua Program Studi.


 Dini Hani Pertawi, S.Kom., M.Kom.




*Terskop Form Revisi dikumpulkan ke BAAK setelah ditandatangani Kaprodi

 PalComTech	FORMULIR REVISI UJIAN PROPOSAL INSTITUT TEKNOLOGI DAN BISNIS PALCOMTECH
Kode Formulir FM-IPCT-BAAK-PSB-127	Institut : INSTITUT TEKNOLOGI DAN BISNIS PALCOMTECH

**Revisi Ujian Proposal Skripsi
 Mahasiswa Institut Teknologi dan Bisnis PalComTech**

Program Studi : Sistem Informasi Program Sarjana
 Tanggal Pelaksanaan : 26 Maret 2024
 Judul Proposal Skripsi : E-Commerce Dapunta Inc Menggunakan Metode RUP

No.	NPM	Nama	Semester
1	021200071	Gustina Dwi Pratiwi	VIII
2	021200044	Oni Lestari	VIII

No	Revisi	Nama Penguji	Tanda Tangan
1	Sesuai dengan pedoman	Teci Snyeni	
2	Kerangka Pemikiran		
3	Landasan teori		
4	Penelitian terdahulu		
	latar belakang disesuaikan dengan bentuk ruang lingkup disesuaikan juga. Aplikasi RUP revisi UML White box	Jaka P.	
			

Perubahan Judul Skripsi : E-Catalog Dapunta Inc Menggunakan Metode Rational Unified Process (RUP)

Palembang, 26 Maret 2024
 Ketua Program Studi,



Dini Hari Pertiwi, S.Kom., M.Kom.

*Fotokopi Form Revisi dikumpul ke BAAK setelah ditandatangani Kaprosdi

LISTING CODE

1. Dashboard Controller

```
<?php

namespace App\Http\Controllers\Admin;

use App\Contracts\Repositories\AdminWalletRepositoryInterface;
use App\Contracts\Repositories\BrandRepositoryInterface;
use App\Contracts\Repositories\CustomerRepositoryInterface;
use App\Contracts\Repositories\DeliveryManRepositoryInterface;
use App\Contracts\Repositories\OrderDetailRepositoryInterface;
use App\Contracts\Repositories\OrderRepositoryInterface;
use App\Contracts\Repositories\OrderTransactionRepositoryInterface;
use App\Contracts\Repositories\ProductRepositoryInterface;
use App\Contracts\Repositories\VendorRepositoryInterface;
use App\Contracts\Repositories\VendorWalletRepositoryInterface;
use App\Enums\ViewPaths\Admin\Dashboard;
use App\Http\Controllers\BaseController;
use App\Services\DashboardService;
use Carbon\Carbon;
use Illuminate\Contracts\View\View;
use Illuminate\Database\Eloquent\Collection;
use Illuminate\Http\JsonResponse;
use Illuminate\Http\RedirectResponse;
use Illuminate\Http\Request;
use Illuminate\Pagination\LengthAwarePaginator;
use function Laravel\Prompts>alert;

class DashboardController extends BaseController
{
    public function __construct(
        private readonly
AdminWalletRepositoryInterface $adminWalletRepo,
        private readonly
CustomerRepositoryInterface $customerRepo,
        private readonly OrderTransactionRepositoryInterface
$orderTransactionRepo,
        private readonly
ProductRepositoryInterface $productRepo,
        private readonly
DeliveryManRepositoryInterface $deliveryManRepo,
```

```

        private readonly
OrderRepositoryInterface          $orderRepo,
        private readonly
OrderDetailRepositoryInterface    $orderDetailRepo,
        private readonly
BrandRepositoryInterface          $brandRepo,
        private readonly
VendorRepositoryInterface        $vendorRepo,
        private readonly
VendorWalletRepositoryInterface   $vendorWalletRepo,
        private readonly DashboardService $dashboardService,
    )
    {
    }

    /**
     * @param Request|null $request
     * @param string|null $type
     * @return
View|Collection|LengthAwarePaginator|callable|RedirectResponse|null
     * Index function is the starting point of a controller
     */
    public function index(Request|null $request, string $type =
null):
View|Collection|LengthAwarePaginator|null|callable|RedirectResponse
    {
        return $this->dashboard();
    }

    public function dashboard(): View
    {
        $mostRatedProducts = $this->productRepo->getTopRatedList()-
>take(DASHBOARD_DATA_LIMIT);
        $topSellProduct = $this->productRepo-
>getTopSellList(relations: ['orderDetails'])-
>take(DASHBOARD_TOP_SELL_DATA_LIMIT);
        $topCustomer = $this->orderRepo-
>getTopCustomerList(relations: ['customer'], dataLimit: 'all')-
>take(DASHBOARD_DATA_LIMIT);
        $topRatedDeliveryMan = $this->deliveryManRepo-
>getTopRatedList(filters: ['seller_id' => 0], relations:
['deliveredOrders'], dataLimit: 'all')->take(DASHBOARD_DATA_LIMIT);

```

```

        $topVendorByEarning = $this->vendorWalletRepo-
>getListWhere(orderBy: ['total_earning' => 'desc'], relations:
['seller.shop'])->take(DASHBOARD_DATA_LIMIT);
        $topVendorByOrderReceived = $this->orderRepo-
>getTopVendorListByOrderReceived(relations: ['seller.shop'],
dataLimit: 'all')->take(DASHBOARD_DATA_LIMIT);

        $data = self::getOrderStatusData();
        $admin_wallet = $this->adminWalletRepo-
>getFirstWhere(params: ['admin_id' => 1]);

        $from = now()->startOfYear()->format('Y-m-d');
        $to = now()->endOfYear()->format('Y-m-d');
        $range = range(1,12);
        $label = ["Jan", "Feb", "Mar", "Apr", "May", "Jun", "Jul",
"Aug", "Sep", "Oct", "Nov", "Dec"];
        $inHouseOrderEarningArray = $this-
>getOrderStatisticsData(from:$from ,to: $to,range:
$range,type:'month',userType:'admin');
        $vendorOrderEarningArray = $this-
>getOrderStatisticsData(from: $from, to: $to, range:
$range,type:'month',userType:'seller');
        $inHouseEarning = $this->getEarning(from: $from, to: $to,
range: $range, type: 'month',userType: 'admin');
        $vendorEarning = $this->getEarning(from: $from, to: $to,
range: $range, type: 'month',userType: 'seller');
        $commissionEarn = $this->getAdminCommission(from: $from, to:
$to, range: $range, type: 'month');
        $dateType = 'yearEarn';
        $data += [
            'order' => $this->orderRepo->getListWhere(dataLimit:
'all')->count(),
            'brand' => $this->brandRepo->getListWhere(dataLimit:
'all')->count(),
            'topSellProduct' => $topSellProduct,
            'mostRatedProducts' => $mostRatedProducts,
            'topVendorByEarning' => $topVendorByEarning,
            'top_customer' => $topCustomer,
            'top_store_by_order_received' =>
$topVendorByOrderReceived,
            'topRatedDeliveryMan' => $topRatedDeliveryMan,
            'inhouse_earning' => $admin_wallet['inhouse_earning'] ??
0,
            'commission_earned' =>
$admin_wallet['commission_earned'] ?? 0,

```

```

        'delivery_charge_earned' =>
$admin_wallet['delivery_charge_earned'] ?? 0,
        'pending_amount' => $admin_wallet['pending_amount'] ??
0,
        'total_tax_collected' =>
$admin_wallet['total_tax_collected'] ?? 0,
        'getTotalCustomerCount' => $this->customerRepo-
>getList()->count(),
        'getTotalVendorCount' => $this->vendorRepo-
>getListWhere(dataLimit: 'all')->count(),
        'getTotalDeliveryManCount' => $this->deliveryManRepo-
>getListWhere(filters:['seller_id' => 0],dataLimit: 'all')->count(),
    ];
    return view(Dashboard::VIEW[VIEW], compact('data',
'inHouseEarning', 'vendorEarning',
'commissionEarn','inHouseOrderEarningArray','vendorOrderEarningArray
','label','dateType'));
}

    public function getOrderStatus(Request $request): JsonResponse
    {
        session()->put('statistics_type',
$request['statistics_type']);
        $data = self::getOrderStatusData();
        return response()->json(['view' => view('admin-
views.partials._dashboard-order-status', compact('data'))-
>render()], 200);
    }

    public function getOrderStatusData(): array
    {
        $orderQuery = $this->orderRepo->getListWhere(dataLimit:
'all');
        $storeQuery = $this->vendorRepo->getListWhere(dataLimit:
'all');
        $productQuery = $this->productRepo->getListWhere(dataLimit:
'all');
        $customerQuery = $this->customerRepo-
>getListWhere(dataLimit: 'all');
        $failedQuery = $this->orderRepo->getListWhere(filters:
['order_status' => 'failed'], dataLimit: 'all');
        $pendingQuery = $this->orderRepo->getListWhere(filters:
['order_status' => 'pending'], dataLimit: 'all');
        $returnedQuery = $this->orderRepo->getListWhere(filters:
['order_status' => 'returned'], dataLimit: 'all');
    }

