

**IMPLEMENTASI *FULLSTACK WEB DEVELOPMENT* PADA
PEMBUATAN APLIKASI *WEBSTORE*
(STUDI KASUS : PROYEK MBKM ARKATAMA)**

(Tugas Akhir)

Oleh

**ANDRE AGUNG
2007051035**



**FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM
UNIVERSITAS LAMPUNG
BANDAR LAMPUNG
2023**

ABSTRAK

IMPLEMENTASI FULLSTACK WEB DEVELOPMENT PADA PEMBUATAN APLIKASI WEBSTORE (STUDI KASUS : PROYEK MBKM ARKATAMA)

Oleh:

ANDRE AGUNG

Teknologi informasi sangatlah berpengaruh untuk kemajuan suatu usaha. Berbagai macam cara dilakukan untuk mempromosikan usaha yang dimiliki, salah satunya dengan menggunakan website *online store*. *Online Store* adalah suatu kegiatan jual beli barang yang dilakukan secara *online* melalui situs atau *website*. Pada studi kasus ini *website* yang dibangun setidaknya mempunyai standar *online store* pada umumnya.

Misi dari laporan ini adalah memberi solusi pemecahan masalah-masalah yang ada dengan merancang sebuah *online store* berbasis *website*. Untuk memudahkan hal ini, maka dibangun sebuah *arkatama store* dengan sistem pengelolaan seperti pengelolaan produk, transaksi, dan pengiriman. Selain itu, juga dilakukan integrasi dengan RajaOngkir, sebuah layanan penyedia informasi mengenai jasa ekspedisi dan ongkos kirim untuk sistem penghitungan ongkos kirim otomatis.

Hasil dan kesimpulan dari laporan ini adalah *arkatama store* berbasis *website* dapat dijadikan sebagai media promosi, dapat memproses penjualan yang dilakukan secara *online*, serta dapat mempermudah dalam pengolahan data dan pencetakan laporan.

Kata Kunci : *Online Store*, *Website*, RajaOngkir, dan Penjualan.

**IMPLEMENTASI *FULLSTACK WEB DEVELOPMENT* PADA
PEMBUATAN APLIKASI *WEBSTORE***

(STUDI KASUS : PROYEK MBKM ARKATAMA)

Oleh

ANDRE AGUNG

Tugas Akhir

**Sebagai Salah Satu Syarat Mencapai Gelar
AHLI MADYA MANAJEMEN INFORMATIKA**

Pada

**Jurusan Ilmu Komputer
Fakultas Matematika Dan Ilmu Pengetahuan Alam**



**FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM
UNIVERSITAS LAMPUNG
BANDAR LAMPUNG
2023**

Judul Tugas Akhir : **Implementasi *Fullstack Web Development***
Pada Pembuatan Aplikasi *Webstore*
(Studi Kasus : Proyek MBKM Arkatama)

Nama Mahasiswa : Andre Agung

Nomor Pokok Mahasiswa : 2007051035

Program Studi : DIII Manajemen Informatika

Jurusan : Ilmu Komputer

Fakultas : Matematika dan Ilmu Pengetahuan Alam



Pembimbing Utama

Pembimbing Kedua

Anie Rose Irawati, S.T., M.Cs.
NIP. 19791031 200604 2 002

Yohana Tri Utami, S.Kom., M.Kom.
NIP. 19900110 201903 2 010

2. Mengetahui,

Ketua Jurusan Ilmu Komputer

Ketua Program Studi
DIII Manajemen Informatika

Didik Kurniawan, S.Si., M.T.
NIP. 19800419 200501 1 001

Anie Rose Irawati, S.T., M.Cs.
NIP. 19791031 200604 2 002

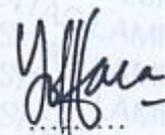
MENGESAHKAN

1. Tim Penguji

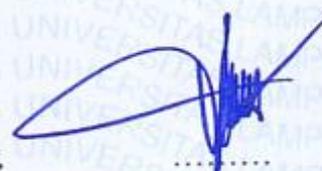
Pembimbing Utama : **Anie Rose Irawati, S.T., M.Cs.**



Pembimbing Kedua : **Yohana Tri Utami, S.Kom., M.Kom.**



Penguji/Pembahas : **Didik Kurniawan, S.Si., M.T.**

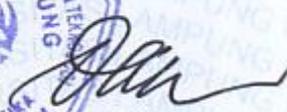


2. Dekan Fakultas Matematika dan Ilmu Pengetahuan Alam



Dr. Eng. Suripto Dwi Yuwono, S.Si., M.T.

NIP. 19740705 200003 1 001



Tanggal Lulus Ujian Tugas Akhir : **09 Februari 2023**

PERNYATAAN MENGENAI TUGAS AKHIR DAN SUMBER INFORMASI

Dengan ini saya menyatakan bahwa Tugas Akhir yang berjudul **Implementasi Fullstack Web Development Pada Pembuatan Aplikasi Webstore (Studi Kasus : Proyek MBKM Arkatama)** adalah karya saya dengan arahan komisi pembimbing dan belum diajukan dalam bentuk apapun kepada perguruan tinggi manapun. Sumber informasi yang berasal atau dikutip dari karya yang diterbitkan maupun tidak diterbitkan dari penulis lain telah disebutkan dalam teks dan dicantumkan dalam daftar pustaka di bagian Tugas Akhir ini.

Bandar Lampung, 09 Februari 2023



Andre Agung
NPM : 2007051035

Hak Cipta Milik UNILA, Tahun 2023
Hak Cipta dilindungi Undang-Undang

Dilarang mengutip sebagian atau seluruh Karya Tulis ini tanpa mencantumkan atau menyebutkan sumbernya. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penyusunan laporan, penulisan kritik, atau tinjauan suatu masalah dan pengutipan tersebut tidak merugikan kepentingan yang wajar UNILA.

Dilarang mengumumkan dan memperbanyak sebagian atau seluruh Karya Tulis dalam bentuk apapun tanpa izin UNILA.

RIWAYAT HIDUP

Penulis dilahirkan di Kotabumi, Lampung Utara pada tanggal 8 Juli 2002, sebagai anak ketiga dari empat bersaudara, dari Bapak Nizar Agung dan Ibu Maryana.

Pendidikan Taman Kanak-kanak (TK) Kemala Kotabumi diselesaikan tahun 2008, Sekolah Dasar (SD) diselesaikan di SDN 6 Tanjung Aman, Kotabumi pada tahun 2014, Sekolah Menengah Pertama (SMP) di MTSN 1 Kotabumi pada tahun 2017, dan Sekolah Menengah Atas (SMA) di SMAN 1 Kotabumi pada tahun 2020.

Tahun 2020, penulis terdaftar sebagai mahasiswa Jurusan Ilmu Komputer, Program Studi DIII Manajemen Informatika, FMIPA, Unila melalui jalur proses seleksi SIMANILA Program Diploma 3. Pada tahun 2022, penulis melakukan kerja praktik di PT Jasa Raharja Cabang Lampung dan ikut serta pada kegiatan MSIB dari program Merdeka Belajar Kampus Merdeka (MBKM) yang diselenggarakan oleh Kemendikbudristek.

MOTTO

“Kegagalan tidak akan pernah mengalahkan keinginan kuat untuk mencapai kesuksesan”

“Cara terbaik untuk memulai adalah dengan berhenti berbicara dan mulai melakukan”

“Satu-satunya cara untuk melakukan pekerjaan hebat adalah dengan mencintai apa yang kita lakukan”

“Usaha keras selalu tidak akan mengkhianati hasil”

“Tidak peduli apa yang terjadi dalam hidup, jadilah orang yang baik. Menjadi baik kepada orang lain adalah warisan yang indah untuk ditinggalkan”

PERSEMBAHAN

Dengan segala rasa syukur kepada Allah SWT yang mendalam, dan dengan telah diselesaikannya tugas akhir ini penulis mempersembahkannya kepada:

1. Bapak dan Ibu tersayang yang selalu mengingatkan dan memberikan semangat, doa, serta segala dukungan dengan sangat tulus.
2. Seluruh keluarga besar penulis yang telah banyak memberikan motivasi dan bantuan dalam menyelesaikan tugas akhir.
3. Sahabat-sahabat yang selalu memberikan semangat, nasihat dan dukungan.
4. Seluruh teman-teman seangkatan di jurusan ilmu komputer.

KATA PENGANTAR

Puji syukur saya panjatkan kehadirat Allah SWT yang telah melimpahkan rahmat, hidayah dan inayah-Nya sehingga penyusunan Laporan Tugas Akhir dapat diselesaikan dengan baik dan tepat pada waktu yang telah ditentukan. Sholawat serta salam tak lupa kita haturkan kepada junjungan Nabi Besar Muhammad SAW dan para sahabatnya, yang telah memberikan tauladan baik sehingga akal dan pikiran penulis mampu menyelesaikan Laporan Tugas Akhir ini sebagai syarat mencapai gelar ahli madya manajemen informatika.

Laporan yang berjudul “Implementasi *Fullstack Web Development* Pada Pembuatan Aplikasi *Webstore* (Studi Kasus : Proyek MBKM Arkatama)” yang merupakan hasil dari pengerjaan tugas akhir yang telah diselesaikan.

Laporan Tugas Akhir disusun dengan maksimal dan memperoleh bantuan dari berbagai pihak sehingga dapat memudahkan dalam pembuatan laporan. Maka dari itu, Pada kesempatan kali ini izinkan penulis untuk mengucapkan terimakasih dan rasa hormat atas segala bantuan yang telah diberikan sehingga dapat menyelesaikan Laporan Tugas Akhir, yaitu kepada:

1. Allah SWT yang telah memberikan rahmat, karunia, serta hidayah-Nya selama proses pengerjaan tugas akhir hingga laporan ini dapat tersusun dengan baik.
2. Kedua Orang Tua yang telah mendoakan dan memberikan semangat yang luar biasa sehingga dapat menyelesaikan laporan tugas akhir.

3. Bapak Didik Kurniawan, S.Si., M.T., selaku Ketua Jurusan Ilmu Komputer Universitas Lampung dan Dosen Penguji Tugas Akhir yang telah memberikan arahan dan saran terhadap laporan maupun aplikasi yang diuji.
4. Ibu Anie Rose Irawati S.T., M.Cs. selaku Ketua Program Studi DIII Manajemen Informatika dan Dosen Pembimbing I yang telah banyak memberikan bimbingan, dan saran.
5. Yohana Tri Utami, S.Kom., M.Kom. selaku Dosen Pembimbing II atas bimbingan, dan masukan yang diberikan.
6. Semua teman-teman yang telah membantu ketika mendapatkan kesulitan dalam pengerjaan Tugas Akhir.

Bandar Lampung, 09 Februari 2023

Penulis,

Andre Agung

NPM : 2007051035

DAFTAR ISI

	Halaman
DAFTAR ISI.....	x
DAFTAR TABEL	xiii
DAFTAR GAMBAR	xiv
I.PENDAHULUAN.....	1
1.1. Latar Belakang	1
1.2. Rumusan Masalah	2
1.3. Batasan Masalah.....	2
1.4. Tujuan.....	3
1.5. Manfaat.....	3
II.TINJAUAN PUSTAKA.....	4
2.1. Aplikasi	4
2.2. Website	4
2.3. Online Store.....	4
2.4. HTML	4
2.5. CSS	5
2.6. Java Script.....	5
2.7. PHP	5
2.8. Database.....	5
2.9. Bootstrap	6
2.10. Codeigniter	6
2.11. XAMMP	6
2.12. Web Server.....	7
2.13. Use Case Diagram	7
2.14. Activity Diagram	7
2.15. ERD.....	8

2.16.	<i>Framework</i>	8
2.17.	RajaOngkir <i>API</i>	9
2.18.	Metode <i>Waterfall</i>	9
III.ANALISIS DAN PERANCANGAN		10
3.1.	Analisis Kebutuhan Aplikasi.....	10
3.1.1.	Analisis Kebutuhan Fungsional	10
3.1.2.	Analisis Kebutuhan Non Fungsional	11
3.2.	Perancangan Aplikasi	12
3.2.1.	Desain Aplikasi	13
3.2.2.	Desain Proses	14
3.2.3.	Desain Data	34
3.2.4.	Desain Antarmuka.....	39
IV.HASIL DAN PEMBAHASAN.....		66
4.1.	Hasil Aplikasi	66
4.1.1.	Tampilan <i>Login</i>	66
4.1.2.	Tampilan Registrasi	67
4.1.3.	Tampilan <i>Home - Customer</i>	68
4.1.4.	Tampilan Halaman Produk - <i>Customer</i>	68
4.1.5.	Tampilan <i>Detail</i> Produk - <i>Customer</i>	69
4.1.6.	Tampilan Kelola Keranjang - <i>Customer</i>	70
4.1.7.	Tampilan <i>Checkout - Customer</i>	71
4.1.8.	Tampilan Kelola Pesanan ‘Diorder’ - <i>Customer</i>	72
4.1.9.	Tampilan Kelola Pesanan ‘Diorder’ - Admin	73
4.1.10.	Tampilan <i>Detail</i> Pesanan - Admin.....	75
4.1.11.	Tampilan Kelola Pesanan ‘Diproses’ - <i>Customer</i>	76
4.1.12.	Tampilan Kelola Pesanan ‘Diproses’ - Admin	76
4.1.13.	Tampilan Kelola Pesanan ‘Dikirim’ - Admin.....	78
4.1.14.	Tampilan Kelola Pesanan ‘Dikirim’ - <i>Customer</i>	78
4.1.15.	Tampilan Kelola Pesanan ‘Selesai’ - <i>Customer</i>	80
4.1.16.	Tampilan Kelola Pesanan ‘Selesai - Admin	81
4.1.17.	Tampilan Kelola Data Barang - Admin	81
4.1.18.	Tampilan <i>Detail</i> Barang - Admin	83

4.1.19.	Tampilan Kelola Data Kategori - Admin.....	84
4.1.20.	Tampilan Kelola <i>Hero</i> - Admin	85
4.1.21.	Tampilan Data Testimoni - Admin	87
4.1.22.	Tampilan Cetak Laporan Penjualan - Admin dan <i>Manager</i>	88
4.1.23.	Tampilan Data Barang - <i>Manager</i>	89
4.1.24.	Tampilan Data <i>Hero</i> - <i>Manager</i>	90
4.2.	Pengujian Aplikasi	90
4.2.1.	Pengujian bagi <i>customer</i>	91
4.2.2.	Pengujian bagi admin	91
4.2.3.	Pengujian bagi <i>manager</i>	92
4.3.	Hasil Pengujian Aplikasi	93
V.	SIMPULAN DAN SARAN	94
5.1.	Simpulan.....	94
5.2.	Saran	94
	DAFTAR PUSTAKA	96
	LAMPIRAN.....	98

DAFTAR TABEL

	Halaman
Tabel 1. Simbol-simbol <i>use case</i> diagram.	7
Tabel 2. Simbol-simbol <i>activity</i> diagram.	8
Tabel 3. <i>User</i>	35
Tabel 4. <i>Role</i>	35
Tabel 5. Barang	35
Tabel 6. Kategori.....	36
Tabel 7. <i>Hero</i>	36
Tabel 8. Keranjang	37
Tabel 9. Penjualan.....	37
Tabel 10. <i>Detail</i> Penjualan.....	38
Tabel 11. Testimoni	38
Tabel 12. Pengujian pengguna <i>customer</i>	99
Tabel 13. Pengujian pengguna admin	104
Tabel 14. Pengujian pengguna <i>manager</i>	112

DAFTAR GAMBAR

	Halaman
Gambar 1. Metode <i>Waterfall</i> (Rosa & Shalahuddin, 2015).....	9
Gambar 2. <i>Use Case</i> Diagram <i>Arkatama Store</i>	13
Gambar 3. <i>Activity</i> diagram <i>login</i>	14
Gambar 4. <i>Activity</i> diagram registrasi - <i>customer</i>	15
Gambar 5. <i>Activity</i> diagram lihat produk - <i>customer</i>	16
Gambar 6. <i>Activity</i> diagram kelola keranjang - <i>customer</i>	17
Gambar 7. <i>Activity</i> diagram <i>checkout</i> - <i>customer</i>	18
Gambar 8. <i>Activity</i> diagram kelola pesanan - <i>customer</i>	20
Gambar 9. <i>Activity</i> Diagram kelola data produk - admin.....	22
Gambar 10. <i>Activity</i> diagram kelola data kategori - admin.....	24
Gambar 11. <i>Activity</i> diagram kelola <i>hero</i> - admin	26
Gambar 12. <i>Activity</i> diagram kelola data pesanan - admin	28
Gambar 13. <i>Activity</i> diagram hapus testimoni - admin	30
Gambar 14. <i>Activity</i> diagram cetak laporan penjualan - admin dan <i>manager</i>	31
Gambar 15. <i>Activity</i> diagram kelola status produk - <i>manager</i>	32
Gambar 16. <i>Activity</i> diagram kelola status <i>hero</i> - <i>manager</i>	33
Gambar 17. <i>Entity Relationship</i> Diagram (<i>ERD</i>) <i>Arkatama Store</i>	34
Gambar 18. Desain <i>Interface Login</i>	39
Gambar 19. Desain <i>Interface Registrasi</i>	40
Gambar 20. Desain <i>interface home</i> - <i>customer</i>	41
Gambar 21. Desain <i>interface</i> menu produk - <i>customer</i>	42
Gambar 22. Desain <i>interface</i> detail produk - <i>customer</i>	43
Gambar 23. Desain <i>interface</i> keranjang - <i>customer</i>	44
Gambar 24. Desain <i>interface checkout</i> - <i>customer</i>	45
Gambar 25. Desain <i>interface</i> pesanan ‘diorder’ - <i>customer</i>	46

Gambar 26. Desain <i>interface</i> pembayaran pesanan - <i>customer</i>	46
Gambar 27. Desain <i>interface</i> kelola pesanan ‘masuk’ - admin	47
Gambar 28. Desain <i>interface</i> bukti pembayaran - admin	48
Gambar 29. Desain <i>interface detail</i> pesanan - admin	49
Gambar 30. Desain <i>interface</i> pesanan ‘diproses’ - <i>customer</i>	49
Gambar 31. Desain <i>interface</i> kelola pesanan ‘diproses’ - admin.....	50
Gambar 32. Desain <i>interface form</i> no resi - admin.....	51
Gambar 33. Desain <i>interface</i> kelola pesanan ‘dikirim’ - admin	51
Gambar 34. Desain <i>interface</i> pesanan ‘dikirim’ - <i>customer</i>	52
Gambar 35. Desain <i>interface form</i> testimoni - <i>customer</i>	53
Gambar 36. Desain <i>interface</i> halaman testimoni - <i>customer</i>	53
Gambar 37. Desain <i>interface</i> pesanan ‘selesai’ - <i>customer</i>	54
Gambar 38. Desain <i>interface</i> kelola pesanan ‘selesai’ - admin	54
Gambar 39. Desain <i>interface</i> kelola data barang - admin.....	55
Gambar 40. Desain <i>interface</i> tambah data barang - admin.....	56
Gambar 41. Desain <i>interface</i> edit data barang - admin.....	57
Gambar 42. Desain <i>interface detail</i> data barang - admin.....	57
Gambar 43. Desain <i>interface</i> kelola data kategori - admin.....	58
Gambar 44. Desain <i>interface</i> tambah data kategori - admin.....	59
Gambar 45. Desain <i>interface</i> edit data kategori - admin	59
Gambar 46. Desain <i>interface setting hero</i> - admin	60
Gambar 47. Desain <i>interface</i> tambah <i>hero</i> - admin	61
Gambar 48. Desain <i>interface</i> edit <i>hero</i> - admin.....	61
Gambar 49. Desain <i>interface</i> testimoni - admin	62
Gambar 50. Desain <i>interface</i> cetak laporan penjualan – admin dan <i>manager</i>	63
Gambar 51. Desain <i>interface</i> data barang - <i>manager</i>	64
Gambar 52. Desain <i>interface</i> data <i>hero</i> - <i>manager</i>	65
Gambar 53. Tampilan halaman <i>login</i>	66
Gambar 54. Tampilan halaman registrasi - <i>customer</i>	67
Gambar 55. Tampilan halaman <i>home</i> - <i>customer</i>	68
Gambar 56. Tampilan halaman menu produk - <i>customer</i>	68
Gambar 57. Tampilan halaman <i>detail</i> produk - <i>customer</i>	69

Gambar 58. Tampilan halaman keranjang - <i>customer</i>	70
Gambar 59. Tampilan halaman <i>checkout</i> - <i>customer</i>	71
Gambar 60. Tampilan halaman pesanan ‘diorder’ - <i>customer</i>	72
Gambar 61. Tampilan halaman pembayaran pesanan - <i>customer</i>	73
Gambar 62. Tampilan halaman kelola pesanan ‘diorder’ - admin.....	73
Gambar 63. Tampilan halaman bukti pembayaran - admin.....	74
Gambar 64. Tampilan halaman <i>detail</i> pesanan - admin.....	75
Gambar 65. Tampilan halaman pesanan ‘diproses’ - <i>customer</i>	76
Gambar 66. Tampilan halaman kelola pesanan ‘diproses’ - admin.....	76
Gambar 67. Tampilan halaman <i>form</i> no resi - admin	77
Gambar 68. Tampilan halaman kelola pesanan ‘dikirim’ - admin.....	78
Gambar 69. Tampilan halaman pesanan ‘dikirim’ - <i>customer</i>	78
Gambar 70. Tampilan halaman <i>form</i> testimoni - <i>customer</i>	79
Gambar 71. Tampilan halaman testimoni - <i>customer</i>	79
Gambar 72. Tampilan halaman pesanan ‘selesai’ - <i>customer</i>	80
Gambar 73. Tampilan halaman kelola pesanan ‘selesai’ - admin.....	81
Gambar 74. Tampilan halaman kelola data barang - admin	81
Gambar 75. Tampilan halaman tambah data barang - admin	82
Gambar 76. Tampilan halaman edit data barang - admin	83
Gambar 77. Tampilan halaman <i>detail</i> data barang - admin	83
Gambar 78. Tampilan halaman kelola data kategori - admin	84
Gambar 79. Tampilan halaman tambah data kategori - admin	84
Gambar 80. Tampilan halaman edit data kategori - admin.....	85
Gambar 81. Tampilan halaman kelola data <i>hero</i> - admin.....	85
Gambar 82. Tampilan halaman tambah data <i>hero</i> - admin.....	86
Gambar 83. Tampilan halaman edit data <i>hero</i> - admin.....	87
Gambar 84. Tampilan halaman data testimoni - admin	87
Gambar 85. Tampilan halaman cetak laporan penjualan - admin.....	88
Gambar 86. Tampilan halaman data barang - <i>manager</i>	89
Gambar 87. Tampilan halaman data <i>hero</i> - <i>manager</i>	90

I. PENDAHULUAN

1.1. Latar Belakang

PT. Arkatama Multi Solusindo adalah perusahaan yang kompeten dalam menangani berbagai pekerjaan di bidang teknologi informasi khususnya pengembangan aplikasi *custom*, pengadaan dan konfigurasi infrastruktur *hardware*, optimasi dan automasi proses bisnis, serta jasa pelatihan *skill* dibidang teknologi informasi yang dikhususkan untuk para *developer*, *engineer*, *user*, maupun *manager* pada proyek teknologi informasi.

Penulis ikut serta pada program Studi Independen Bersertifikat dari Kampus Merdeka dengan skema *Fullstack Web Development* yang bertujuan untuk melatih mahasiswa menjadi *web developer* yang kompeten dan profesional. Pada akhir program, Peserta diminta untuk membuat *website project* akhir berdasarkan *study case* yang diberikan oleh instansi.

Adapun *study case* yang diberikan PT Arkatama Multi Solusindo yaitu dengan tema *online store*. Peserta diminta untuk membuat aplikasi *online store* berbasis *web* yang menampilkan suatu produk tertentu dan aplikasi yang dibuat dapat digunakan untuk melakukan jual beli produk antara *customer* dan penjual.

Aplikasi *online store* yang dibuat memiliki fitur pengelolaan penjualan seperti menampilkan menu halaman produk, keranjang, pesanan, dan pengelolaan data penjualan. Fitur-fitur tersebut diperlukan karena sebagai dasar utama dalam pembuatan aplikasi *online store*.

Berdasarkan program tersebut, pada tugas akhir ini maka dibuatkan Aplikasi Arkatama *Store* berbasis *Web* yang memiliki sistem pengelolaan penjualan seperti pengelolaan produk, transaksi, dan pemesanan.

1.2. Rumusan Masalah

Berdasarkan latar belakang permasalahan yang ada, maka rumusan masalah ini adalah bagaimana merancang dan membuat sebuah Aplikasi Arkatama *Store* berbasis *Web* sehingga aplikasi dapat memiliki sistem pengelolaan penjualan seperti pengelolaan data produk, transaksi, dan pemesanan.

1.3. Batasan Masalah

Adapun batasan masalah pada aplikasi yang dibuat sebagai berikut :

1. Aplikasi Arkatama *Store* diperuntukkan bagi *Customer/Pembeli*, *Admin/Staff* Penjualan, dan *Manager* Toko/Perusahaan.
2. Aplikasi Arkatama *Store* memiliki 3 aktor yaitu *user/customer*, *admin/staff* penjualan, dan *manager* yang dapat melakukan *login*.
3. Aplikasi Arkatama *Store* digunakan untuk pengelolaan data jual beli suatu produk berdasarkan kategori tertentu.
4. Aplikasi Arkatama *Store* dibuat menggunakan *framework* dari *Cascading Style Sheet (CSS)* yaitu *bootstrap* sebagai tampilan *front-end* dan *framework* dari *Hypertext Preprocessor (PHP)* yaitu *codeigniter 3* sebagai *back-end*.

1.4. Tujuan

Tujuan dari pembuatan tugas akhir ini adalah merancang dan membuat Aplikasi Arkatama *Store* berbasis *Web* sehingga aplikasi memiliki fitur-fitur yang dapat digunakan untuk melakukan proses transaksi jual beli barang.

1.5. Manfaat

Manfaat dari pembuatan tugas akhir ini adalah sebagai bahan rujukan dan bahan masukan kepada penulis dalam membuat *online store* khususnya dalam bentuk *website* yang membahas mengenai Aplikasi Arkatama *Store* berbasis *Web* yang dibuat.

II. TINJAUAN PUSTAKA

2.1. Aplikasi

Aplikasi adalah suatu bagian dari perangkat lunak yang dapat digunakan untuk menyelesaikan masalah-masalah yang khusus yang dihadapi *user* dengan menggunakan kemampuan Komputer (Listianto, Fauzi, Irviani, Kasmi, & Garaika, 2017).

2.2. Website

Website atau disingkat *web*, dapat diartikan sekumpulan halaman yang terdiri dari beberapa laman yang berisi informasi dalam bentuk data digital baik berupa teks, gambar, video, audio, dan animasi lainnya yang disediakan melalui jalur koneksi internet (Abdulloh, 2015).

2.3. Online Store

Online Store atau Toko Online merupakan sistem yang memfasilitasi jual beli barang antara penjual dan pembeli dari mencari calon pembeli sampai menawarkan produk atau barang dengan memanfaatkan jaringan internet yang didukung dengan seperangkat alat elektronik sebagai penghubung dengan jaringan internet (Harmianto, Fuad, & Khairan, 2018).

2.4. HTML

HTML (hyper text markup language) adalah sebuah Bahasa standar yang digunakan untuk membuat halaman *web* dengan *hypertext*, dokumen yang dapat ditampilkan yaitu teks, gambar, dan tipe informasi lain seperti data *file* atau audio (Sutarman, 2009).

2.5. *CSS*

CSS singkatan dari *cascading style sheets*, yaitu skrip yang digunakan untuk mengatur desain *website*. Walaupun *HTML* mempunyai kemampuan untuk mengatur tampilan *website*, namun kemampuannya sangat terbatas. Fungsi *CSS* adalah memberikan pengaturan yang lebih lengkap agar struktur *website* yang dibuat dengan *HTML* terlihat lebih rapi dan indah (Abdulloh, 2015).

2.6. *Java Script*

Java Script adalah bahasa yang berbentuk kumpulan skrip berjalan pada suatu dokumen *HTML*. Bahasa ini adalah bahasa pemrograman untuk memberikan kemampuan tambahan terhadap *HTML* dengan mengizinkan pengekseskuan perintah-perintah disisi *user* variabel dan fungsi. Setiap instruksi diakhiri pada sisi *browser* bukan disisi server (Kustiyahningsih & Anamisa, 2011).

2.7. *PHP*

PHP singkatan dari *Hypertext Preprocessor* yang merupakan *server-side programing*, yaitu bahasa pemrograman yang diproses disisi server. Fungsi utama *PHP* dalam membangun *website* adalah untuk melakukan pengolahan data pada *database*. Data *website* akan dimasukkan ke *database*, diedit, dihapus, dan ditampilkan pada *website* yang diatur oleh *PHP* (Abdulloh, 2015).

2.8. *Database*

Basis Data atau *Database* adalah kumpulan data terstruktur. Sehingga dapat menambahkan, mengakses, dan memproses data yang tersimpan dalam *database* komputer, untuk membuat *database* dibutuhkan sistem manajemen basis data atau *database management system* (Suharyanto, Chandra, & Gunawan, 2017).

2.9. *Bootstrap*

Bootstrap merupakan *framework* ataupun *tools* untuk membuat aplikasi *web* ataupun situs *web responsive* secara cepat, mudah dan gratis. *Bootstrap* terdiri dari *CSS* dan *HTML* untuk menghasilkan *Grid*, *Layout*, *Typography*, *Table*, *Form*, *Navigation*, dan lain-lain. Di dalam *Bootstrap* juga sudah terdapat *jQuery plugins* untuk menghasilkan komponen *UI* yang cantik seperti *Transitions*, *Modal*, *Dropdown*, *Scrollspy*, *Tooltip*, *Tab*, *Popover*, *Alert*, *Button*, *Carousel* dan lain-lain. Dengan bantuan *Bootstrap*, kita bisa membuat *responsive website* dengan cepat dan mudah dan dapat berjalan sempurna pada *browser-browser* populer seperti *Chrome*, *Firefox*, *Safari*, *Opera* dan *Internet Explorer* (Alatas, 2013).

2.10. *Codeigniter*

Codeigniter adalah sebuah *web application framework* yang digunakan untuk membangun aplikasi *PHP* dinamis yang dibangun menggunakan konsep *Model View Controller development pattern*. *Codeigniter* menyediakan berbagai macam *library* yang dapat mempermudah dalam pengembangan dan termasuk *framework* tercepat dibandingkan dengan *framework* lainnya (Erinton, Negara, & Sanjoyo, 2017).

2.11. *XAMPP*

XAMPP adalah aplikasi yang berfungsi sebagai server yang berdiri sendiri (*localhost*), yang terdiri beberapa program antara lain : *Apache HTTP Server*, *MySQL database*, dan penerjemah bahasa yang ditulis dengan bahasa pemrograman *PHP* dan *Perl*. Nama *XAMPP* sendiri merupakan singkatan dari X empat sistem operasi, yang meliputi *Apache*, *MySQL*, *PHP* dan *Perl*. Program ini tersedia dalam *GNU (General Public License)*, merupakan *web server* yang mudah untuk digunakan yang dapat menampilkan halaman *web* yang dinamis (Jamil & Bunyamin, 2015).

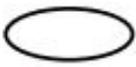
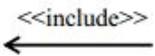
2.12. Web Server

Web Server adalah sebuah perangkat lunak server yang berfungsi menerima permintaan *HTTP* atau *HTTPS* dari klien yang dikenal dengan *web browser* dan mengirimkan kembali hasilnya dalam bentuk halaman-halaman *web* yang umumnya berbentuk dokumen *HTML* (Josi, 2017).

2.13. Use Case Diagram

Use Case Diagram digunakan untuk mengetahui fungsi apa saja yang ada didalam sistem informasi dan siapa saja yang berhak menggunakan fungsi-fungsi tersebut. *Use Case* diagram merupakan pemodelan untuk kelakuan (*behavior*) sistem informasi yang akan dibuat (Irawan & Simargolang, 2018). Simbol-simbol *use case* diagram dapat dilihat pada Tabel 1.

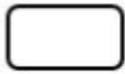
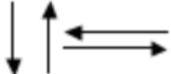
Tabel 1. Simbol-simbol *use case* diagram.

No	Simbol	Keterangan
1		Aktor : mewakili peran orang atau alat ketika berkomunikasi dengan <i>use case</i> .
2		<i>Use case</i> : deskripsi dari urutan aksi-aksi yang ditampilkan sistem yang menghasilkan suatu hasil yang terukur bagi suatu aktor.
3		<i>Association</i> : Abstraksi dari penghubung antara aktor dengan <i>use case</i> .
4		<i>Generalisasi</i> : Menunjukkan spesialisasi aktor untuk dapat berpartisipasi dengan <i>use case</i> .
5		Menunjukkan bahwa suatu <i>use case</i> seluruhnya merupakan fungsionalitas dari <i>use case</i> lainnya.

2.14. Activity Diagram

Activity Diagram merupakan gambaran *workflow* (aliran kerja) atau aktivitas dari sebuah sistem atau proses bisnis (Irawan & Simargolang, 2018). Simbol-simbol *activity* diagram dapat dilihat pada Tabel 2.