```

```

        $canceledQuery = $this->orderRepo->getListWhere(filters:
['order_status' => 'canceled'], dataLimit: 'all');
        $confirmedQuery = $this->orderRepo->getListWhere(filters:
['order_status' => 'confirmed'], dataLimit: 'all');
        $deliveredQuery = $this->orderRepo->getListWhere(filters:
['order_status' => 'delivered'], dataLimit: 'all');
        $processingQuery = $this->orderRepo->getListWhere(filters:
['order_status' => 'processing'], dataLimit: 'all');
        $outForDeliveryQuery = $this->orderRepo-
>getListWhere(filters: ['order_status' => 'out_for_delivery'],
dataLimit: 'all');

        return [
            'order' => self::getCommonQueryOrderStatus($orderQuery),
            'store' => self::getCommonQueryOrderStatus($storeQuery),
            'failed' =>
self::getCommonQueryOrderStatus($failedQuery),
            'pending' =>
self::getCommonQueryOrderStatus($pendingQuery),
            'product' =>
self::getCommonQueryOrderStatus($productQuery),
            'customer' =>
self::getCommonQueryOrderStatus($customerQuery),
            'returned' =>
self::getCommonQueryOrderStatus($returnedQuery),
            'canceled' =>
self::getCommonQueryOrderStatus($canceledQuery),
            'confirmed' =>
self::getCommonQueryOrderStatus($confirmedQuery),
            'delivered' =>
self::getCommonQueryOrderStatus($deliveredQuery),
            'processing' =>
self::getCommonQueryOrderStatus($processingQuery),
            'out_for_delivery' =>
self::getCommonQueryOrderStatus($outForDeliveryQuery),
        ];
    }

    public function getCommonQueryOrderStatus($query)
    {
        $today = session()->has('statistics_type') &&
session('statistics_type') == 'today' ? 1 : 0;
        $this_month = session()->has('statistics_type') &&
session('statistics_type') == 'this_month' ? 1 : 0;
    }

```

```

        return $query->when($today, function ($query) {
            return $query->where('created_at', '>=', now()-
>startOfDay())
                                ->where('created_at', '<', now()-
>endOfDay());
        }->when($this_month, function ($query) {
            return $query->whereBetween('created_at', [now()-
>startOfMonth(), now()->endOfMonth()]);
        }->count());
    }
    public function getOrderStatistics(Request
$request):JsonResponse
    {
        $dateType = $request['type'];
        $dateTypeArray = $this->dashboardService-
>getDateTypeData(dateType:$dateType);
        $from = $dateTypeArray['from']; $to = $dateTypeArray['to'];
$type = $dateTypeArray['type']; $range = $dateTypeArray['range'];
        $inHouseOrderEarningArray = $this-
>getOrderStatisticsData(from: $from, to: $to, range: $range, type:
$type,userType:'admin');
        $vendorOrderEarningArray = $this-
>getOrderStatisticsData(from: $from, to: $to, range: $range, type:
$type,userType:'seller');
        $label = $dateTypeArray['keyRange'] ?? [];
        $inHouseOrderEarningArray =
array_values($inHouseOrderEarningArray);
        $vendorOrderEarningArray =
array_values($vendorOrderEarningArray);
        return response()->json([
            'view' => view(Dashboard::ORDER_STATISTICS[VIEW],
compact('inHouseOrderEarningArray','vendorOrderEarningArray','label'
,'dateType'))->render(),
        ]);
    }
    public function getEarningStatistics(Request
$request):JsonResponse
    {
        $dateType = $request['type'];
        $dateTypeArray = $this->dashboardService-
>getDateTypeData(dateType:$dateType);
        $from = $dateTypeArray['from']; $to = $dateTypeArray['to'];
$type = $dateTypeArray['type']; $range = $dateTypeArray['range'];
        $inHouseEarning = $this->getEarning(from: $from, to: $to,
range: $range, type: $type,userType: 'admin');

```

```

        $vendorEarning = $this->getEarning(from: $from, to: $to,
range: $range, type: $type, userType: 'seller');
        $commissionEarn = $this->getAdminCommission(from: $from, to:
$to, range: $range, type: $type);
        $label = $dateTypeArray['keyRange'] ?? [];
        $inHouseEarning = array_values($inHouseEarning);
        $vendorEarning = array_values($vendorEarning);
        $commissionEarn = array_values($commissionEarn);
        return response()->json([
            'view' => view(Dashboard::EARNING_STATISTICS[VIEW],
compact('inHouseEarning', 'vendorEarning', 'commissionEarn', 'label', 'd
ateType'))->render(),
        ]);
    }
    protected function
getOrderStatisticsData($from,$to,$range,$type,$userType):array
    {
        $orderEarnings = $this->orderRepo->getListWhereBetween(
            filters: [
                'seller_is'=>$userType,
                'payment_status' => 'paid'
            ],
            selectColumn: 'order_amount',
            whereBetween: 'created_at',
            whereBetweenFilters: [$from, $to],
        );
        $orderEarningArray = [];
        foreach ($range as $value){
            $matchingEarnings = $orderEarnings->where($type,
$value);
            if ($matchingEarnings->count() > 0) {
                $orderEarningArray[$value] =
            usdToDefaultCurrency($matchingEarnings->sum('sums'));
            } else {
                $orderEarningArray[$value] = 0;
            }
        }
        return $orderEarningArray;
    }

    protected function getEarning(string|Carbon $from, string|Carbon
$to, array $range, string $type,$userType):array
    {
        $earning = $this->orderTransactionRepo->getListWhereBetween(
            filters: [

```

```

        'seller_is'=>$userType,
        'status'=>'disburse'
    ],
    selectColumn: 'seller_amount',
    whereBetween: 'created_at',
    whereBetweenFilters: [$from, $to],
    );
    return $this->dashboardService->getDateWiseAmount(range:
$range,type: $type,amountArray: $earning);
}

/**
 * @param string|Carbon $from
 * @param string|Carbon $to
 * @param array $range
 * @param string $type
 * @return array
 */
protected function getAdminCommission(string|Carbon $from,
string|Carbon $to, array $range, string $type ):array
{
    $commissionGiven = $this->orderTransactionRepo-
>getListWhereBetween(
        filters: [
            'seller_is'=>'seller',
            'status'=>'disburse'
        ],
        selectColumn: 'admin_commission',
        whereBetween: 'created_at',
        whereBetweenFilters: [$from, $to],
    );
    return $this->dashboardService->getDateWiseAmount(range:
$range,type: $type,amountArray: $commissionGiven);
}
}