Tabel 2. Simbol-simbol *activity diagram*.

No	Simbol	Nama	Keterangan
1		<i>Activity</i>	Aktivitas atau perilaku yang dilakukan sistem atau aktor.
2		<i>Initial node</i>	Sebuah diagram aktivitas yang memiliki status awal.
3		<i>Activity final node</i>	Sebuah diagram aktivitas yang memiliki status akhir.
4		<i>Decision</i>	Percabangan dimana ada pilihan aktivitas yang lebih dari satu.
5		<i>Line connector</i>	Digunakan untuk menghubungkan satu simbol dengan simbol yang lain.

2.15. ERD

Entity Relationship Diagram (ERD) merupakan data model didasarkan pada persepsi terhadap dunia nyata yang tersusun atas kumpulan objek-objek dasar yang disebut entitas dan hubungan antar objek (Simarmata & Paryudi, 2006).

2.16. Framework

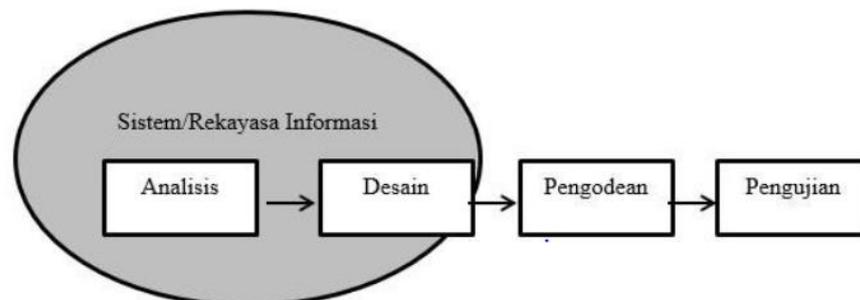
Framework adalah kumpulan instruksi-instruksi yang dikumpulkan dalam *class* dan *function-function* dengan fungsi masing-masing untuk memudahkan *developer* dalam memanggilnya tanpa harus menuliskan *syntax* program yang sama berulang-ulang serta dapat menghemat waktu (Wahyuni, Indriyanti, Ermawati, Fatah, & Ichsan, 2021).

2.17. RajaOngkir API

RajaOngkir merupakan *web service (API)* yang menyajikan informasi ongkos kirim dari berbagai jasa ekspedisi di Indonesia seperti POS Indonesia, JNE, TIKI, PCP, ESL, dan RPX. Salah satu kelebihan dari sistem RajaOngkir adalah data yang terpadu. Proses pengecekan biaya ongkos kirim ke jasa ekspedisi cukup dengan memasukkan nama kota asal, kota tujuan, dan berat, maka sistem RajaOngkir otomatis melakukan pengecekan ke semua jasa ekspedisi yang didukung (Prabowo, Arwani, & Pramono, 2021).

2.18. Metode Waterfall

Model *SDLC* air terjun (*waterfall*) sering juga disebut model sekuensial linier (*sequential linear*) atau alur hidup klasik (*classic life cycle*). Model air terjun menyediakan pendekatan alur hidup perangkat lunak secara sekuensial atau terurut dimulai dari analisis, desain, pengodean, dan pengujian (Rosa & Shalahuddin, 2015). Ilustrasi alur metode *waterfall* dapat dilihat melalui Gambar 1.



Gambar 1. Metode *Waterfall* (Rosa & Shalahuddin, 2015).

III. ANALISIS DAN PERANCANGAN

3.1. Analisis Kebutuhan Aplikasi

Penerapan metode *waterfall* yang pertama pada pembuatan Aplikasi Arkatama Store berbasis Web diperlukan analisis kebutuhan agar aplikasi dapat berjalan dengan semestinya. Tahapan ini bertujuan untuk mengklasifikasikan masalah-masalah, peluang, dan solusi yang mungkin diterapkan, serta menganalisa kebutuhan pada aplikasi. Analisis aplikasi yang dibuat dibagi menjadi dua yaitu kebutuhan fungsional dan kebutuhan non fungsional.

3.1.1. Analisis Kebutuhan Fungsional

Analisa kebutuhan fungsional dilakukan untuk mengetahui fungsi-fungsi yang diperlukan oleh aplikasi yang berkaitan dengan proses *input* dan *output*. Pada aplikasi yang dibuat, berdasarkan informasi dan hasil diskusi yang diperoleh dari instansi maka dibutuhkannya 3 aktor yaitu *user/customer*, *admin/staff* penjualan, dan *manager*. Oleh karena itu, berdasarkan penjelasan tersebut maka kebutuhan fungsional yang diperlukan diantaranya :

- a) Fungsi pada *user/customer*
 - 1) *Customer* dapat melihat informasi dan produk pada halaman *web*.
 - 2) *Customer* dapat mengelola produk ke dalam keranjang.
 - 3) *Customer* dapat melakukan *checkout*.
 - 4) *Customer* dapat mengelola pesanan.

- b) Fungsi pada *admin/staff* penjualan
- 1) Admin dapat mengelola produk.
 - 2) Admin dapat mengelola kategori.
 - 3) Admin dapat mengelola *hero*.
 - 4) Admin dapat mengelola data penjualan.
 - 5) Admin dapat menghapus testimoni
 - 6) Admin dapat mencetak laporan hasil penjualan.
- c) Fungsi pada *manager*
- 1) *Manager* dapat mengubah status dan menghapus produk yang ditampilkan.
 - 2) *Manager* dapat mengubah status dan menghapus *hero* yang ditampilkan.
 - 3) *Manager* dapat mencetak laporan hasil penjualan.

3.1.2. Analisis Kebutuhan Non Fungsional

Kebutuhan non fungsional merupakan kebutuhan penunjang berupa *hardware* dan *software* agar fungsi-fungsi yang diperlukan aplikasi dapat berjalan dengan baik. Kebutuhan non fungsional antara lain :

- a) Kebutuhan perangkat lunak (*software*)
- 1) Perangkat lunak (*software*) yang digunakan dalam proses pengembangan aplikasi adalah sebagai berikut :
 - *Windows 10 (64 bit)* sebagai sistem operasi
 - *Visual Studio Code* sebagai *text editor* untuk menulis kode program *HTML*, *CSS*, *Java Script*, dan *PHP*.
 - *XAMPP* untuk *web server* dan *database server*.
 - *Web Browser* untuk menjalankan aplikasi.
 - *Visual Paradigm Online* untuk membuat *Activity Diagram*, *ERD* dan *Use Case Diagram*.

- *Figma.com* untuk membuat desain antarmuka aplikasi.

2) Perangkat lunak (*software*) yang diperlukan agar dapat menjalankan aplikasi dengan spesifikasi minimum sebagai berikut :

- *Web browser* untuk menjalankan aplikasi.
- Akses internet.

b) Kebutuhan Perangkat Keras (*Hardware*)

1) Perangkat keras (*hardware*) yang digunakan dalam proses pengembangan aplikasi adalah sebagai berikut :

- *Processor* : *Intel Core i3-4005U*
- *Ram* : *10GB*
- Penyimpanan : *HDD 500GB*

2) Perangkat keras (*hardware*) yang digunakan agar dapat menjalankan aplikasi dengan spesifikasi minimum sebagai berikut :

- *Processor* : *Intel Celeron*
- *Ram* : *1GB*
- Penyimpanan : *HDD/SSD 120GB*

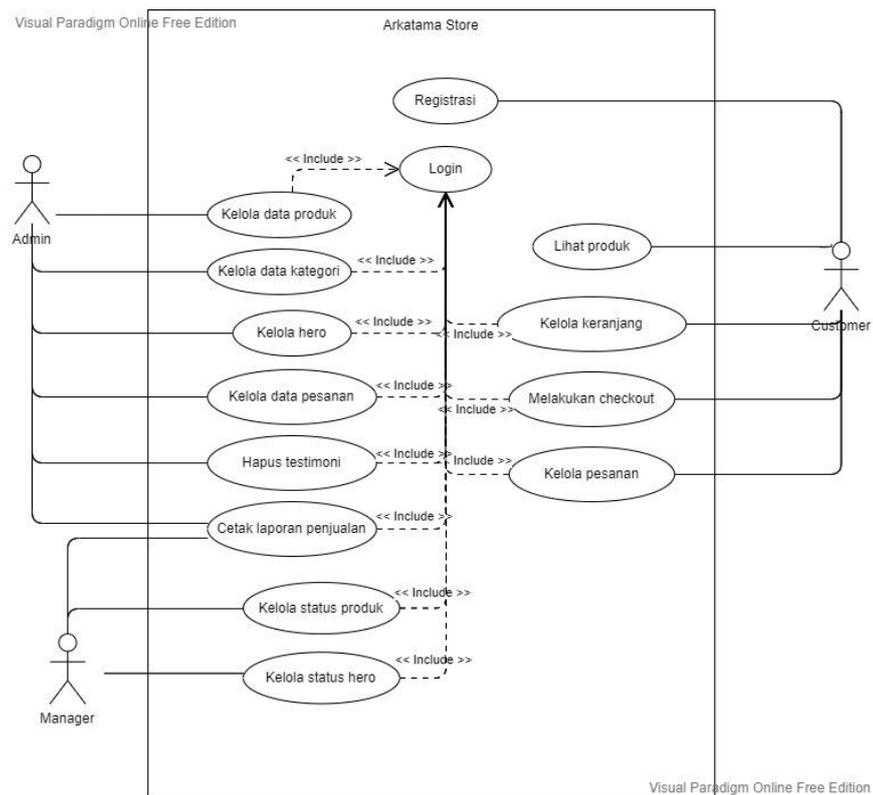
3.2. Perancangan Aplikasi

Selanjutnya tahapan metode *waterfall* yang kedua yaitu desain atau perancangan aplikasi. Perancangan aplikasi dibuat untuk mengilustrasikan aplikasi yang berjalan agar pengguna dapat memahami alur dari proses jalannya aplikasi tersebut. Tujuan dari perancangan aplikasi adalah untuk

memenuhi kebutuhan para pemakai aplikasi. Perancangan aplikasi terdiri dari desain aplikasi, desain proses, desain data, dan desain antarmuka.

3.2.1. Desain Aplikasi

Desain aplikasi pada Aplikasi *Arkatama Store* yang digambarkan melalui *Use Case* diagram sebagai berikut :



Gambar 2. *Use Case* Diagram *Arkatama Store*

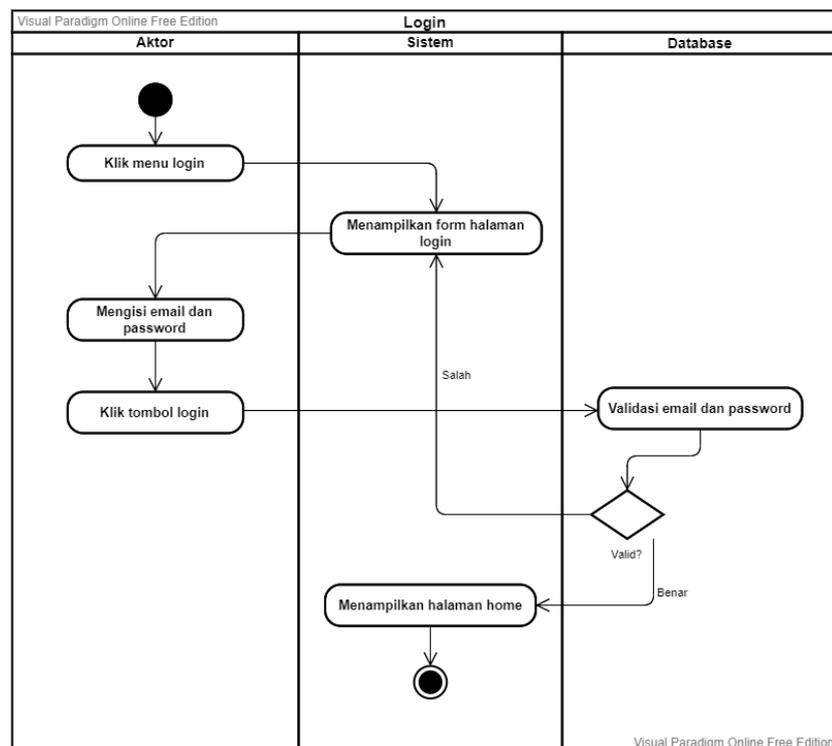
Gambar 2 merupakan *Use Case* Diagram Aplikasi *Arkatama Store*, terdapat *customer* yang bisa melihat produk-produk, mengelola data keranjang, *customer* dapat melakukan *checkout*, dan mengelola pesanan pada halaman *web*. Pada sisi admin, admin dapat mengelola produk, kategori, *hero*, data pesanan, hapus testimoni, dan admin juga dapat mencetak data hasil laporan penjualan. Kemudian pada sisi *manager*, *manager* dapat mengelola

status produk dan *hero* yang ingin ditampilkan pada halaman *web*, dan *manager* juga dapat mencetak hasil laporan penjualan.

3.2.2. Desain Proses

Desain Proses pada Aplikasi Arkatama *Store* berbasis *Web* digambarkan melalui *activity* diagram sebagai berikut :

a. Activity Diagram Login

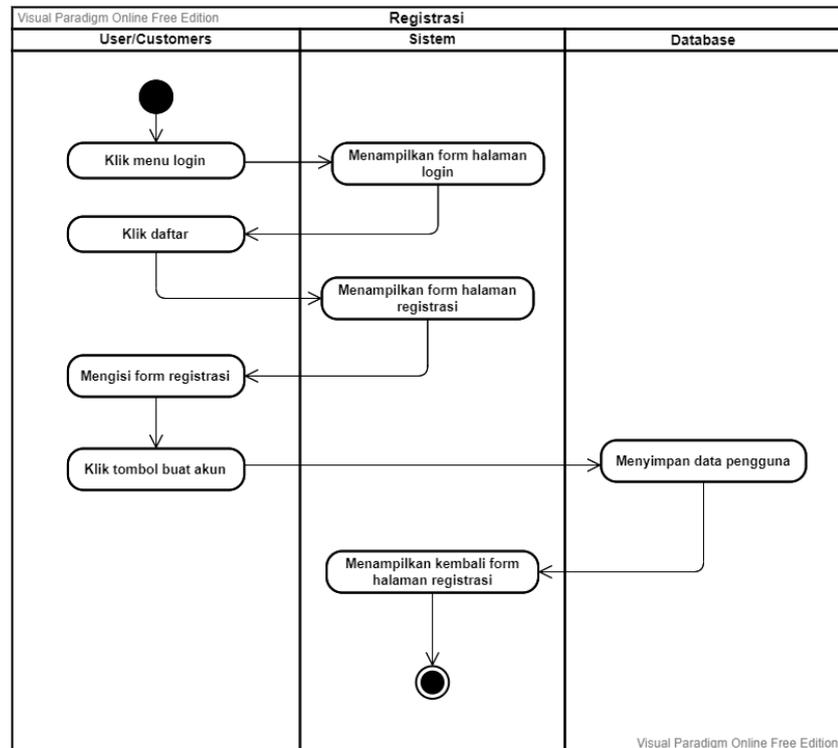


Gambar 3. Activity diagram login

Pada Gambar 3 merupakan *activity* diagram *login*. Pengguna yaitu *customer*, admin, atau *manager* mengakses aplikasi arkatama *store* dan klik menu *login*, pengguna diminta untuk mengisi *form email* dan *password*, kemudian klik tombol '*login*', jika *email* dan *password* yang di-input sesuai dan terdaftar pada *database* maka aplikasi akan menampilkan halaman *home* untuk *customer*, halaman admin untuk admin,

atau halaman *manager* untuk *manager*, jika salah maka akan kembali ke halaman *form login*.