```


2. Profil Controller

```
3. <?php
4.
5. namespace App\Http\Controllers\Admin;
6.
7. use App\Contracts\Repositories\AdminRepositoryInterface;
8. use App\Enums\ViewPaths\Admin\Profile;
9. use App\Http\Controllers\BaseController;
10. use App\Http\Requests\Admin\AdminPasswordRequest;
11. use App\Http\Requests\Admin\AdminRequest;
12. use App\Services\AdminService;
13. use Brian2694\Toastr\Facades\Toastr;
14. use Illuminate\Contracts\View\View;
15. use Illuminate\Database\Eloquent\Collection;
16. use Illuminate\Http\RedirectResponse;
17. use Illuminate\Http\Request;
18. use Illuminate\Pagination\LengthAwarePaginator;
19.
20. class ProfileController extends BaseController
21. {
22.     public function __construct(
23.         private readonly AdminRepositoryInterface $adminRepo,
24.         private readonly AdminService $adminService,
25.     )
26.     {
27.     }
28.     public function index(?Request $request, string $type =
        null):
        View|Collection|LengthAwarePaginator|null|callable|RedirectRes
        ponse
29.     {
30.         return $this->getListView();
31.     }
32.     public function getListView():View
33.     {
34.         $admin = $this->adminRepo-
            >getFirstWhere(['id'=>auth('admin')->id()]);
35.         return view(Profile::INDEX[VIEW],compact('admin'));
36.     }
37.
38.     /**
39.      * @param string|int $id
40.      * @return View|RedirectResponse
41.      */
```

```
42.     public function getView(string|int
43.         $id):View|RedirectResponse
44.     {
45.         $admin = $this->adminRepo->getFirstWhere(['id' =>
46.             $id]);
47.         $shopBanner = getWebConfig('shop_banner');
48.         return
49.             view(Profile::UPDATE[VIEW],compact('admin','shopBanner'));
50.     }
51.
52.     /**
53.      * @param AdminRequest $request
54.      * @param string|int $id
55.      * @return RedirectResponse
56.      */
57.     public function update(AdminRequest $request, string|int
58.         $id):RedirectResponse
59.     {
60.         $admin = $this->adminRepo->getFirstWhere(['id' =>
61.             $id]);
62.         $this->adminRepo->update(id: $id, data: $this->
63.             >adminService->getAdminDataForUpdate(request: $request, admin:
64.             $admin));
65.         Toastr::success(translate('profile_updated_successfull
66.             y'));
67.         return redirect()->back();
68.     }
69.
70.     /**
71.      * @param AdminPasswordRequest $request
72.      * @param string|int $id
73.      * @return RedirectResponse
74.      */
75.     public function updatePassword(AdminPasswordRequest
76.         $request , string|int $id):RedirectResponse
77.     {
78.         $this->adminRepo->update(id:$id,data:$this->
79.             >adminService->getAdminPasswordData(request:$request));
80.         Toastr::success(translate('admin_password_updated_succ
81.             essfully'));
82.         return redirect()->back();
83.     }
84. }
```

76.

3 Chatting Controller

```
<?php

namespace App\Http\Controllers\Admin;

use App\Contracts\Repositories\ChattingRepositoryInterface;
use App\Contracts\Repositories\CustomerRepositoryInterface;
use App\Contracts\Repositories\DeliveryManRepositoryInterface;
use App\Contracts\Repositories\ShopRepositoryInterface;
use App\Enums\ViewPaths\Admin\Chatting;
use App\Events\ChattingEvent;
use App\Http\Controllers\BaseController;
use App\Http\Requests\Admin\ChattingRequest;
use App\Services\ChattingService;
use App\Traits\PushNotificationTrait;
use Illuminate\Contracts\View\View;
use Illuminate\Database\Eloquent\Collection;
use Illuminate\Http\JsonResponse;
use Illuminate\Http\RedirectResponse;
use Illuminate\Http\Request;
use Illuminate\Pagination\LengthAwarePaginator;

class ChattingController extends BaseController
{
    use PushNotificationTrait;

    /**
     * @param ChattingRepositoryInterface $chattingRepo
     * @param ShopRepositoryInterface $shopRepo
     * @param ChattingService $chattingService
     * @param DeliveryManRepositoryInterface $deliveryManRepo
     * @param CustomerRepositoryInterface $customerRepo
     */
    public function __construct(
        private readonly ChattingRepositoryInterface $chattingRepo,
        private readonly ShopRepositoryInterface $shopRepo,
        private readonly ChattingService $chattingService,
        private readonly DeliveryManRepositoryInterface
$deliveryManRepo,
        private readonly CustomerRepositoryInterface $customerRepo,
```

```

)
{
}

/**
 * @param Request|null $request
 * @param string|array|null $type
 * @return
View|Collection|LengthAwarePaginator|callable|RedirectResponse|null
 */
public function index(?Request $request, string|array $type =
null):
View|Collection|LengthAwarePaginator|null|callable|RedirectResponse
{
    return $this->getListView(type:$type);
}

/**
 * @param string|array $type
 * @return View
 */
public function getListView(string|array $type):View
{
    $shop = $this->shopRepo->getFirstWhere(params: ['seller_id'
=> auth('seller')->id()]);
    $adminId = 0;
    if ($type == 'delivery-man') {
        $allChattingUsers = $this->chattingRepo-
>getListWhereNotNull(
            orderBy: ['created_at' => 'DESC'],
            filters: ['admin_id' => $adminId],
            whereNotNull: ['delivery_man_id', 'admin_id'],
            relations: ['deliveryMan'],
            dataLimit: 'all'
        )->unique('delivery_man_id');

        if (count($allChattingUsers) > 0) {
            $lastChatUser = $allChattingUsers[0]->deliveryMan;
            $this->chattingRepo->updateAllWhere(
                params: ['admin_id' => $adminId,
'delivery_man_id' => $lastChatUser['id']],
                data: ['seen_by_admin' => 1]
            );
        }
    }
}

```

```

        $chattingMessages = $this->chattingRepo-
>getListWhereNotNull(
            orderBy: ['created_at' => 'DESC'],
            filters: ['admin_id' => $adminId,
'delivery_man_id'=>$lastChatUser->id],
            whereNotNull: ['delivery_man_id', 'admin_id'],
            relations: ['deliveryMan'],
            dataLimit: 'all'
        );

        return view(Chatting::INDEX[VIEW], [
            'userType' => $type,
            'allChattingUsers' => $allChattingUsers,
            'lastChatUser' => $lastChatUser,
            'chattingMessages' => $chattingMessages,
        ]);
    }
    } elseif ($type == 'customer') {
        $allChattingUsers = $this->chattingRepo-
>getListWhereNotNull(
            orderBy: ['created_at' => 'DESC'],
            filters: ['admin_id' => $adminId],
            whereNotNull: ['user_id', 'admin_id'],
            relations: ['customer'],
            dataLimit: 'all'
        )->unique('user_id');

        if (count($allChattingUsers) > 0) {
            $lastChatUser = $allChattingUsers[0]->customer;
            $this->chattingRepo->updateAllWhere(
                params: ['admin_id' => $adminId, 'user_id' =>
$lastChatUser['id']],
                data: ['seen_by_admin' => 1]
            );

            $chattingMessages = $this->chattingRepo-
>getListWhereNotNull(
                orderBy: ['created_at' => 'DESC'],
                filters: ['admin_id' => $adminId,
'admin_id'=>$lastChatUser->id],
                whereNotNull: ['user_id', 'admin_id'],
                relations: ['customer'],
                dataLimit: 'all'
            );
        }
    }
}

```

```

        return view(Chatting::INDEX[VIEW], [
            'userType' => $type,
            'allChattingUsers' => $allChattingUsers,
            'lastChatUser' => $lastChatUser,
            'chattingMessages' => $chattingMessages,
        ]);
    }
}
return view(Chatting::INDEX[VIEW], compact('shop'));
}

/**
 * @param Request $request
 * @return JsonResponse
 */
public function getMessageByUser(Request $request): JsonResponse
{
    $adminId = 0;
    $data = [];
    if ($request->has(key: 'delivery_man_id')) {
        $getUser = $this->deliveryManRepo->getFirstWhere(params:
['id' => $request['delivery_man_id']]);
        $this->chattingRepo->updateAllWhere(
            params: ['admin_id' => $adminId, 'delivery_man_id'
=> $request['delivery_man_id']],
            data: ['seen_by_admin' => 1]);

        $chattingMessages = $this->chattingRepo-
>getListWhereNotNull(
            orderBy: ['created_at' => 'DESC'],
            filters: ['admin_id' => $adminId, 'delivery_man_id'
=> $request['delivery_man_id']],
            whereNotNull: ['delivery_man_id', 'admin_id'],
            dataLimit: 'all'
        );
        $data = self::getRenderMessagesView(user: $getUser,
message: $chattingMessages, type: 'delivery_man');
    } elseif ($request->has(key: 'user_id')) {
        $getUser = $this->customerRepo->getFirstWhere(params:
['id' => $request['user_id']]);
        $this->chattingRepo->updateAllWhere(
            params: ['admin_id' => $adminId, 'user_id' =>
$request['user_id']],
            data: ['seen_by_admin' => 1]
    }
}