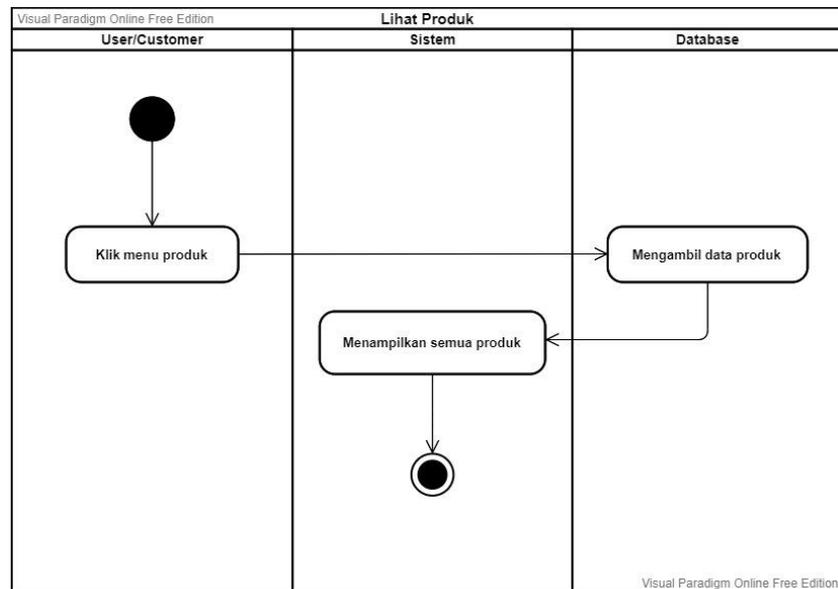
b. Activity Diagram Registrasi - Customer



Gambar 4. Activity diagram registrasi - customer

Pada Gambar 4 merupakan *activity* diagram registrasi. *Customer* mengakses aplikasi arkatama *store* dan memilih menu *login*. Kemudian pengguna melakukan klik pada *link* daftar maka *customer* diminta untuk mengisi *form* registrasi berupa nama, *email*, dan *password*. Setelah mengisi *form* dengan benar maka pengguna dapat menekan tombol 'buat akun' sehingga aplikasi menyimpan data tersebut ke *database* dan aplikasi menampilkan kembali *form* registrasi.

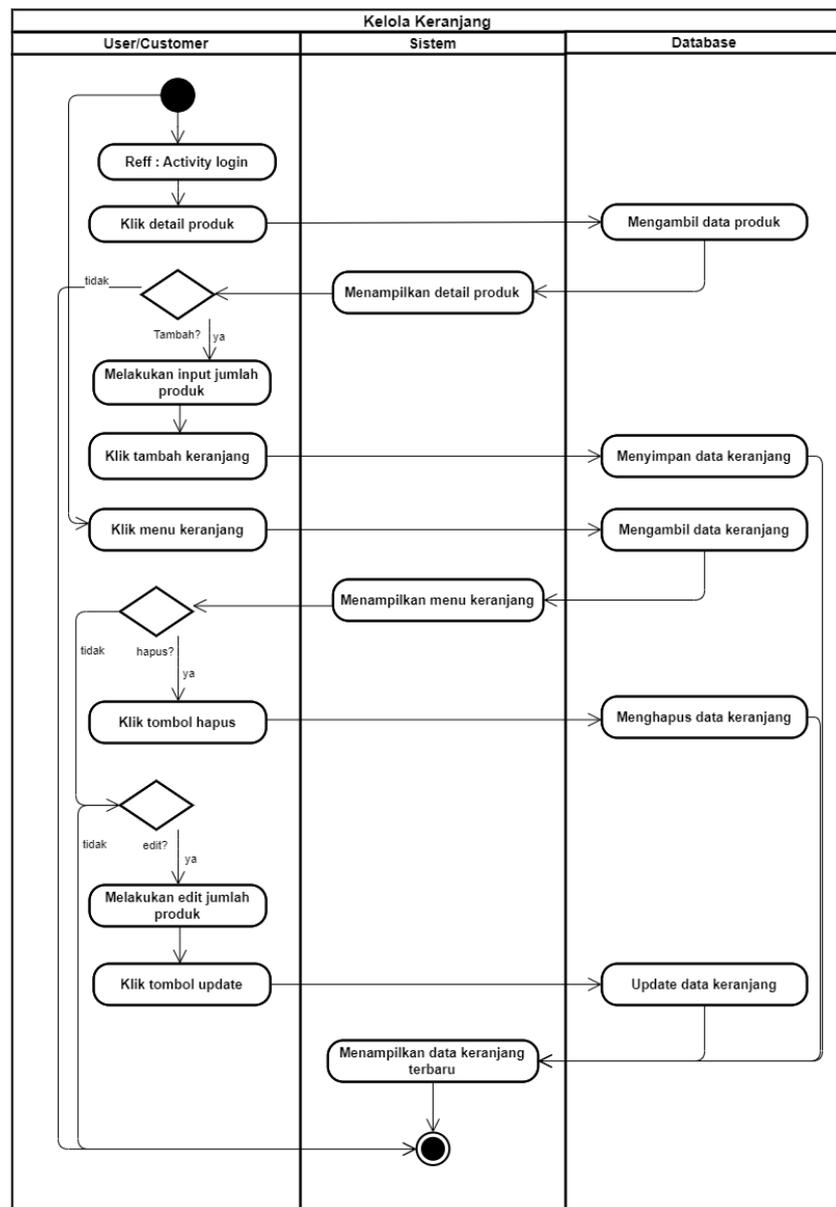
c. Activity Diagram Lihat Produk - Customer



Gambar 5. Activity diagram lihat produk - customer

Pada Gambar 5 merupakan *activity* diagram yang menjelaskan proses atau alur dari *customer* pada saat mengakses halaman produk, *Customer* dapat memilih menu produk untuk melihat semua produk yang dijual di aplikasi, kemudian *database* mengambil data produk untuk ditampilkan pada halaman menu produk aplikasi.

d. Activity Diagram Kelola Keranjang - Customer

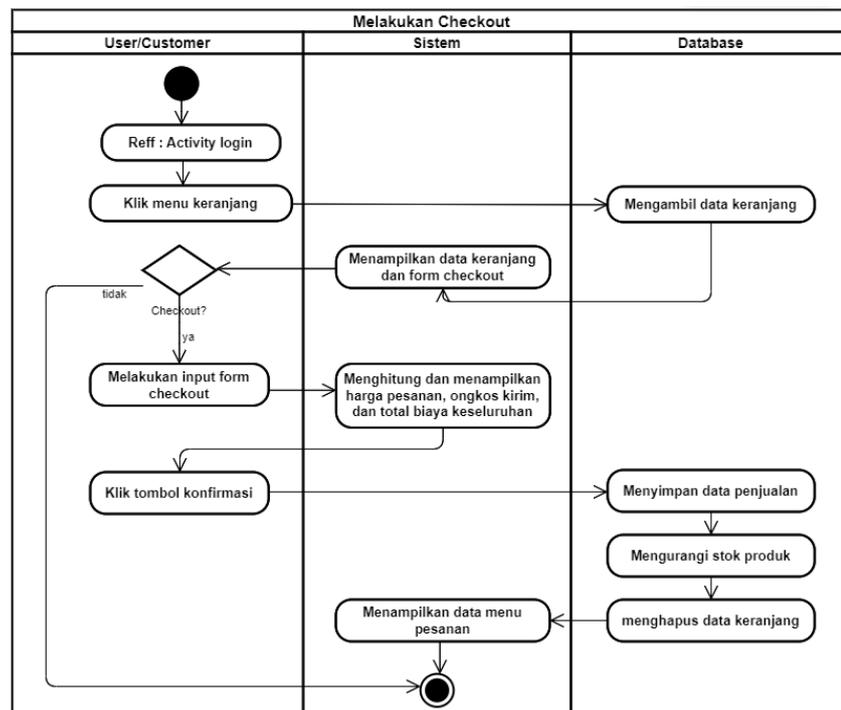


Gambar 6. Activity diagram kelola keranjang - customer

Pada Gambar 6 merupakan *activity* diagram yang menjelaskan proses atau alur *customer* pada saat mengelola keranjang. *Customer* telah *login* dapat melakukan klik *detail* pada produk yang ingin ditambahkan di keranjang lalu aplikasi mengambil data produk didalam *database* dan menampilkan seluruh *detail* produk pada halaman aplikasi, kemudian jika *customer* ingin

menambahkan produk pada keranjang, *customer* diminta untuk meng-*input*-kan jumlah produk dan melakukan klik pada tombol ‘tambah keranjang’ maka data akan disimpan ke *database*. Selain itu *customer* dapat melakukan proses edit dan hapus produk yang ada pada keranjang yaitu dengan cara klik menu keranjang, kemudian jika *customer* ingin menghapus data keranjang, *customer* dapat melakukan klik tombol ‘hapus’ maka aplikasi akan menghapus data keranjang di *database*. Kemudian *customer* juga dapat melakukan edit keranjang dengan cara meng-*input*-kan jumlah keranjang pada kolom *input* dan menekan tombol ‘*update keranjang*’ maka aplikasi akan meng-*update* data keranjang di *database*.

e. *Activity Diagram Checkout - Customer*

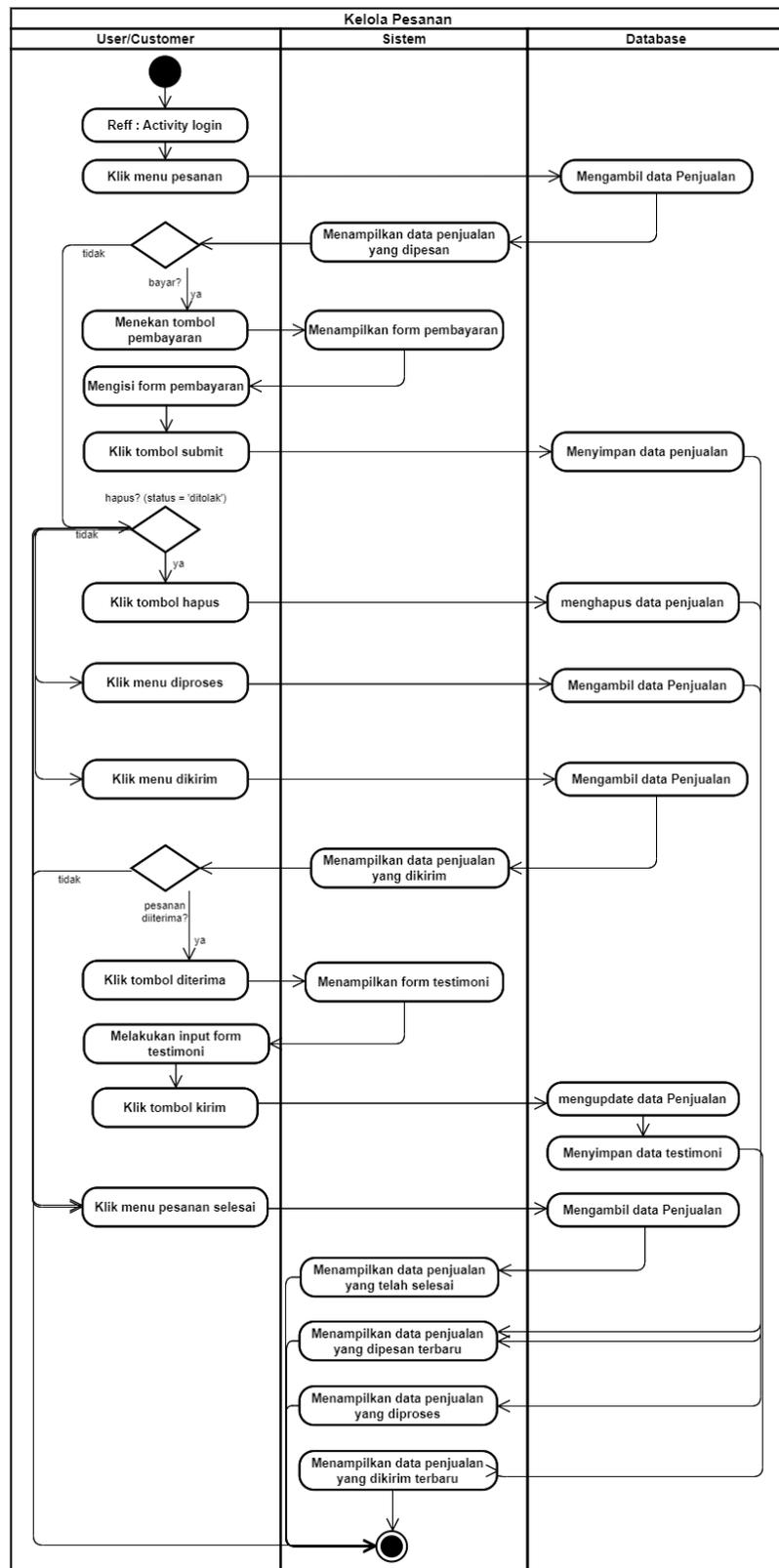


Gambar 7. *Activity diagram checkout - customer*

Pada Gambar 7 merupakan *activity diagram* yang menjelaskan proses atau alur *customer* pada saat melakukan *checkout*.

Customer telah *login* dapat memilih menu keranjang kemudian aplikasi menampilkan data keranjang yang ada pada *database* dan *form checkout*. Jika *customer* akan melakukan *checkout*, *Customer* diminta untuk mengisi *form* maka aplikasi menghitung dan menampilkan keseluruhan biaya yaitu biaya pesanan, ongkos kirim, dan total biaya berdasarkan biaya pesanan ditambah ongkos kirim. Kemudian jika telah mengisi *customer* dapat menekan tombol ‘konfirmasi’ atau ‘*checkout*’. Setelah itu data pesanan akan disimpan ke *database* dan menghapus data keranjang sebelumnya yang ada pada *database*. Setelah itu aplikasi menampilkan halaman pesanan *customer*.

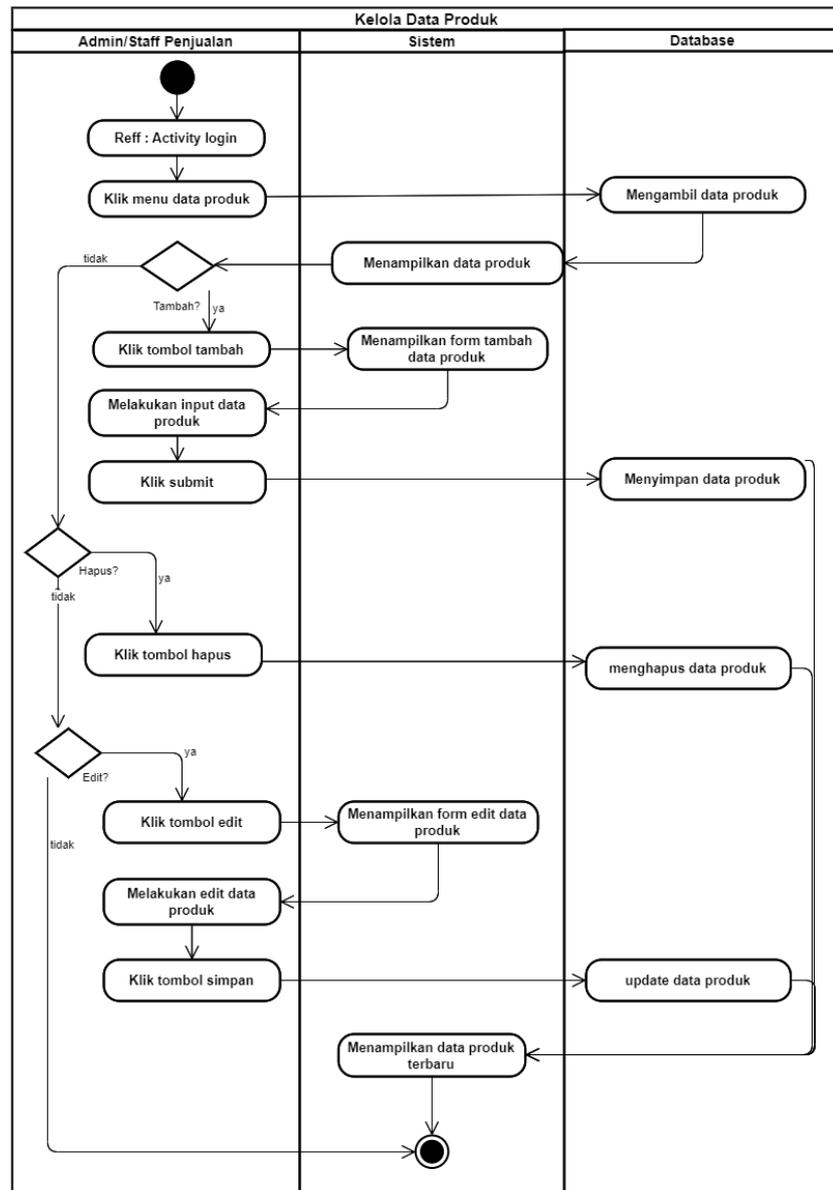
f. Activity Diagram Kelola Pesanan - Customer