```

```

    );

    $chattingMessages = $this->chattingRepo-
>getListWhereNotNull(
        orderBy: ['created_at' => 'DESC'],
        filters: ['admin_id' => $adminId, 'user_id' =>
$request['user_id']],
        whereNotNull: ['user_id', 'admin_id'],
        dataLimit: 'all'
    );
    $data = self::getRenderMessagesView(user: $getUser,
message: $chattingMessages, type: 'customer');
    }
    return response()->json($data);
}

/**
 * @param ChattingRequest $request
 * @return JsonResponse
 */
public function addAdminMessage(ChattingRequest
$request):JsonResponse
{
    $data = [];
    $messageForm = (object)[
        'f_name'=>'admin',
        'shop'=> [
            'name'=> getWebConfig(name: 'company_name')
        ]
    ];
    if ($request->has(key: 'delivery_man_id')) {
        $this->chattingRepo->add(
            data: $this->chattingService->addChattingData(
                request: $request,
                type:'delivery-man',
            )
        );
        $deliveryMan = $this->deliveryManRepo-
>getFirstWhere(params: ['id' => $request['delivery_man_id']]);
        ChattingEvent::dispatch('message_from_admin',
'delivery_man', $deliveryMan, $messageForm);

        $chattingMessages = $this->chattingRepo-
>getListWhereNotNull(
            orderBy: ['created_at' => 'DESC'],

```

```

        filters: ['admin_id' => 0, 'delivery_man_id' =>
$request['delivery_man_id']],
        whereNotNull: ['delivery_man_id', 'admin_id'],
        dataLimit: 'all'
    );
    $data = self::getRenderMessagesView(user: $deliveryMan,
message: $chattingMessages, type: 'delivery_man');
    } elseif ($request->has(key: 'user_id')) {
        $this->chattingRepo->add(
            data: $this->chattingService->addChattingData(
                request: $request,
                type: 'customer',
            )
        );
        $customer = $this->customerRepo->getFirstWhere(params:
['id' => $request['user_id']]);
        ChattingEvent::dispatch('message_from_admin',
'customer', $customer, $messageForm);

        $chattingMessages = $this->chattingRepo-
>getListWhereNotNull(
            orderBy: ['created_at' => 'DESC'],
            filters: ['admin_id' => 0, 'user_id' =>
$request['user_id']],
            whereNotNull: ['user_id', 'admin_id'],
            dataLimit: 'all'
        );
        $data = self::getRenderMessagesView(user: $customer,
message: $chattingMessages, type: 'customer');
    }
    return response()->json($data);
}

/**
 * @param string $tableName
 * @param string $orderBy
 * @param string|int|null $id
 * @return Collection
 */
protected function getChatList(string $tableName, string
$orderBy, string|int $id = null) :Collection
{
    $adminId = 0;
    $columnName = $tableName == 'users' ? 'user_id' :
'delivery_man_id';

```



```

        $filters = isset($id) ? ['chattings.admin_id' => $adminId,
$columnName => $id] : ['chattings.admin_id' => $adminId];
        return $this->chattingRepo->getListBySelectWhere(
            joinColumn: [$tableName, $tableName . '.id', '=',
'chattings.' . $columnName],
            select: ['chattings.*', $tableName . '.f_name',
$tableName . '.l_name', $tableName . '.image', $tableName .
'.country_code', $tableName . '.phone'],
            filters: $filters,
            orderBy: ['chattings.created_at' => $orderBy],
        );
    }

    /**
     * @param object $user
     * @param object $message
     * @param string $type
     * @return array
     */
    protected function getRenderMessagesView(object $user, object
$message, string $type): array
    {
        $userData = ['name' => $user['f_name'],
'.user['l_name'], 'phone' => $user['country_code'].$user['phone']];

        if ($type == 'customer') {
            $userData['image'] = getValidImage(path:
'storage/app/public/profile/' . ($user['image']), type: 'backend-
profile');
        } else {
            $userData['image'] = getValidImage(path:
'storage/app/public/delivery-man/' . ($user['image']), type:
'backend-profile');
        }

        return [
            'userData' => $userData,
            'chattingMessages' => view('admin-
views.chatting.messages', [
                'lastChatUser' => $user,
                'userType' => $type,
                'chattingMessages' => $message
            ])->render(),
        ];
    }
}

```

```

public function getNewNotification(): JsonResponse
{
    $chatting = $this->chattingRepo->getListWhereNotNull(
        filters: ['admin_id' => 0, 'seen_by_admin' => 0,
'notification_receiver' => 'admin', 'seen_notification' => 0],
        whereNotNull: ['admin_id'],
    )->count();

    $this->chattingRepo->updateListWhereNotNull(
        filters: ['admin_id' => 0, 'seen_by_admin' => 0,
'notification_receiver' => 'admin', 'seen_notification' => 0],
        whereNotNull: ['admin_id'],
        data: ['seen_notification' => 1]
    );

    return response()->json([
        'newMessagesExist' => $chatting,
        'message' => $chatting > 1 ? $chatting .'
.translate('New_Message') : translate('New_Message'),
    ]);
}
}

```

4 Order Controller

```

<?php

namespace App\Http\Controllers\Admin\Settings;

use App\Contracts\Repositories\BusinessSettingRepositoryInterface;
use App\Enums\ViewPaths\Admin\BusinessSettings;
use App\Http\Controllers\BaseController;
use Brian2694\Toastr\Facades\Toastr;
use Illuminate\Contracts\View\View;
use Illuminate\Http\RedirectResponse;
use Illuminate\Http\Request;

class OrderSettingsController extends BaseController
{
    public function __construct(

```

```
        private readonly BusinessSettingRepositoryInterface
$businessSettingRepo,
    )
    {
    }

    /**
     * @param Request|null $request
     * @param string|null $type
     * @return View Index function is the starting point of a
controller
     * Index function is the starting point of a controller
     */
    public function index(Request|null $request, string $type =
null): View
    {
        return $this->getView();
    }

    public function getView(): View
    {
        return view(BusinessSettings::ORDER_VIEW[VIEW]);
    }

    public function update(Request $request): RedirectResponse
    {
        $this->businessSettingRepo->updateOrInsert(type:
'billing_input_by_customer', value: $request-
>get('billing_input_by_customer', 0));
        $this->businessSettingRepo->updateOrInsert(type:
'minimum_order_amount_status', value: $request-
>get('minimum_order_amount_status', 0));
        $this->businessSettingRepo->updateOrInsert(type:
'refund_day_limit', value: $request->get('refund_day_limit', 0));
        $this->businessSettingRepo->updateOrInsert(type:
'order_verification', value: $request->get('order_verification',
0));
        $this->businessSettingRepo->updateOrInsert(type:
'free_delivery_status', value: $request->get('free_delivery_status',
0));
        $this->businessSettingRepo->updateOrInsert(type:
'free_delivery_responsibility', value:
$request['free_delivery_responsibility']);
        $this->businessSettingRepo->updateOrInsert(type:
'guest_checkout', value: $request->get('guest_checkout', 0));
    }
}
```

```

        $this->businessSettingRepo->updateOrInsert(type:
'free_delivery_over_amount_seller', value: currencyConverter(amount:
$request['free_delivery_over_amount_seller']) ?? 0);
        Toastr::success(translate('successfully_updated'));
        return back();
    }
}

```

5 Payment Method Controller

```

<?php

namespace App\Http\Controllers\Admin;

use App\Utils\Helpers;
use App\Enums\GlobalConstant;
use App\Http\Controllers\Controller;
use App\Models\Setting;
use App\Models\BusinessSetting;
use App\Traits\Processor;
use Brian2694\Toastr\Facades\Toastr;
use Illuminate\Http\RedirectResponse;
use Illuminate\Http\Request;
use Illuminate\Support\Facades\Validator;
use Illuminate\Validation\ValidationException;

class PaymentMethodController extends Controller
{
    use Processor;

    public function index()
    {
        $payment_published_status =
config('get_payment_publish_status') ?? 0;
        $payment_gateway_published_status =
isset($payment_published_status[0]['is_published']) ?
$payment_published_status[0]['is_published'] : 0;
    }
}

```

```

        $payment_gateways = Setting::whereIn('settings_type',
['payment_config']->whereIn('key_name',
GlobalConstant::DEFAULT_PAYMENT_GATEWAYS)->get());

        $payment_gateways = $payment_gateways->sortBy(function
($item) {
            return count($item['live_values']);
        })->values()->all();

        $routes = config('addon_admin_routes');
        $desiredName = 'payment_setup';
        $payment_url = '';

        foreach ($routes as $routeArray) {
            foreach ($routeArray as $route) {
                if ($route['name'] === $desiredName) {
                    $payment_url = $route['url'];
                    break 2;
                }
            }
        }

        return view('admin-views.business-settings.payment-
method.index',
            compact('payment_gateways',
'payment_gateway_published_status', 'payment_url'));
    }

    public function update(Request $request)
    {
        BusinessSetting::updateOrCreate(['type' =>
'cash_on_delivery'], [
            'value' => json_encode(['status' =>
$request['cash_on_delivery'] ?? 0]),
            'updated_at' => now()
        ]);

        BusinessSetting::updateOrCreate(['type' =>
'digital_payment'], [
            'value' => json_encode(['status' =>
$request['digital_payment'] ?? 0]),
            'updated_at' => now()
        ]);
    }

```

```

        BusinessSetting::updateOrCreate(['type' =>
'offline_payment'], [
            'value' => json_encode(['status' =>
$request['offline_payment'] ?? 0]),
            'updated_at' => now()
        ]);

        Toastr::success(translate('successfully_updated'));
        return back();
    }

    /**
     * Display a listing of the resource.
     * @param Request $request
     * @return RedirectResponse
     * @throws ValidationException
     */
    public function payment_config_set(Request $request)
    {
        collect(['status'])->each(fn($item, $key) => $request[$item]
= $request->has($item) ? (int)$request[$item] : 0);
        $validation = [
            'gateway' =>
'required|in:ssl_commerz,sixcash,worldpay,payfast,swish,esewa,maxica
sh,hubtel,viva_wallet,tap,thawani,moncash,pvit,ccavenue,foloosi,iyzi
_pay,xendit,fatoorah,hyper_pay,amazon_pay,paypal,stripe,razor_pay,se
nang_pay,paytabs,paystack,paymob_accept,paytm,flutterwave,liqpay,bka
sh,mercadopago,cash_after_service,digital_payment,momo',
            'mode' => 'required|in:live,test'
        ];

        $additional_data = [];

        if ($request['gateway'] == 'ssl_commerz') {
            $additional_data = [
                'status' => 'required|in:1,0',
                'store_id' => 'required',
                'store_password' => 'required'
            ];
        } elseif ($request['gateway'] == 'paypal') {
            $additional_data = [
                'status' => 'required|in:1,0',
                'client_id' => 'required',
                'client_secret' => 'required'
            ];
        }
    }

```

```
    } elseif ($request['gateway'] == 'stripe') {
        $additional_data = [
            'status' => 'required|in:1,0',
            'api_key' => 'required',
            'published_key' => 'required',
        ];
    } elseif ($request['gateway'] == 'razor_pay') {
        $additional_data = [
            'status' => 'required|in:1,0',
            'api_key' => 'required',
            'api_secret' => 'required'
        ];
    } elseif ($request['gateway'] == 'senang_pay') {
        $additional_data = [
            'status' => 'required|in:1,0',
            'callback_url' => 'required',
            'secret_key' => 'required',
            'merchant_id' => 'required'
        ];
    } elseif ($request['gateway'] == 'paytabs') {
        $additional_data = [
            'status' => 'required|in:1,0',
            'profile_id' => 'required',
            'server_key' => 'required',
            'base_url' => 'required'
        ];
    } elseif ($request['gateway'] == 'paystack') {
        $additional_data = [
            'status' => 'required|in:1,0',
            'public_key' => 'required',
            'secret_key' => 'required',
            'merchant_email' => 'required'
        ];
    } elseif ($request['gateway'] == 'paymob_accept') {
        $additional_data = [
            'status' => 'required|in:1,0',
            'callback_url' => 'required',
            'api_key' => 'required',
            'iframe_id' => 'required',
            'integration_id' => 'required',
            'hmac' => 'required'
        ];
    } elseif ($request['gateway'] == 'mercadopago') {
        $additional_data = [
            'status' => 'required|in:1,0',
```

```
        'access_token' => 'required',
        'public_key' => 'required'
    ];
} elseif ($request['gateway'] == 'liqpay') {
    $additional_data = [
        'status' => 'required|in:1,0',
        'private_key' => 'required',
        'public_key' => 'required'
    ];
} elseif ($request['gateway'] == 'flutterwave') {
    $additional_data = [
        'status' => 'required|in:1,0',
        'secret_key' => 'required',
        'public_key' => 'required',
        'hash' => 'required'
    ];
} elseif ($request['gateway'] == 'paytm') {
    $additional_data = [
        'status' => 'required|in:1,0',
        'merchant_key' => 'required',
        'merchant_id' => 'required',
        'merchant_website_link' => 'required'
    ];
} elseif ($request['gateway'] == 'bkash') {
    $additional_data = [
        'status' => 'required|in:1,0',
        'app_key' => 'required',
        'app_secret' => 'required',
        'username' => 'required',
        'password' => 'required',
    ];
} elseif ($request['gateway'] == 'cash_after_service') {
    $additional_data = [
        'status' => 'required|in:1,0'
    ];
} elseif ($request['gateway'] == 'digital_payment') {
    $additional_data = [
        'status' => 'required|in:1,0'
    ];
} elseif ($request['gateway'] == 'momo') {
    $additional_data = [
        'status' => 'required|in:1,0',
        'api_key' => 'required',
        'api_user' => 'required',
        'subscription_key' => 'required',
    ];
```



```
];
} elseif ($request['gateway'] == 'hyper_pay') {
    $additional_data = [
        'status' => 'required|in:1,0',
        'entity_id' => 'required',
        'access_code' => 'required',
    ];
} elseif ($request['gateway'] == 'amazon_pay') {
    $additional_data = [
        'status' => 'required|in:1,0',
        'pass_phrase' => 'required',
        'access_code' => 'required',
        'merchant_identifier' => 'required',
    ];
} elseif ($request['gateway'] == 'sixcash') {
    $additional_data = [
        'status' => 'required|in:1,0',
        'public_key' => 'required',
        'secret_key' => 'required',
        'merchant_number' => 'required',
        'base_url' => 'required',
    ];
} elseif ($request['gateway'] == 'worldpay') {
    $additional_data = [
        'status' => 'required|in:1,0',
        'OrgUnitId' => 'required',
        'jwt_issuer' => 'required',
        'mac' => 'required',
        'merchantCode' => 'required',
        'xml_password' => 'required',
    ];
} elseif ($request['gateway'] == 'payfast') {
    $additional_data = [
        'status' => 'required|in:1,0',
        'merchant_id' => 'required',
        'secured_key' => 'required',
    ];
} elseif ($request['gateway'] == 'swish') {
    $additional_data = [
        'status' => 'required|in:1,0',
        'number' => 'required',
    ];
} elseif ($request['gateway'] == 'esewa') {
    $additional_data = [
        'status' => 'required|in:1,0',
```

```
        'merchantCode' => 'required',
    ];
} elseif ($request['gateway'] == 'maxicash') {
    $additional_data = [
        'status' => 'required|in:1,0',
        'merchantId' => 'required',
        'merchantPassword' => 'required',
    ];
} elseif ($request['gateway'] == 'hubtel') {
    $additional_data = [
        'status' => 'required|in:1,0',
        'account_number' => 'required',
        'api_id' => 'required',
        'api_key' => 'required',
    ];
} elseif ($request['gateway'] == 'viva_wallet') {
    $additional_data = [
        'status' => 'required|in:1,0',
        'client_id' => 'required',
        'client_secret' => 'required',
        'source_code' => 'required',
    ];
} elseif ($request['gateway'] == 'tap') {
    $additional_data = [
        'status' => 'required|in:1,0',
        'secret_key' => 'required',
    ];
} elseif ($request['gateway'] == 'thawani') {
    $additional_data = [
        'status' => 'required|in:1,0',
        'public_key' => 'required',
        'private_key' => 'required',
    ];
} elseif ($request['gateway'] == 'moncash') {
    $additional_data = [
        'status' => 'required|in:1,0',
        'client_id' => 'required',
        'secret_key' => 'required',
    ];
} elseif ($request['gateway'] == 'pvit') {
    $additional_data = [
        'status' => 'required|in:1,0',
        'mc_tel_merchant' => 'required',
        'access_token' => 'required',
        'mc_merchant_code' => 'required',
    ];
}
```

```

];
} elseif ($request['gateway'] == 'ccavenue') {
    $additional_data = [
        'status' => 'required|in:1,0',
        'merchant_id' => 'required',
        'working_key' => 'required',
        'access_code' => 'required',
    ];
} elseif ($request['gateway'] == 'foloosi') {
    $additional_data = [
        'status' => 'required|in:1,0',
        'merchant_key' => 'required',
    ];
} elseif ($request['gateway'] == 'iyzi_pay') {
    $additional_data = [
        'status' => 'required|in:1,0',
        'api_key' => 'required',
        'secret_key' => 'required',
        'base_url' => 'required',
    ];
} elseif ($request['gateway'] == 'xendit') {
    $additional_data = [
        'status' => 'required|in:1,0',
        'api_key' => 'required',
    ];
} elseif ($request['gateway'] == 'fatorah') {
    $additional_data = [
        'status' => 'required|in:1,0',
        'api_key' => 'required',
    ];
}
}

$request->validate(array_merge($validation,
$additional_data));

$settings = Setting::where('key_name', $request['gateway'])-
>where('settings_type', 'payment_config')->first();

$additional_data_image = $settings['additional_data'] !=
null ? json_decode($settings['additional_data']) : null;

if( !$additional_data_image ||
!isset($additional_data_image->gateway_image) ||
(isset($additional_data_image->gateway_image) &&
$additional_data_image->gateway_image == '')) ||