Gambar 8. Activity diagram kelola pesanan - customer

Pada Gambar 8 merupakan *activity* diagram yang menjelaskan proses atau alur *customer* pada saat mengelola pesanan, *customer* dapat memilih menu pesanan kemudian aplikasi menampilkan pesanan *customer* yang ada pada *database*. Jika *customer* ingin membayar pesanan maka *customer* dapat melakukan klik pada tombol ‘pembayaran’ dan mengisi *form* data pembayaran, setelah mengisi *form* maka *customer* dapat melakukan klik pada tombol ‘submit’ dan aplikasi menambahkan data pesanan tersebut ke *database*. Apabila *customer* ingin menghapus pesanan yang sudah ‘ditolak’ maka *customer* dapat menekan tombol ‘hapus’ dan aplikasi akan menghapus data pesanan pada *database*. Selain itu *customer* dapat melihat menu diproses, halaman ini menampilkan data pesanan yang telah diverifikasi admin yang diambil melalui *database*. Selain itu juga *customer* dapat memilih menu dikirim, menu ini menampilkan pesanan *customer* yang telah dikirim ke kurir oleh admin, apabila *customer* telah menerima pesanan maka *customer* dapat menekan tombol ‘diterima’ dan mengisi *form* testimoni yang ditampilkan, setelah mengisi *form* *customer* dapat menekan tombol ‘kirim’ maka aplikasi akan meng-*update* status penjualan menjadi ‘selesai’ dan menambahkan data testimoni di *database*. Kemudian *customer* dapat juga memilih menu pesanan selesai yang menampilkan data transaksi-transaksi yang pernah *customer* beli.

g. Activity Diagram Kelola Data Produk - Admin

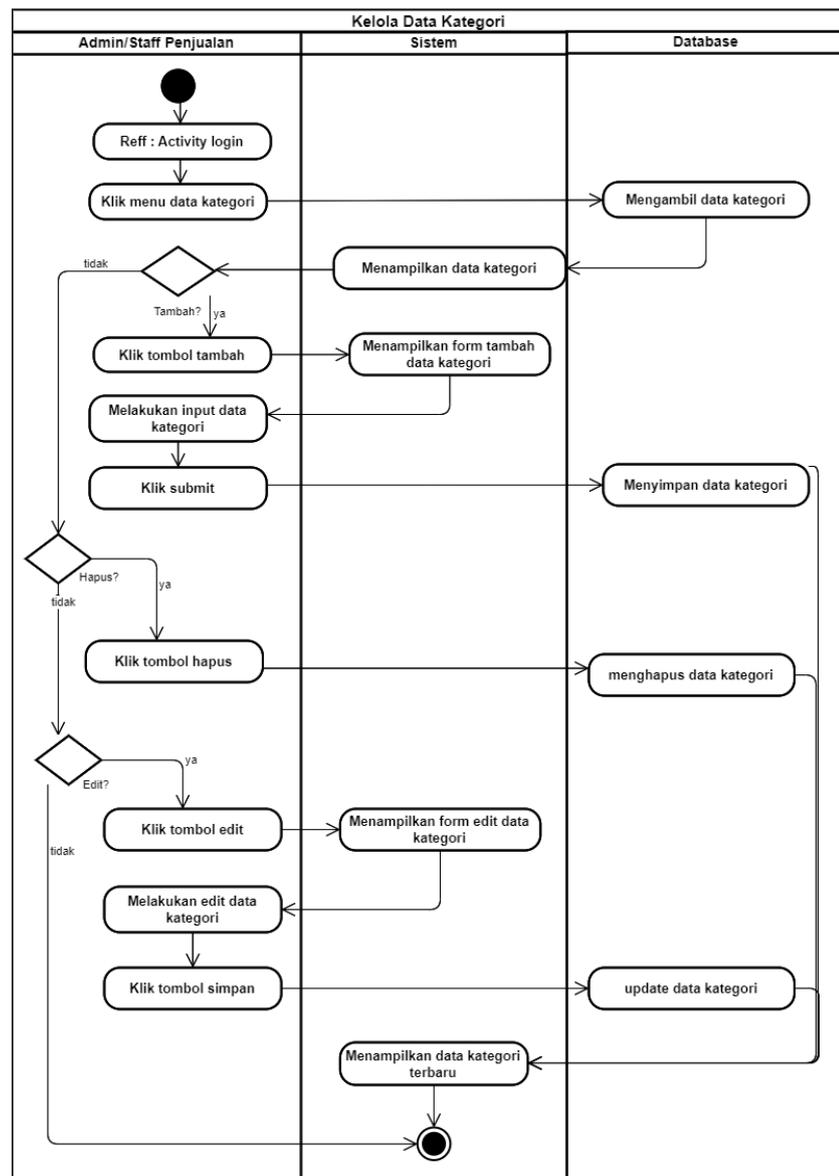


Gambar 9. Activity Diagram kelola data produk - admin

Pada Gambar 9 merupakan *activity* diagram yang menjelaskan proses atau alur admin pada saat mengelola data produk, admin telah *login* dapat memilih menu data produk kemudian aplikasi menampilkan data produk yang ada pada *database*. Jika admin ingin menambahkan produk maka admin dapat melakukan klik pada tombol 'tambah produk' dan mengisi *form* data produk,

setelah mengisi *form* maka admin dapat melakukan klik pada tombol '*submit*' dan aplikasi menambahkan data tersebut ke *database*. Setelah itu jika admin ingin menghapus data produk, admin dapat melakukan klik pada tombol 'hapus' dan aplikasi akan menghapus data tersebut di *database*. Kemudian jika admin ingin mengubah data produk, admin dapat melakukan klik pada tombol 'edit' dan melakukan edit pada *form* produk yang ingin diubah, setelah mengubah data produk pada *form* admin dapat menekan tombol 'simpan' dan aplikasi meng-*update* data tersebut pada *database*. Setelah itu aplikasi menampilkan kembali halaman data produk yang terbaru.

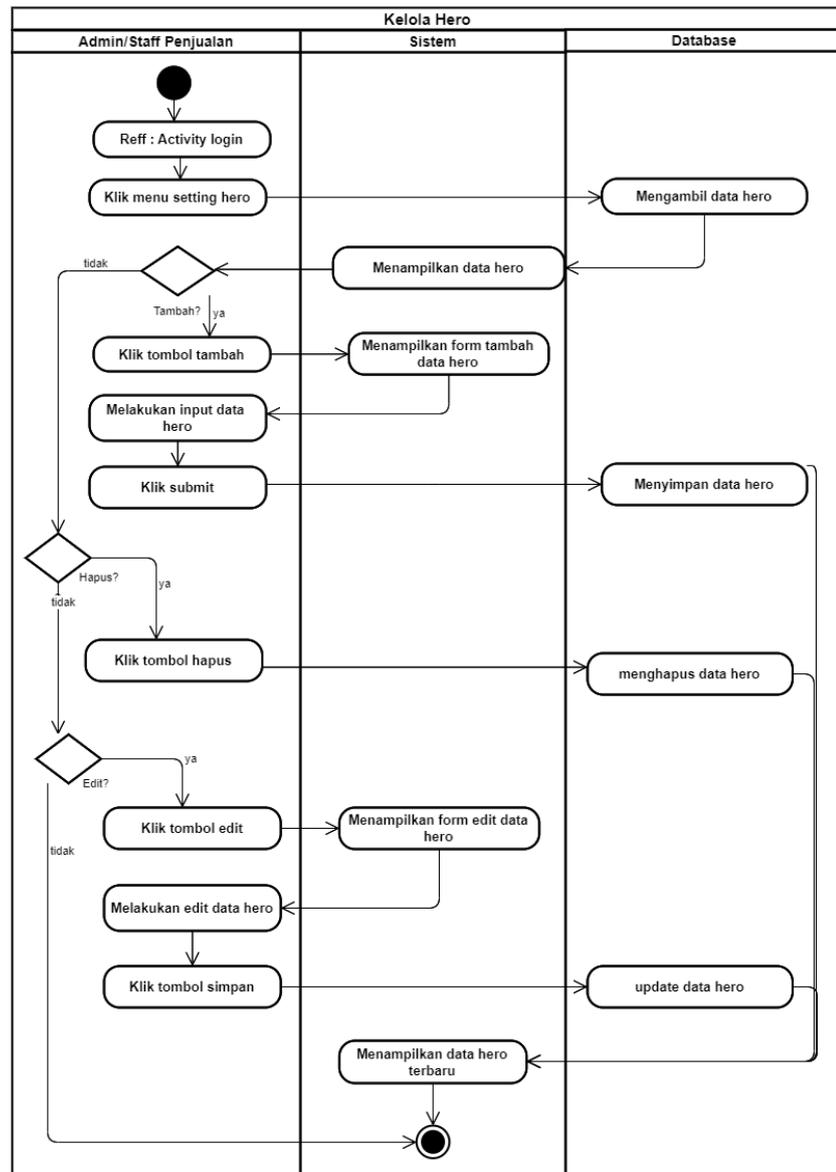
h. Activity Diagram Kelola Data Kategori - Admin



Gambar 10. Activity diagram kelola data kategori - admin

Pada Gambar 10 *activity* diagram menjelaskan proses atau alur admin pada saat mengelola data kategori, admin telah *login* dapat memilih menu data kategori kemudian aplikasi menampilkan data kategori yang ada pada *database*. Jika admin ingin menambahkan kategori maka admin dapat melakukan klik

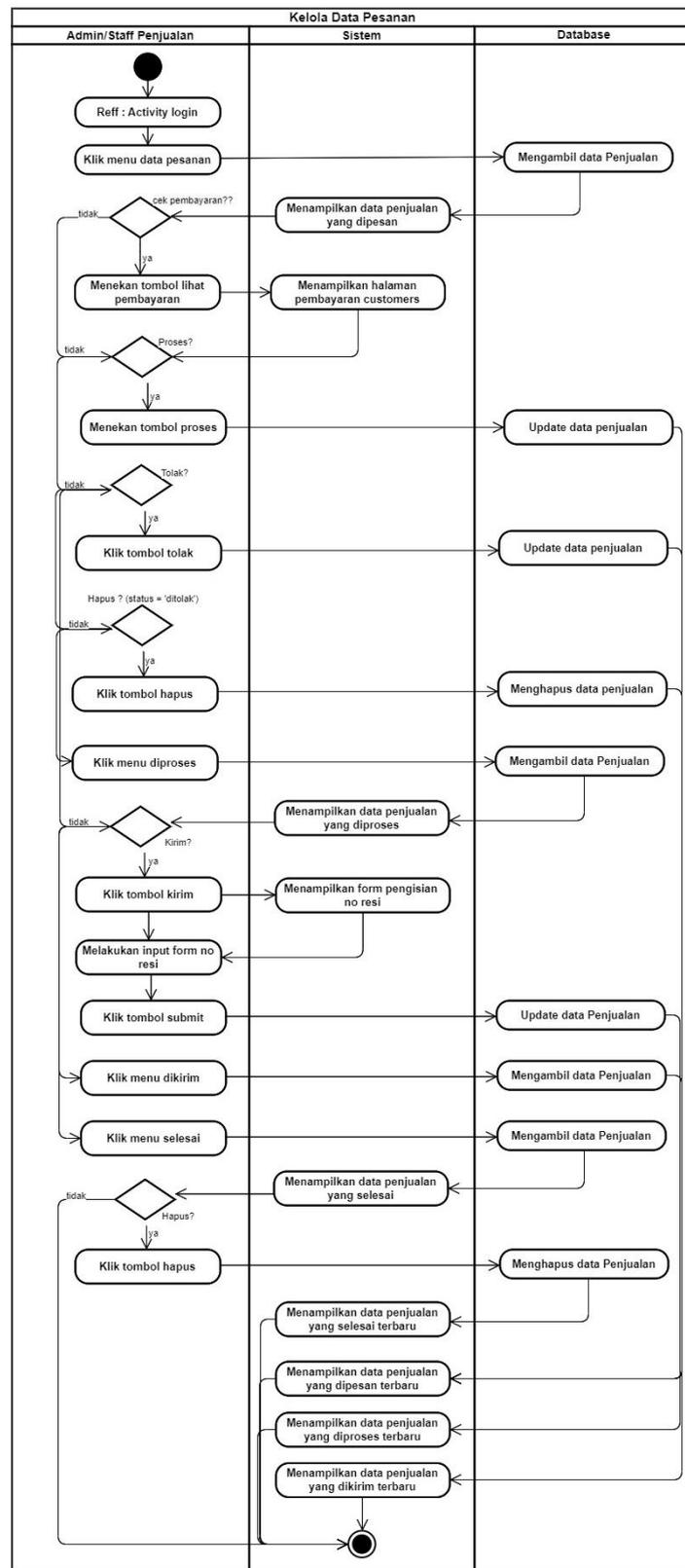
pada tombol ‘tambah kategori’ dan mengisi *form* data kategori, setelah mengisi *form* maka admin dapat melakukan klik pada tombol ‘*submit*’ dan aplikasi menambahkan data tersebut ke *database*. Setelah itu jika admin ingin menghapus data kategori, admin dapat melakukan klik pada tombol ‘hapus’ dan aplikasi akan menghapus data tersebut ke *database*. Kemudian jika admin ingin mengubah data kategori, admin dapat melakukan klik pada tombol ‘edit’ dan melakukan edit pada *form* kategori yang ingin diubah, setelah admin mengubah data kategori pada *form*, admin dapat menekan tombol ‘simpan’ dan aplikasi meng-*update* data tersebut ke *database*. Setelah itu aplikasi menampilkan kembali halaman data kategori yang terbaru.

i. Activity Diagram Kelola *Hero* - AdminGambar 11. Activity diagram kelola *hero* - admin

Pada Gambar 11 merupakan *activity* diagram yang menjelaskan proses atau alur admin pada saat mengelola data *hero*, admin telah *login* dapat memilih menu *setting hero* kemudian aplikasi menampilkan data *hero* yang ada pada *database*. Jika admin ingin menambahkan *hero* maka admin dapat melakukan klik pada tombol 'tambah *hero*' dan mengisi *form* data *hero*, setelah

mengisi *form* maka admin dapat melakukan klik pada tombol '*submit*' dan aplikasi menambahkan data tersebut ke *database*. Setelah itu jika admin ingin menghapus data *hero*, admin dapat melakukan klik pada tombol 'hapus' dan aplikasi akan menghapus data tersebut di *database*. Kemudian jika admin ingin mengubah data *hero*, admin dapat melakukan klik pada tombol 'edit' dan melakukan edit pada *form hero* yang ingin diubah, setelah mengubah data *hero* pada *form* admin dapat menekan tombol 'simpan' dan aplikasi meng-*update* data tersebut pada *database*. Setelah itu aplikasi menampilkan kembali halaman data *hero* yang terbaru.

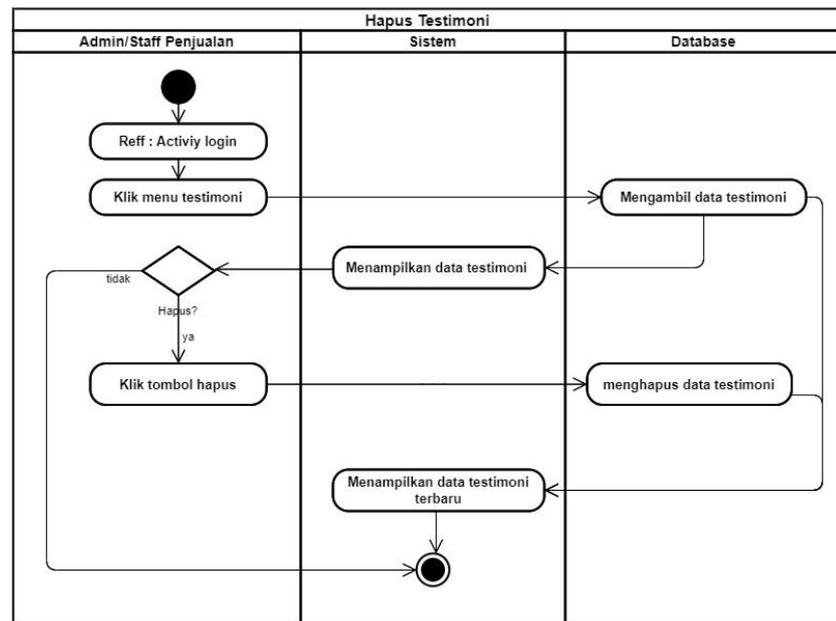
j. Activity Diagram Kelola Data Pesanan - Admin



Gambar 12. Activity diagram kelola data pesanan - admin

Pada Gambar 12 merupakan *activity* diagram yang menjelaskan proses atau alur admin pada saat mengelola pesanan, admin telah *login* dapat memilih menu data pesanan kemudian aplikasi menampilkan seluruh pesanan *customer* yang ada pada *database*. Jika admin ingin melihat pembayaran yang dilakukan oleh *customer* maka admin dapat melakukan klik pada tombol 'lihat pembayaran' dan aplikasi menampilkan data pembayaran *customer*. Apabila admin ingin memproses pesanan maka admin dapat menekan tombol 'proses' dan aplikasi akan meng-*update* status data pesanan tersebut menjadi 'diproses' pada *database*. Selain itu juga admin dapat menolak pesanan maka admin dapat menekan tombol 'tolak' dan aplikasi akan meng-*update* status data pesanan tersebut menjadi 'ditolak' pada *database*. Selain itu juga admin dapat menghapus pesanan *customer* yang telah ditolak maka admin dapat menekan tombol 'hapus' dan aplikasi akan menghapus data pesanan tersebut pada *database*. Selain itu admin dapat melihat menu 'diproses', halaman ini menampilkan data pesanan yang sedang diproses oleh admin yang diambil melalui *database*, Apabila admin telah mengirimkan barangnya pada kurir maka admin dapat menekan tombol 'kirim', kemudian admin mengisi *form* no resi, dan menekan tombol 'submit' maka setelah itu aplikasi meng-*update* status data pesanan tersebut menjadi 'dikirim' ke *database*. Selain itu juga admin dapat memilih menu 'dikirim', menu ini menampilkan pesanan *customer* yang telah dikirim ke kurir oleh admin dan admin menunggu respon dari *customer*. Kemudian *customer* dapat juga memilih menu pesanan 'selesai' yang menampilkan data transaksi-transaksi yang pernah *customer* beli, admin juga dapat menghapus pesanan yang telah selesai.

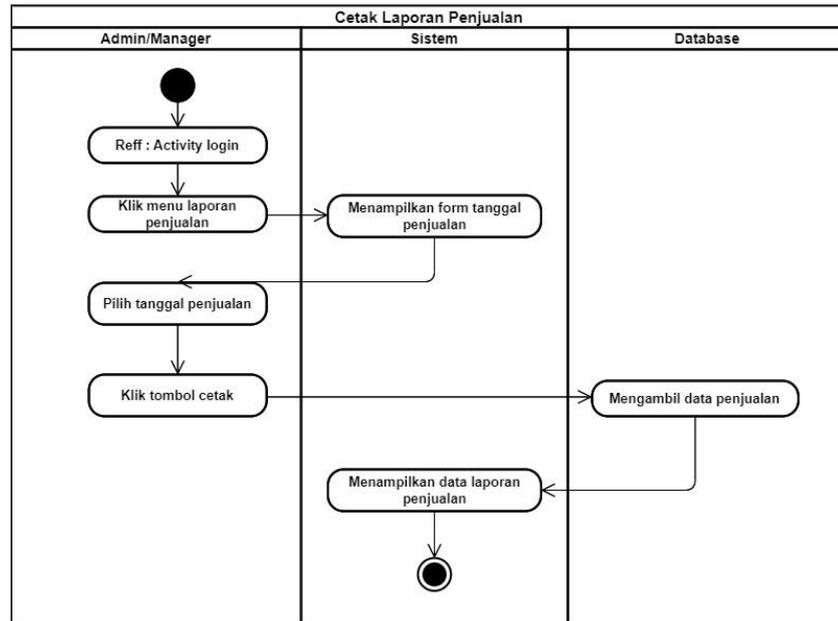
k. Activity Diagram Hapus Testimoni - Admin



Gambar 13. Activity diagram hapus testimoni - admin

Pada Gambar 13 *activity* diagram menjelaskan proses atau alur admin pada saat menghapus data testimoni, admin telah *login* dapat memilih menu data testimoni kemudian aplikasi menampilkan data testimoni yang ada pada *database*. jika admin ingin menghapus data testimoni, admin dapat melakukan klik pada tombol 'hapus' dan aplikasi akan menghapus data tersebut di *database*. Setelah itu aplikasi menampilkan kembali halaman data testimoni yang terbaru.

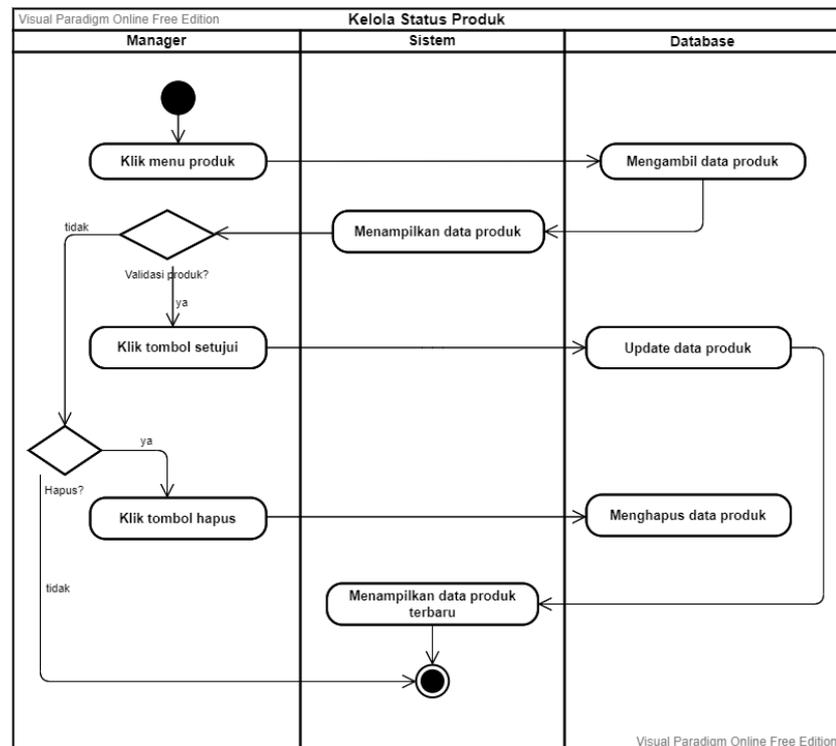
1. *Activity Diagram Cetak Laporan Penjualan - Admin dan Manager*



Gambar 14. *Activity diagram cetak laporan penjualan - admin dan manager*

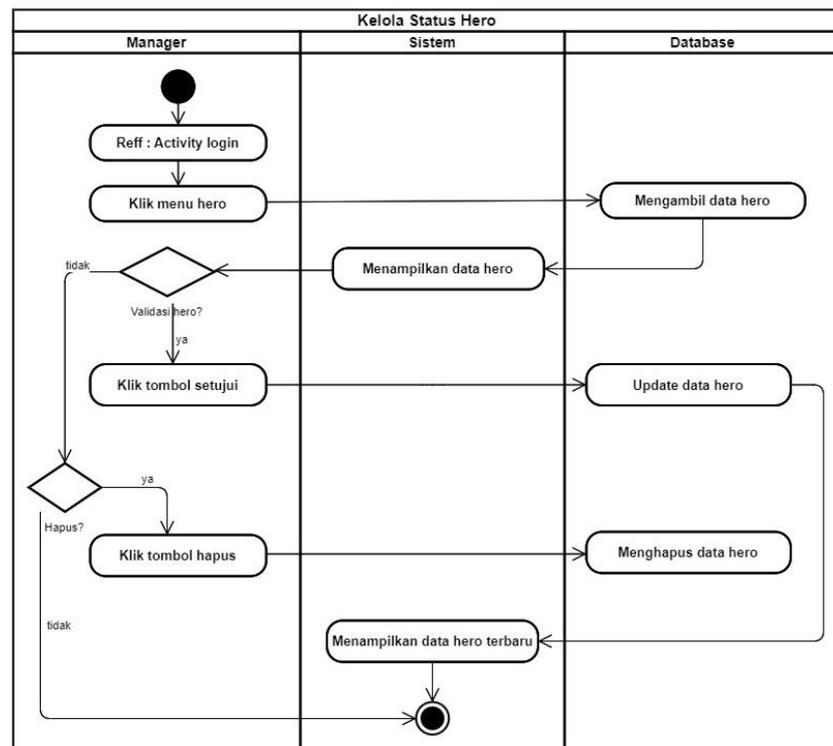
Pada Gambar 14 *activity diagram* menjelaskan proses atau alur admin dan *manager* pada saat melakukan cetak laporan penjualan, admin atau *manager* telah *login* dapat memilih menu laporan penjualan kemudian aplikasi menampilkan *form* tanggal yang digunakan untuk menampilkan data berdasarkan tanggal yang telah di-*input*, jika *form* telah di-*input* maka admin atau *manager* dapat menekan tombol 'cetak', lalu aplikasi mengambil data tersebut ke *database* untuk ditampilkan. Setelah itu aplikasi menampilkan halaman data laporan penjualan dalam bentuk *pdf*.

m. Activity Diagram Kelola Status Produk – *Manager*



Gambar 15. Activity diagram kelola status produk - *manager*

Pada Gambar 15 merupakan *activity* diagram yang menjelaskan proses atau alur *manager* pada saat mengelola status produk, *manager* telah *login* dapat memilih menu produk kemudian aplikasi menampilkan data produk yang ada pada *database*, jika *manager* menyetujui produk maka dapat dilakukan dengan cara menekan tombol ‘setujui’, lalu aplikasi meng-*update* data tersebut ke *database*. Selain itu *manager* dapat menghapus data produk yaitu dengan menekan tombol ‘hapus’ dan aplikasi akan menghapus data tersebut ke *database*. Setelah itu aplikasi menampilkan kembali halaman data produk yang terbaru.

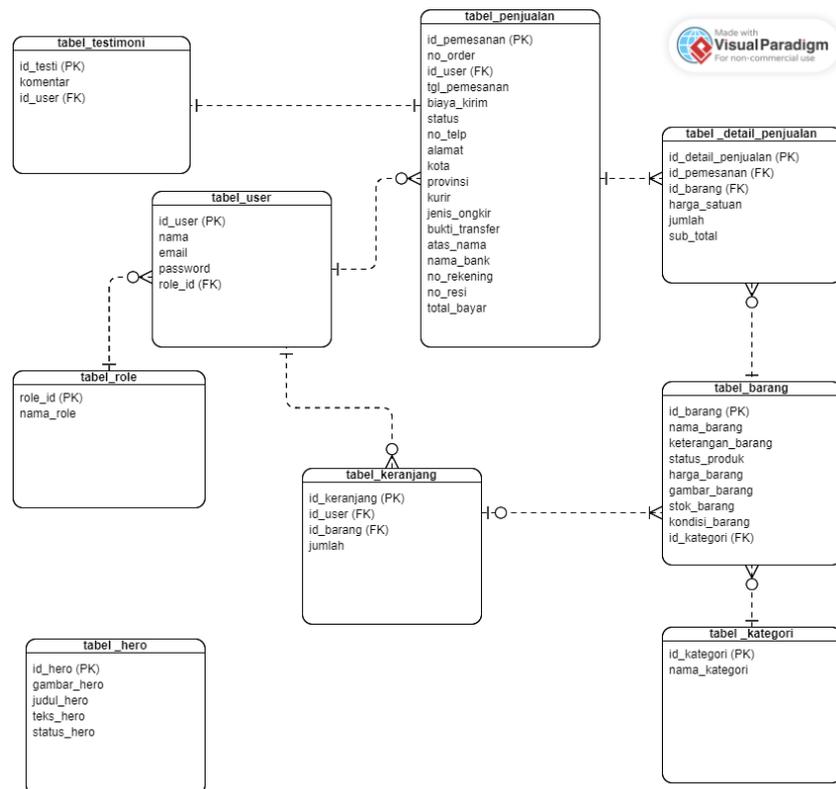
n. Activity Diagram Kelola Status *Hero* - *Manager*Gambar 16. Activity diagram kelola status *hero* - *manager*

Pada Gambar 16 merupakan *activity* diagram yang menjelaskan proses atau alur *manager* pada saat mengelola status *hero*, *manager* telah *login* dapat memilih menu *hero* kemudian aplikasi menampilkan data *hero* yang ada pada *database*, jika *manager* menyetujui *hero* maka dapat dilakukan dengan cara menekan tombol 'setujui', lalu aplikasi meng-*update* data tersebut pada *database*. Selain itu *manager* dapat menghapus data *hero* yaitu dengan menekan tombol 'hapus' dan aplikasi akan menghapus data tersebut pada *database*. Setelah itu aplikasi menampilkan kembali halaman data *hero* yang terbaru.

3.2.3. Desain Data

Desain Data pada proses pembuatan Aplikasi Arkatama *Store* meliputi *Entity Relationship Diagram (ERD)* dan struktur tabel *database*.

a. *Entity Relationship Diagram (ERD)*



Gambar 17. *Entity Relationship Diagram (ERD)* Arkatama *Store*

Pada Gambar 17 merupakan *ERD (Entity Relationship Diagram)* Aplikasi Arkatama *Store*, terdapat 9 entitas atau tabel pada *database* yaitu tabel_user, tabel_testimoni, tabel_role, tabel_barang, tabel_kategori, tabel_keranjang, tabel_hero, tabel_penjualan, dan tabel_detail_penjualan. Masing-masing tabel entitas kecuali tabel_hero dan tabel_testimoni memiliki relasi yaitu *one to many*. Pada tabel_testimoni dan tabel_penjualan memiliki relasi yaitu *one to one*.

b. Struktur Tabel *Database*

- 1) Tabel *User*, digunakan untuk menyimpan data pengguna aplikasi, struktur tabel *user* dapat dilihat pada Tabel 3.

Tabel 3. *User*

No	Atribut	Tipe Data	Length	Keterangan
1	id_user	int	11	Primary key
2	nama	varchar	45	
3	email	varchar	55	
4	password	varchar	256	
5	role_id	int	2	Foreign Key

- 2) Tabel *Role*, digunakan untuk menyimpan data *role* pengguna aplikasi, struktur tabel *role* dapat dilihat pada Tabel 4.

Tabel 4. *Role*

No	Atribut	Tipe Data	Length	Keterangan
1	role_id	int	2	Primary key
2	nama_role	varchar	25	

- 3) Tabel *Barang*, digunakan untuk menyimpan data produk yang dijual di aplikasi, struktur tabel *barang* dapat dilihat pada Tabel 5.

Tabel 5. *Barang*

No	Atribut	Tipe Data	Length	Keterangan
1	id_barang	int	11	Primary key
2	nama_barang	varchar	40	
3	keterangan_barang	varchar	225	

Tabel 5. Barang (lanjutan)

No	Atribut	Tipe Data	Length	Keterangan
4	status_produk	varchar	15	
5	harga_barang	int	11	
6	gambar_barang	varchar	125	
7	stok_barang	int	11	
8	kondisi_barang	varchar	35	
9	id_kategori	int	11	<i>Foreign Key</i>

- 4) Tabel Kategori, digunakan untuk menyimpan data kategori pada produk yang dijual di aplikasi, struktur tabel kategori dapat dilihat pada Tabel 6.

Tabel 6. Kategori

No	Atribut	Tipe Data	Length	Keterangan
1	id_kategori	int	11	<i>Primary key</i>
2	nama_kategori	varchar	35	

- 5) Tabel *Hero*, digunakan untuk menyimpan data *hero* yang ditampilkan di aplikasi, struktur tabel *hero* dapat dilihat pada Tabel 7.

Tabel 7. Hero

No	Atribut	Tipe Data	Length	Keterangan
1	id_hero	int	11	<i>Primary key</i>
2	gambar_hero	varchar	125	
3	judul_hero	varchar	50	
4	teks_hero	varchar	255	
5	status_hero	varchar	15	

- 6) Tabel Keranjang, digunakan untuk menyimpan data keranjang *customer*, struktur tabel keranjang dapat dilihat pada Tabel 8.

Tabel 8. Keranjang

No	Atribut	Tipe Data	Length	Keterangan
1	id_keranjang	int	11	Primary key
2	Id_user	int	11	Foreign Key
3	Id_barang	int	11	Foreign Key
4	jumlah	int	11	

- 7) Tabel Penjualan, digunakan untuk menyimpan data penjualan atau pesanan *customer*, struktur tabel penjualan dapat dilihat pada Tabel 9.

Tabel 9. Penjualan

No	Atribut	Tipe Data	Length	Keterangan
1	id_pemesanan	int	11	Primary key
2	no_order	varchar	6	
3	id_user	int	11	Foreign Key
4	tgl_pemesanan	date		
5	biaya_kirim	int	11	
6	status	varchar	10	
7	no_telp	varchar	15	
8	alamat	varchar	100	
9	kota	varchar	50	
10	provinsi	varchar	50	
11	kurir	varchar	20	
12	jenis_ongkir	varchar	50	
13	bukti_transfer	text		
14	atas_nama	varchar	50	
15	nama_bank	varchar	30	
16	no_rekening	varchar	25	
17	no_resi	varchar	50	
18	total_bayar	int	11	

- 8) Tabel *Detail Penjualan*, digunakan untuk menyimpan *detail* dari data penjualan atau pesanan *customer*, struktur tabel *detail* penjualan dapat dilihat pada Tabel 10.

Tabel 10. Detail Penjualan

No	Atribut	Tipe Data	Length	Keterangan
1	id_detail_penjualan	int	11	Primary key
2	id_pemesanan	int	11	Foreign Key
3	id_barang	int	11	Foreign Key
4	harga_satuan	int	11	
5	Jumlah	int	11	
6	sub_total	int	11	

- 9) Tabel Testimoni, digunakan untuk menyimpan data testimoni *customer*, struktur tabel testimoni dapat dilihat pada Tabel 11.