```

```

(isset($additional_data_image->gateway_image) &&
!file_exists(base_path("storage/app/public/payment_modules/gateway_i
mage/".$additional_data_image->gateway_image))){
    $request->validate([
        'gateway_image' => 'required',
    ]);
}
$request->validate(['gateway_title' => 'required']);

if ($request->has('gateway_image')) {
    $gateway_image = $this-
>file_uploader('payment_modules/gateway_image/', 'png',
$request['gateway_image'], $additional_data_image != null ?
$additional_data_image->gateway_image : '');
} else {
    $gateway_image = $additional_data_image != null ?
$additional_data_image->gateway_image : '';
}

$payment_additional_data = [
    'gateway_title' => $request['gateway_title'],
    'gateway_image' => $gateway_image,
];

$validator = Validator::make($request->all(),
array_merge($validation, $additional_data));

Setting::updateOrCreate(['key_name' => $request['gateway'],
'settings_type' => 'payment_config'], [
    'key_name' => $request['gateway'],
    'live_values' => $validator->validate(),
    'test_values' => $validator->validate(),
    'settings_type' => 'payment_config',
    'mode' => $request['mode'],
    'is_active' => $request['status'] ?? 0,
    'additional_data' =>
json_encode($payment_additional_data),
]);

Toastr::success(GATEWAYS_DEFAULT_UPDATE_200['message']);
return back();
}
}

```

6 Product Wishlist Controller

```
<?php

namespace App\Http\Controllers\Admin;

use App\Utils\Helpers;
use App\Enums\GlobalConstant;
use App\Http\Controllers\Controller;
use App\Models\Setting;
use App\Models\BusinessSetting;
use App\Traits\Processor;
use Brian2694\Toastr\Facades\Toastr;
use Illuminate\Http\RedirectResponse;
use Illuminate\Http\Request;
use Illuminate\Support\Facades\Validator;
use Illuminate\Validation\ValidationException;

class PaymentMethodController extends Controller
{
    use Processor;

    public function index()
    {
        $payment_published_status =
config('get_payment_publish_status') ?? 0;
        $payment_gateway_published_status =
isset($payment_published_status[0]['is_published']) ?
$payment_published_status[0]['is_published'] : 0;

        $payment_gateways = Setting::whereIn('settings_type',
['payment_config'])->whereIn('key_name',
GlobalConstant::DEFAULT_PAYMENT_GATEWAYS)->get();

        $payment_gateways = $payment_gateways->sortBy(function
($item) {
            return count($item['live_values']);
        }->values()->all());

        $routes = config('addon_admin_routes');
        $desiredName = 'payment_setup';
        $payment_url = '';

        foreach ($routes as $routeArray) {
```

```

        foreach ($routeArray as $route) {
            if ($route['name'] === $desiredName) {
                $payment_url = $route['url'];
                break 2;
            }
        }
    }

    return view('admin-views.business-settings.payment-
method.index',
        compact('payment_gateways',
'payment_gateway_published_status', 'payment_url'));
}

public function update(Request $request)
{
    BusinessSetting::updateOrCreate(['type' =>
'cash_on_delivery'], [
        'value' => json_encode(['status' =>
$request['cash_on_delivery'] ?? 0]),
        'updated_at' => now()
    ]);

    BusinessSetting::updateOrCreate(['type' =>
'digital_payment'], [
        'value' => json_encode(['status' =>
$request['digital_payment'] ?? 0]),
        'updated_at' => now()
    ]);

    BusinessSetting::updateOrCreate(['type' =>
'offline_payment'], [
        'value' => json_encode(['status' =>
$request['offline_payment'] ?? 0]),
        'updated_at' => now()
    ]);

    Toastr::success(translate('successfully_updated'));
    return back();
}

/**
 * Display a listing of the resource.
 * @param Request $request

```

```

* @return RedirectResponse
* @throws ValidationException
*/
public function payment_config_set(Request $request)
{
    collect(['status'])->each(fn($item, $key) => $request[$item]
= $request->has($item) ? (int)$request[$item] : 0);
    $validation = [
        'gateway' =>
'required|in:ssl_commerz,sixcash,worldpay,payfast,swish,esewa,maxica
sh,hubtel,viva_wallet,tap,thawani,moncash,pvit,ccavenue,foloosi,iyzi
_pay,xendit,fatoorah,hyper_pay,amazon_pay,paypal,stripe,razor_pay,se
nang_pay,paytabs,pystack,paymob_accept,paytm,flutterwave,liqpay,bka
sh,mercadopago,cash_after_service,digital_payment,momo',
        'mode' => 'required|in:live,test'
    ];

    $additional_data = [];

    if ($request['gateway'] == 'ssl_commerz') {
        $additional_data = [
            'status' => 'required|in:1,0',
            'store_id' => 'required',
            'store_password' => 'required'
        ];
    } elseif ($request['gateway'] == 'paypal') {
        $additional_data = [
            'status' => 'required|in:1,0',
            'client_id' => 'required',
            'client_secret' => 'required'
        ];
    } elseif ($request['gateway'] == 'stripe') {
        $additional_data = [
            'status' => 'required|in:1,0',
            'api_key' => 'required',
            'published_key' => 'required',
        ];
    } elseif ($request['gateway'] == 'razor_pay') {
        $additional_data = [
            'status' => 'required|in:1,0',
            'api_key' => 'required',
            'api_secret' => 'required'
        ];
    } elseif ($request['gateway'] == 'senang_pay') {
        $additional_data = [