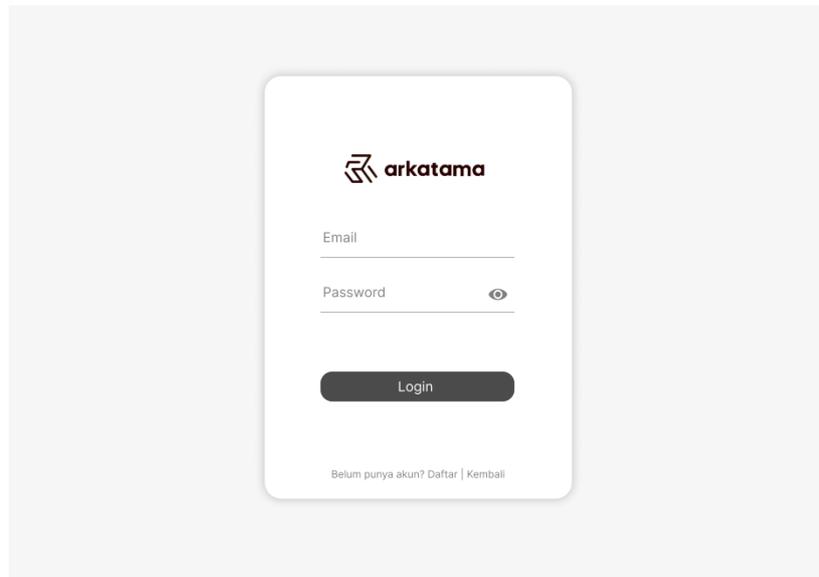
Tabel 11. Testimoni

No	Atribut	Tipe Data	Length	Keterangan
1	id_testi	int	11	Primary key
2	komentar	text		
3	id_user	int	11	Foreign Key

3.2.4. Desain Antarmuka

Desain Antarmuka pada perancangan Aplikasi Arkatama Store berbasis *Web* yang dibuat sebagai berikut.

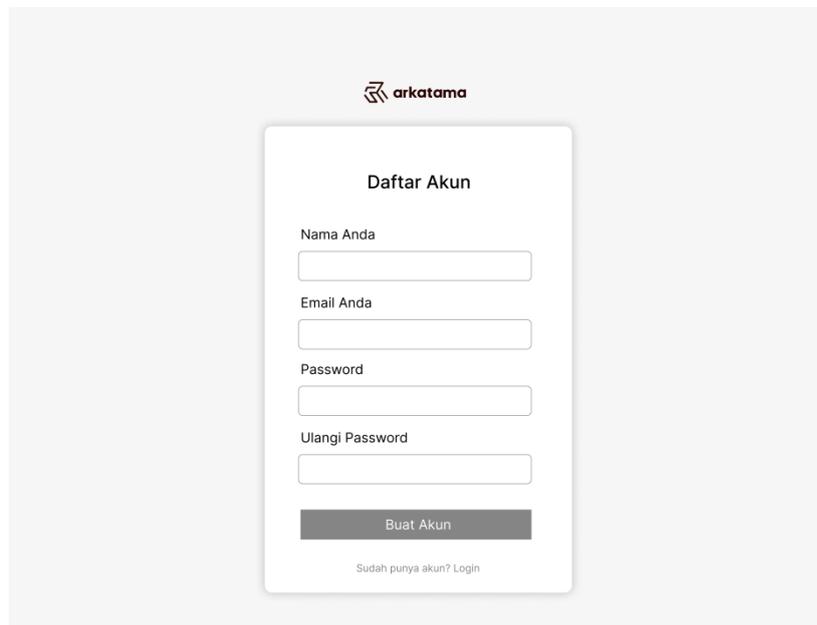
1) Desain *Interface Login*



Gambar 18. Desain *Interface Login*

Pada Gambar 18 merupakan desain *interface login*, tampilan ini akan muncul jika pengguna memilih menu *login*, terdapat *form* yang harus diisi yaitu *email* dan *password*, jika telah mengisi kemudian klik tombol '*login*' untuk *login* sebagai pengguna aplikasi.

2) Desain *Interface* Registrasi

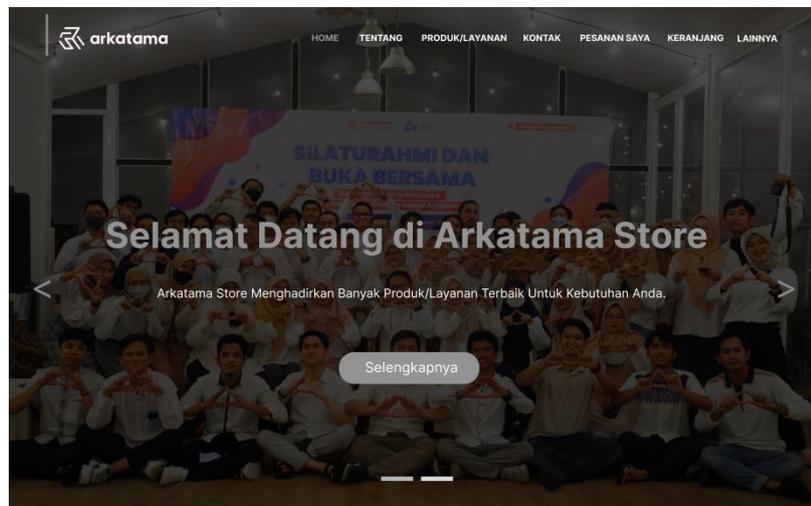


The image shows a registration form titled "Daftar Akun" for the brand "arkatama". The form is centered on a light gray background. It contains the following elements from top to bottom: the "arkatama" logo, the title "Daftar Akun", four input fields labeled "Nama Anda", "Email Anda", "Password", and "Ulangi Password", a dark gray button labeled "Buat Akun", and a link "Sudah punya akun? Login" at the bottom.

Gambar 19. Desain *Interface* Registrasi

Pada Gambar 19 merupakan desain *interface* registrasi *customer*, tampilan ini akan muncul jika pengguna memilih menu *login* dan klik *link daftar*, terdapat *form* registrasi yang harus diisi yaitu nama, *email* dan *password* kemudian klik tombol 'buat akun' untuk membuat akun.

3) Desain *Interface Home - Customer*



Gambar 20. Desain *interface home - customer*

Pada Gambar 20 merupakan desain halaman *home* aplikasi, tampilan ini akan muncul ketika pengguna mengakses *web* *arkatama store* dan *login* ke aplikasi sebagai *customer*. Jika telah *login* sebagai *customer*, terdapat menu halaman yang dapat dilihat dinavigasi yaitu *home*, *tentang*, *produk/layanan*, *kontak*, *pesanan*, *keranjang*, dan *lainnya*.

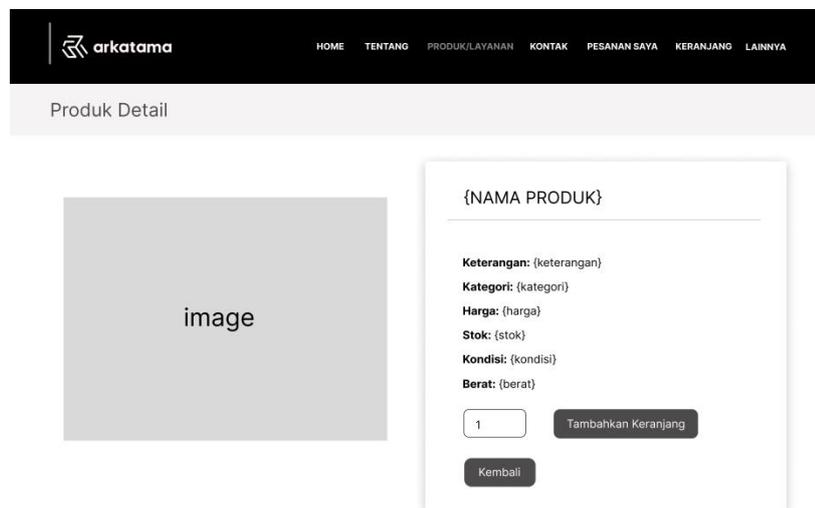
4) Desain *Interface* Menu Produk - *Customer*



Gambar 21. Desain *interface* menu produk - *customer*

Pada Gambar 21 merupakan desain halaman menu produk, tampilan ini akan muncul ketika pengguna mengakses *web* arkatama *store* atau *login* ke aplikasi sebagai *customer*. *Customer* memilih menu produk/layanan maka aplikasi menampilkan seluruh data barang yang dijual di aplikasi. Jika ingin melihat seluruh produk yang dijual maka *customer* bisa klik *link* 'lihat lainnya'. *Customer* dapat menekan tombol 'detail' jika ingin melihat *detail* dari produk

5) Desain *Interface Detail* Produk - *Customer*



Gambar 22. Desain *interface* detail produk - *customer*

Pada Gambar 22 merupakan desain halaman *detail* produk, tampilan ini akan muncul ketika *customer* menekan tombol ‘*detail*’ pada halaman produk. Halaman *detail* ini menampilkan seluruh informasi mengenai produk yang dipilih *customer*. Jika *customer* ingin menambahkan produk ini ke keranjang maka *customer* dapat meng-*input*-kan jumlah barang dan menekan tombol ‘tambah keranjang’.

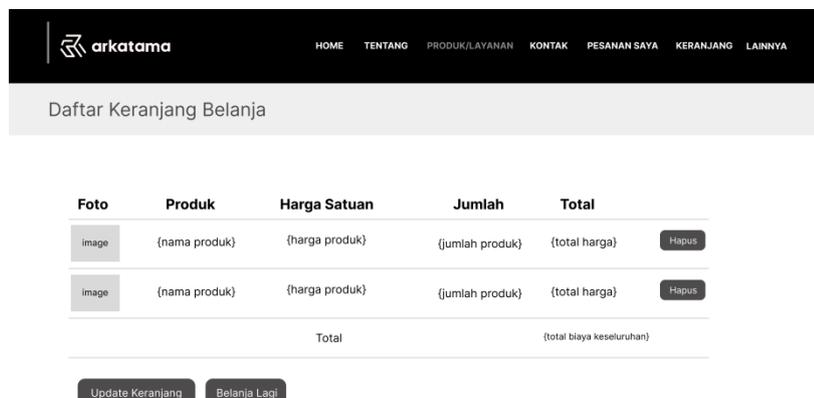
6) Desain *Interface* Keranjang - *Customer*


Foto	Produk	Harga Satuan	Jumlah	Total
image	{nama produk}	{harga produk}	{jumlah produk}	{total harga} <input type="button" value="Hapus"/>
image	{nama produk}	{harga produk}	{jumlah produk}	{total harga} <input type="button" value="Hapus"/>
Total				{total biaya keseluruhan}

Gambar 23. Desain *interface* keranjang - *customer*

Pada Gambar 23 merupakan desain halaman keranjang *customer*, tampilan ini akan muncul ketika *customer* memilih menu keranjang. Halaman keranjang ini menampilkan seluruh produk yang telah *customer* tambahkan ke keranjang. Halaman ini juga menghitung jumlah total biaya dari seluruh produk yang ditambahkan ke keranjang. Selain itu *customer* dapat mengedit dan menghapus data produk yang ada dikeranjang. Apabila *customer* ingin menambahkan produk lain ke dalam keranjang maka *customer* dapat menekan tombol ‘belanja lagi’, tombol ‘hapus’ untuk menghapus produk dikeranjang, dan tombol ‘*update* keranjang’ untuk mengedit keranjang.

7) Desain *Interface Checkout - Customer*

Nama	{nama customer}		
No telp	<input type="text"/>		
Alamat	<input type="text"/>		
Provinsi	<input type="text" value="Pilih provinsi"/>		
Kota	<input type="text" value="Pilih kota"/>		
Kurir	<input type="text" value="Pilih kurir"/>		
Biaya Ongkir	{jenis kurir}	{waktu pengiriman}	{biaya ongkir}
Total Belanja	{total belanja}		
Grand Total	{grand total}		
<input type="button" value="Checkout"/>			

Gambar 24. Desain *interface checkout - customer*

Pada Gambar 24 merupakan desain halaman *checkout customer*, tampilan ini akan muncul ketika *customer* memilih menu keranjang dan halaman *checkout* ini berada dibawah dari halaman keranjang. Halaman *checkout* menampilkan *form* yang harus diisi oleh *customer* dan di halaman ini juga aplikasi menghitung total biaya ongkos kirim ditambah total belanja *customer* yang ada dikeranjang. Penghitungan biaya ongkos kirim ini dilakukan menggunakan *API* dari RajaOngkir. Setelah *customer* menyetujui pesanan maka *customer* dapat menekan tombol '*checkout*', kemudian aplikasi menambahkan data pesanan tersebut ke *database* dan menampilkan halaman data pesanan *customer*.

8) Desain *Interface* Kelola Pesanan ‘diorder’ - *Customer*

Tanggal Transaksi	No. Transaksi	Ekspedisi	Detail	Total Transaksi	Status/Aksi
{tanggal}	{no transaksi}	{ekspedisi}	Detail	{total transaksi} Sekera lakukan pembayaran	Bayar
{tanggal}	{no transaksi}	{ekspedisi}	Detail	{total transaksi} Sekera lakukan pembayaran	Bayar

Gambar 25. Desain *interface* pesanan ‘diorder’ - *customer*

Pada Gambar 25 merupakan desain halaman pesanan *customer*, tampilan ini akan muncul ketika *customer* memilih menu pesanan. Halaman pesanan menampilkan data pesanan *customer* yaitu pesanan yang ‘diorder’, ‘diproses’, ‘dikirim’, dan ‘selesai’. Pada halaman ‘diorder’ *customer* dapat melakukan pembayaran dan memantau proses pesanan.

No. Rekening Toko

Silahkan transfer ke no. rekening dibawah ini sebesar

{total biaya}

Bank	No. Rekening	Atas Nama
{bank}	{rekening}	{atas nama}

Bukti Pembayaran

Atas Nama

Nama Bank

No Rekening

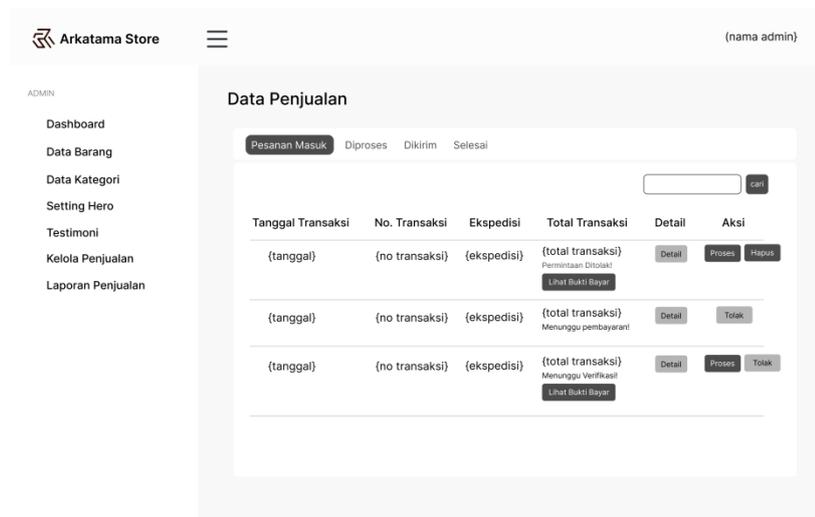
Bukti Bayar
[Chose File](#)

[Submit](#)

Gambar 26. Desain *interface* pembayaran pesanan - *customer*

Gambar 26 merupakan desain halaman pembayaran pesanan di *customer*, tampilan ini akan muncul ketika *customer* menekan tombol ‘bayar’ pada pesanan yang ada di halaman pesanan ‘diorder’. Halaman pembayaran menampilkan *form* yang harus diisi oleh *customer*. *Customer* mentransfer sejumlah uang sesuai dengan total biaya ke no rekening yang tertera diaplikasi, Kemudian setelah itu *customer* mengisi *form* bukti pembayaran dan menekan tombol ‘submit’ agar aplikasi menyimpan data pembayaran tersebut ke *database*.