```

```
        'status' => 'required|in:1,0',
        'callback_url' => 'required',
        'secret_key' => 'required',
        'merchant_id' => 'required'
    ];
} elseif ($request['gateway'] == 'paytabs') {
    $additional_data = [
        'status' => 'required|in:1,0',
        'profile_id' => 'required',
        'server_key' => 'required',
        'base_url' => 'required'
    ];
} elseif ($request['gateway'] == 'paystack') {
    $additional_data = [
        'status' => 'required|in:1,0',
        'public_key' => 'required',
        'secret_key' => 'required',
        'merchant_email' => 'required'
    ];
} elseif ($request['gateway'] == 'paymob_accept') {
    $additional_data = [
        'status' => 'required|in:1,0',
        'callback_url' => 'required',
        'api_key' => 'required',
        'iframe_id' => 'required',
        'integration_id' => 'required',
        'hmac' => 'required'
    ];
} elseif ($request['gateway'] == 'mercadopago') {
    $additional_data = [
        'status' => 'required|in:1,0',
        'access_token' => 'required',
        'public_key' => 'required'
    ];
} elseif ($request['gateway'] == 'liqpay') {
    $additional_data = [
        'status' => 'required|in:1,0',
        'private_key' => 'required',
        'public_key' => 'required'
    ];
} elseif ($request['gateway'] == 'flutterwave') {
    $additional_data = [
        'status' => 'required|in:1,0',
        'secret_key' => 'required',
        'public_key' => 'required',
```



```
        'hash' => 'required'
    ];
} elseif ($request['gateway'] == 'paytm') {
    $additional_data = [
        'status' => 'required|in:1,0',
        'merchant_key' => 'required',
        'merchant_id' => 'required',
        'merchant_website_link' => 'required'
    ];
} elseif ($request['gateway'] == 'bkash') {
    $additional_data = [
        'status' => 'required|in:1,0',
        'app_key' => 'required',
        'app_secret' => 'required',
        'username' => 'required',
        'password' => 'required',
    ];
} elseif ($request['gateway'] == 'cash_after_service') {
    $additional_data = [
        'status' => 'required|in:1,0'
    ];
} elseif ($request['gateway'] == 'digital_payment') {
    $additional_data = [
        'status' => 'required|in:1,0'
    ];
} elseif ($request['gateway'] == 'momo') {
    $additional_data = [
        'status' => 'required|in:1,0',
        'api_key' => 'required',
        'api_user' => 'required',
        'subscription_key' => 'required',
    ];
} elseif ($request['gateway'] == 'hyper_pay') {
    $additional_data = [
        'status' => 'required|in:1,0',
        'entity_id' => 'required',
        'access_code' => 'required',
    ];
} elseif ($request['gateway'] == 'amazon_pay') {
    $additional_data = [
        'status' => 'required|in:1,0',
        'pass_phrase' => 'required',
        'access_code' => 'required',
        'merchant_identifier' => 'required',
    ];
};
```

```
    } elseif ($request['gateway'] == 'sixcash') {
        $additional_data = [
            'status' => 'required|in:1,0',
            'public_key' => 'required',
            'secret_key' => 'required',
            'merchant_number' => 'required',
            'base_url' => 'required',
        ];
    } elseif ($request['gateway'] == 'worldpay') {
        $additional_data = [
            'status' => 'required|in:1,0',
            'OrgUnitId' => 'required',
            'jwt_issuer' => 'required',
            'mac' => 'required',
            'merchantCode' => 'required',
            'xml_password' => 'required',
        ];
    } elseif ($request['gateway'] == 'payfast') {
        $additional_data = [
            'status' => 'required|in:1,0',
            'merchant_id' => 'required',
            'secured_key' => 'required',
        ];
    } elseif ($request['gateway'] == 'swish') {
        $additional_data = [
            'status' => 'required|in:1,0',
            'number' => 'required',
        ];
    } elseif ($request['gateway'] == 'esewa') {
        $additional_data = [
            'status' => 'required|in:1,0',
            'merchantCode' => 'required',
        ];
    } elseif ($request['gateway'] == 'maxicash') {
        $additional_data = [
            'status' => 'required|in:1,0',
            'merchantId' => 'required',
            'merchantPassword' => 'required',
        ];
    } elseif ($request['gateway'] == 'hubtel') {
        $additional_data = [
            'status' => 'required|in:1,0',
            'account_number' => 'required',
            'api_id' => 'required',
            'api_key' => 'required',
        ];
    }
```

```
];
} elseif ($request['gateway'] == 'viva_wallet') {
    $additional_data = [
        'status' => 'required|in:1,0',
        'client_id' => 'required',
        'client_secret' => 'required',
        'source_code' => 'required',
    ];
} elseif ($request['gateway'] == 'tap') {
    $additional_data = [
        'status' => 'required|in:1,0',
        'secret_key' => 'required',
    ];
} elseif ($request['gateway'] == 'thawani') {
    $additional_data = [
        'status' => 'required|in:1,0',
        'public_key' => 'required',
        'private_key' => 'required',
    ];
} elseif ($request['gateway'] == 'moncash') {
    $additional_data = [
        'status' => 'required|in:1,0',
        'client_id' => 'required',
        'secret_key' => 'required',
    ];
} elseif ($request['gateway'] == 'pvit') {
    $additional_data = [
        'status' => 'required|in:1,0',
        'mc_tel_merchant' => 'required',
        'access_token' => 'required',
        'mc_merchant_code' => 'required',
    ];
} elseif ($request['gateway'] == 'ccavenue') {
    $additional_data = [
        'status' => 'required|in:1,0',
        'merchant_id' => 'required',
        'working_key' => 'required',
        'access_code' => 'required',
    ];
} elseif ($request['gateway'] == 'foloosi') {
    $additional_data = [
        'status' => 'required|in:1,0',
        'merchant_key' => 'required',
    ];
} elseif ($request['gateway'] == 'iyzi_pay') {
```

```

        $additional_data = [
            'status' => 'required|in:1,0',
            'api_key' => 'required',
            'secret_key' => 'required',
            'base_url' => 'required',
        ];
    } elseif ($request['gateway'] == 'xendit') {
        $additional_data = [
            'status' => 'required|in:1,0',
            'api_key' => 'required',
        ];
    } elseif ($request['gateway'] == 'fatoorah') {
        $additional_data = [
            'status' => 'required|in:1,0',
            'api_key' => 'required',
        ];
    }
}

$request->validate(array_merge($validation,
$additional_data));

$settings = Setting::where('key_name', $request['gateway'])->
where('settings_type', 'payment_config')->first();

$additional_data_image = $settings['additional_data'] !=
null ? json_decode($settings['additional_data']) : null;

if( !$additional_data_image ||
!isset($additional_data_image->gateway_image) ||
(isset($additional_data_image->gateway_image) &&
$additional_data_image->gateway_image == '') ||
(isset($additional_data_image->gateway_image) &&
!file_exists(base_path("storage/app/public/payment_modules/gateway_i
mage/". $additional_data_image->gateway_image)))){
    $request->validate([
        'gateway_image' => 'required',
    ]);
}
$request->validate(['gateway_title' => 'required']);

if ($request->has('gateway_image')) {
    $gateway_image = $this->file_uploader('payment_modules/gateway_image/', 'png',
$request['gateway_image'], $additional_data_image != null ?
$additional_data_image->gateway_image : '');
}

```

```

        } else {
            $gateway_image = $additional_data_image != null ?
$additional_data_image->gateway_image : '';
        }

        $payment_additional_data = [
            'gateway_title' => $request['gateway_title'],
            'gateway_image' => $gateway_image,
        ];

        $validator = Validator::make($request->all(),
array_merge($validation, $additional_data));

        Setting::updateOrCreate(['key_name' => $request['gateway'],
'settings_type' => 'payment_config'], [
            'key_name' => $request['gateway'],
            'live_values' => $validator->validate(),
            'test_values' => $validator->validate(),
            'settings_type' => 'payment_config',
            'mode' => $request['mode'],
            'is_active' => $request['status'] ?? 0,
            'additional_data' =>
json_encode($payment_additional_data),
        ]);

        Toastr::success(GATEWAYS_DEFAULT_UPDATE_200['message']);
        return back();
    }
}

```

7 Transaksi Controller

```

<?php

namespace App\Http\Controllers\Admin;

use App\Utils\Helpers;
use App\Http\Controllers\Controller;
use App\Models\OrderTransaction;
use App\User;
use Illuminate\Http\Request;
use Rap2hpoutre\FastExcel\FastExcel;

```

```

class TransactionController extends Controller
{
    public function list(Request $request)
    {
        $query_param = [];
        $search       = $request['search'];
        $from         = $request['from'];
        $to           = $request['to'];
        $customer_id  = $request['customer_id'];
        $status       = $request['status'];

        $customers = User::whereNotIn('id',[0])->get();

        $transactions =
OrderTransaction::with(['seller','customer'])
->when($search, function($q) use($search){
    $q->orWhere('order_id', 'like',
"%{$search}%" )
->orWhere('transaction_id', 'like',
"%{$search}%" );
    })
->when($customer_id, function($q)
use($customer_id){
    $q->where('customer_id', $customer_id);
    })
->when($status == 'all', function ($q)
use($status){
    $q;
    })
->when(!empty($status) && ($status !=
'all'), function ($q) use($status){
    $q->where('status', 'like',
"%{$status}%" );
    })
->when(!empty($from) &&
!empty($to),function($query) use($from,$to){
    $query->whereBetween('created_at',
[$from . ' 00:00:00', $to . ' 23:59:59']);
    })
->latest()-
>paginate(Helpers::pagination_limit())->appends([
    'customer_id'=>$customer_id,
    'status'=>$status,
    'from'=>$from,
    'to'=>$to,

```

```

        'search'=>$search]);

        return view('admin-views.transaction.list',
compact('customers', 'transactions','search','status', 'from', 'to',
'customer_id'));
    }

    /**
     * Transaction report export by excel
     * @param Request $request
     * @return
string|\Symfony\Component\HttpFoundation\StreamedResponse
     * @throws \Box\Spout\Common\Exception\IOException
     * @throws \Box\Spout\Common\Exception\InvalidArgumentException
     * @throws \Box\Spout\Common\Exception\UnsupportedTypeException
     * @throws \Box\Spout\Writer\Exception\WriterNotOpenedException
     */
    public function export(Request $request):
\Symfony\Component\HttpFoundation\StreamedResponse|string
    {
        $from      = $request['from'];
        $to        = $request['to'];
        $customer_id = $request['customer_id'];
        $status    = $request['status'];

        $transactions =
OrderTransaction::with(['seller','customer'])
        ->when($customer_id, function($q) use($customer_id){
            $q->where('customer_id', $customer_id);
        })
        ->when($status == 'all', function ($q) use($status){
            $q;
        })
        ->when(!empty($status) && ($status != 'all'), function
($q) use($status){
            $q->where('status', 'like', "%{$status}%");
        })
        ->when(!empty($from) && !empty($to),function($query)
use($from,$to){
            $query->whereBetween('created_at', [$from . '
00:00:00', $to . ' 23:59:59']);
        })
        ->latest()->get();
    }

```

```

        $tranData = array();
        foreach($transactions as $tran){
            if($tran['seller_is'] == 'admin'){
                $seller_name =
\App\Utils\Helpers::get_business_settings('company_name');
            }else{
                $seller_name = $tran->seller ? $tran->seller->f_name
.' '. $tran->seller->l_name : translate('not_found');
            }

            $tranData[] = array(
                'Seller Name' => $seller_name,
                'Customer Name' => $tran->customer ? $tran-
>customer->f_name.' '.$tran->customer->l_name :
translate('not_found'),
                'Order ID' => $tran->order_id,
                'Transaction ID' => $tran->transaction_id,
                'Order Amount' =>
\App\Utils\BackEndHelper::set_symbol(\App\Utils\BackEndHelper::usd_t
o_currency($tran->order_amount)),
                'Seller Amount' =>
\App\Utils\BackEndHelper::set_symbol(\App\Utils\BackEndHelper::usd_t
o_currency($tran->seller_amount)),
                'Admin Commission' =>
\App\Utils\BackEndHelper::set_symbol(\App\Utils\BackEndHelper::usd_t
o_currency($tran->admin_commission)),
                'Received By' => $tran->received_by,
                'Delivered By' => $tran->delivered_by,
                'Delivery Charge' => $tran->delivery_charge,
                'Payment Method' => $tran->payment_method,
                'Tax' => $tran->tax,
                'Date' => date('d M Y',strtotime($tran-
>created_at)),
                'Status' => $tran->status,
            );
        }

        return (new FastExcel($tranData))-
>download('Transaction_All_details.xlsx');
    }
}

```


8 Login Controller

```
<?php

namespace App\Http\Controllers;

use App\Utils\Helpers;
use App\Models\Admin;
use App\Models\BusinessSetting;
use Brian2694\Toastr\Facades\Toastr;
use Gregwar\Captcha\CaptchaBuilder;
use Gregwar\Captcha\PhraseBuilder;
use Illuminate\Http\Request;
use Illuminate\Support\Facades\Http;
use Illuminate\Support\Facades\Session;

class LoginController extends Controller
{
    public function __construct()
    {
        $this->middleware('guest:admin', ['except' => 'logout']);
    }

    public function login($login_url)
    {
        $data =
array_column(BusinessSetting::whereIn('type',['employee_login_url','
admin_login_url'])->get(['type','value'])->toArray(), 'value',
'type');

        $loginTypes = [
            'admin' => 'admin_login_url',
            'employee' => 'employee_login_url'
        ];

        $role = null;

        $user_type = array_search($login_url,$data);
        abort_if(!$user_type, 404 );
        $role = array_search($user_type,$loginTypes,true);
        abort_if($role == null ,404);

        $custome_recaptcha = new CaptchaBuilder;
        $custome_recaptcha->build();
    }
}
```

```

        Session::put('six_captcha', $custome_recaptcha-
>getPhrase());

        if ($role == 'admin') {
            return view('admin-views.auth.login',
compact('custome_recaptcha', 'role'));
        } else if ($role == 'employee') {
            return view('admin-views.auth.login',
compact('custome_recaptcha', 'role'));
        }
    }

    public function submit(Request $request)
    {
        $request->validate([
            'email' => 'required|email',
            'password' => 'required|min:6',
            'role' => 'required'
        ]);

        $recaptcha = Helpers::get_business_settings('recaptcha');
        if (isset($recaptcha) && $recaptcha['status'] == 1) {
            $request->validate([
                'g-recaptcha-response' => [
                    function ($attribute, $value, $fail) {
                        $secret_key =
Helpers::get_business_settings('recaptcha')['secret_key'];
                        $response = $value;
                        $url =
'https://www.google.com/recaptcha/api/siteverify?secret=' .
$secret_key . '&response=' . $response;
                        $response = Http::get($url);
                        $response = $response->json();
                        if (!isset($response['success']) ||
!$response['success']) {
                            $fail(translate('ReCAPTCHA_Failed'));
                        }
                    },
                ],
            ]);
        } else
        if(strtolower(session('default_recaptcha_id_'. $request-
>role.'_login')) != strtolower($request->default_captcha_value))
        {

```

```

        Toastr::error(translate('ReCAPTCHA_Failed'));
        return back();
    }

    if ($request->role == 'admin') {
        $data = Admin::where('email', $request->email)-
>where('admin_role_id', 1)->first();

        if (!isset($data)) {
            return redirect()->back()->withInput($request-
>only('email', 'remember'))
                ->withErrors(['Credentials does not
match.']);
        } else if (isset($data) && $data->status != 1) {
            return redirect()->back()->withInput($request-
>only('email', 'remember'))
                ->withErrors(['You are blocked!!, contact with
admin.']);
        }
    } elseif ($request->role == 'employee') {

        $data = Admin::where('email', $request->email)-
>where('admin_role_id', '!=', 1)->first();

        if (!isset($data)) {
            return redirect()->back()->withInput($request-
>only('email', 'remember'))
                ->withErrors(['Credentials does not
match.']);
        } else if (isset($data) && $data->status != 1) {
            return redirect()->back()->withInput($request-
>only('email', 'remember'))
                ->withErrors(['You are blocked!!, contact with
admin.']);
        }
    } else {
        Toastr::error(translate('role_missing'));
        return back();
    }

    $data = $this->login_attemp($request->role, $request->email,
$request->password, $request->remember);

    if ($data == 'admin' || $data == 'employee') {

```

```
        return redirect()->route('admin.dashboard.index');
    }

    return redirect()->back()->withInput($request->only('email',
'remember'))
        ->withErrors(['Credentials does not match.']);
    }

    public function login_attemp($role,$email ,$password, $remember
= false)
    {
        if ($role == 'admin' || $role == 'employee') {
            if (auth('admin')->attempt(['email' => $email,
'password' => $password], $remember)) {
                return $role;
            }
        }
        return false;
    }

    public function captcha(Request $request, $tmp)
    {
        $phrase = new PhraseBuilder;
        $code = $phrase->build(4);
        $builder = new CaptchaBuilder($code, $phrase);
        $builder->setBackgroundColor(220, 210, 230);
        $builder->setMaxAngle(25);
        $builder->setMaxBehindLines(0);
        $builder->setMaxFrontLines(0);
        $builder->build($width = 100, $height = 40, $font = null);
        $phrase = $builder->getPhrase();

        if(Session::has($request->captcha_session_id)) {
            Session::forget($request->captcha_session_id);
        }
        Session::put($request->captcha_session_id, $phrase);
        header("Cache-Control: no-cache, must-revalidate");
        header("Content-Type:image/jpeg");
        $builder->output();
    }
}
```