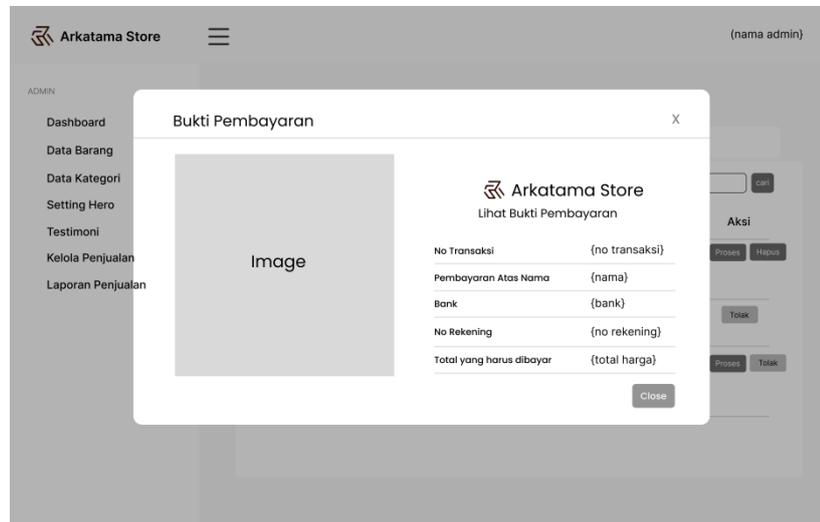
9) Desain *Interface* Kelola Pesanan ‘diorder’ - Admin



Gambar 27. Desain *interface* kelola pesanan ‘masuk’ - admin

Pada Gambar 27 merupakan desain halaman kelola penjualan admin, tampilan ini akan muncul ketika admin memilih menu kelola penjualan. Halaman kelola penjualan menampilkan data pesanan *customer* yaitu pesanan yang ‘masuk’, ‘diproses’, ‘dikirim’, dan ‘selesai’. Pada halaman pesanan yang ‘masuk’ admin dapat melihat seluruh data pesanan yang masuk, dan dapat melakukan beberapa hal seperti admin dapat

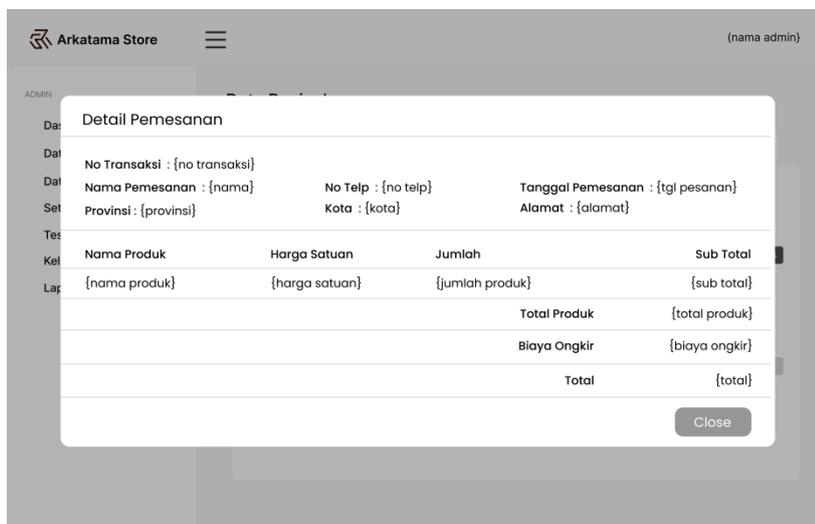
melanjutkan proses pesanan, menolak pesanan, dan menghapus pesanan.



Gambar 28. Desain *interface* bukti pembayaran - admin

Gambar 28 merupakan desain halaman 'lihat bukti pembayaran' di admin, tampilan ini akan muncul ketika admin menekan tombol 'lihat bukti bayar' pada pesanan yang ada di halaman pesanan 'masuk'. Halaman bukti bayar ini menampilkan foto beserta data pembayaran yang telah di-*input*-kan oleh *customer*.

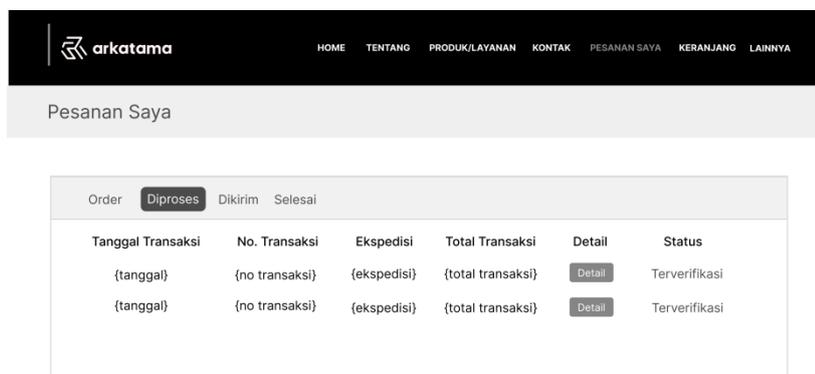
10) Desain *Interface Detail Pesanan* - Admin



Gambar 29. Desain *interface detail* pesanan - admin

Pada Gambar 29 merupakan desain halaman *detail* pesanan di admin, tampilan ini akan muncul ketika admin menekan tombol '*detail*' pada pesanan yang ada di halaman kelola penjualan. Halaman *detail* menampilkan seluruh data informasi pesanan *customer*.

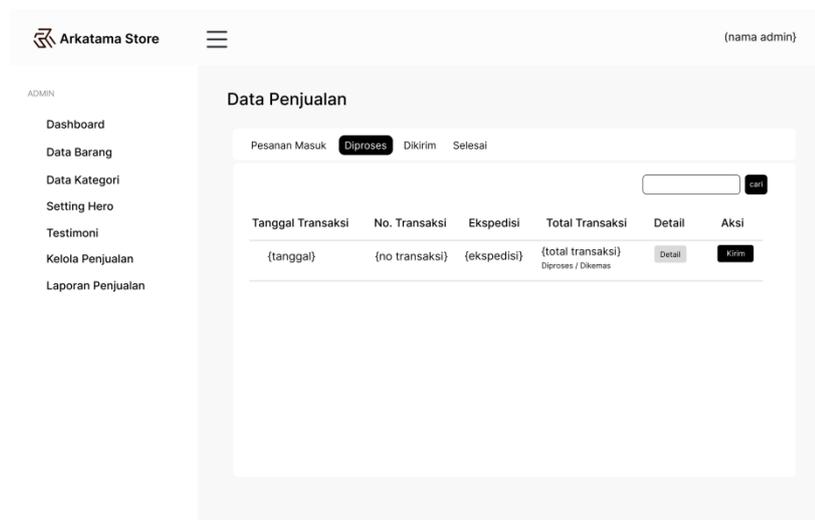
11) Desain *Interface Kelola Pesanan 'diproses'* - Customer



Gambar 30. Desain *interface* pesanan '*diproses*' - customer

Pada Gambar 30 merupakan desain halaman pesanan *customer*, tampilan ini akan muncul ketika *customer* memilih menu pesanan. Pada halaman ‘diproses’ menampilkan seluruh pesanan yang telah diverifikasi oleh admin. Pada tahap ini admin sedang menyiapkan pesanan dan mengirimkan produk ke kurir.

12) Desain *Interface* Kelola Pesanan ‘diproses’ - Admin



Gambar 31. Desain *interface* kelola pesanan ‘diproses’ - admin

Pada Gambar 31 merupakan desain halaman kelola penjualan di admin, tampilan ini akan muncul ketika admin memilih menu kelola penjualan. Pada halaman ‘diproses’ admin menyiapkan dan mengirimkan produk yang dipesan oleh *customer* ke kurir. Setelah mengirimkan pesanan ke kurir maka admin dapat menekan tombol ‘kirim’.

Gambar 32. Desain *interface form* no resi - admin

Gambar 32 merupakan desain halaman *form* pengisian no resi di admin. Tampilan ini akan muncul ketika *customer* menekan tombol ‘*irim*’ pada pesanan yang ada di halaman pesanan ‘*diproses*’. Halaman ini menampilkan *form* no resi yang harus diisi oleh *customer*. *Customer* mengisi no resi dan menekan tombol ‘*irim*’ maka aplikasi menambahkan data no resi tersebut ke *database*.

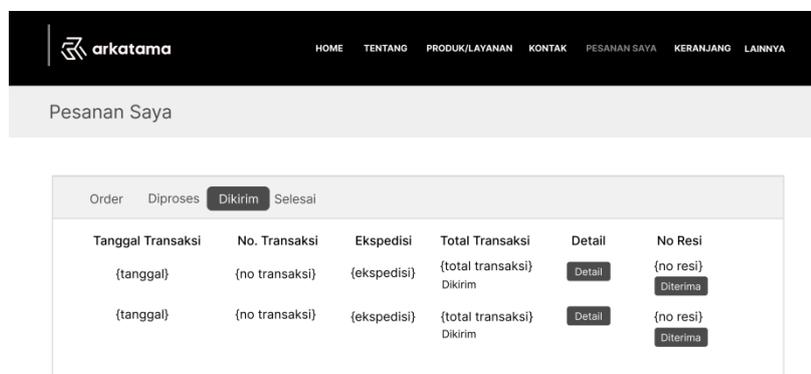
13) Desain *Interface* Kelola Pesanan ‘*dikirim*’ - Admin

Tanggal Transaksi	No. Transaksi	Ekspedisi	Total Transaksi	Detail	No Resi
{tanggal}	{no transaksi}	{ekspedisi}	{total transaksi} Dikirim	Detail	{no resi}

Gambar 33. Desain *interface* kelola pesanan ‘*dikirim*’ - admin

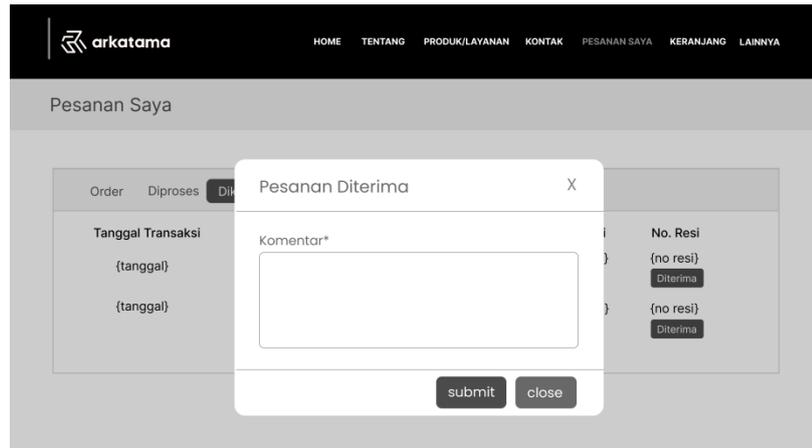
Pada Gambar 33 merupakan desain halaman kelola penjualan di admin, tampilan ini akan muncul ketika admin memilih menu kelola penjualan. Pada halaman ‘dikirim’ admin dapat melihat seluruh data pesanan *customer* yang telah dikirim ke kurir dan menunggu konfirmasi dari *customer* bahwa produk telah diterima.

14) Desain *Interface* Kelola Pesanan ‘dikirim’ - *Customer*



Gambar 34. Desain *interface* pesanan ‘dikirim’ - *customer*

Pada Gambar 34 merupakan desain halaman pesanan di *customer*, tampilan ini akan muncul ketika admin memilih menu pesanan. Pada halaman ‘dikirim’ *customer* dapat melihat seluruh data pesanan yang telah dikirim ke kurir dan *customer* dapat menekan tombol ‘diterima’ jika produk telah diterima.



Gambar 35. Desain *interface form* testimoni - *customer*

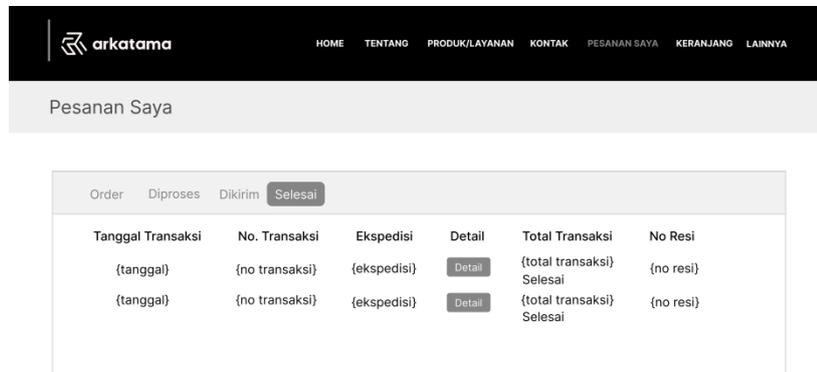
Gambar 35 merupakan desain halaman *form* pengisian testimoni di *customer*. Tampilan ini akan muncul ketika *customer* menekan tombol ‘diterima’ pada pesanan yang ada di halaman pesanan ‘dikirim’. Halaman ini menampilkan *form* testimoni yang harus di isi oleh *customer*. *Customer* mengisi pesan atau komentar terhadap produk atau layanan yang telah diterima.



Gambar 36. Desain *interface* halaman testimoni - *customer*

Gambar 36 merupakan desain halaman data testimoni di *customer*. Tampilan ini dapat dilihat di halaman testimoni yang letaknya berada dibawah halaman menu produk. Halaman ini menampilkan seluruh data testimoni terbaru 1-10 dari *customer* yang pernah melakukan transaksi di aplikasi.

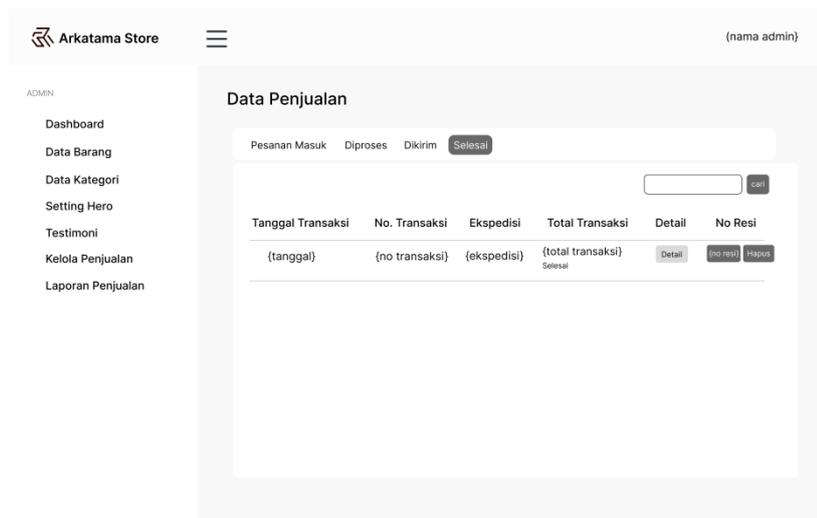
15) Desain *Interface* Kelola Pesanan ‘selesai’ - *Customer*



Gambar 37. Desain *interface* pesanan ‘selesai’ - *customer*

Pada Gambar 37 merupakan desain halaman pesanan *customer*, tampilan ini akan muncul ketika *customer* memilih menu pesanan. Pada halaman ‘selesai’ menampilkan seluruh pesanan yang telah diterima oleh *customer*.

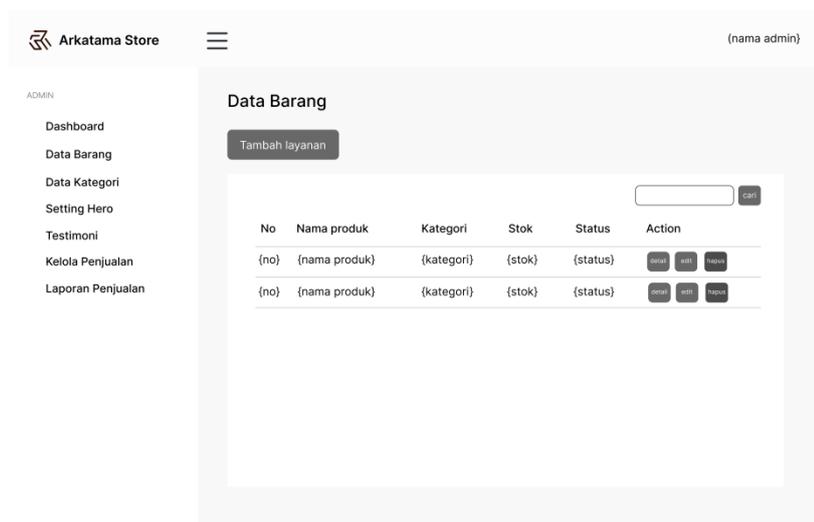
16) Desain *Interface* Kelola Pesanan ‘selesai’ - Admin



Gambar 38. Desain *interface* kelola pesanan ‘selesai’ - admin

Pada Gambar 38 merupakan desain halaman kelola penjualan di admin, tampilan ini akan muncul ketika admin memilih menu kelola penjualan. Pada halaman ‘selesai’ admin dapat melihat seluruh data pesanan yang telah diterima oleh *customer*.

17) Desain *Interface* Kelola Data Barang - Admin



Gambar 39. Desain *interface* kelola data barang - admin

Pada Gambar 39 merupakan tampilan halaman data barang di admin, tampilan ini akan muncul ketika admin memilih menu data barang. Halaman data barang menampilkan seluruh data barang atau produk yang dijual di aplikasi. Pada halaman ini admin dapat menambah, mengedit, menghapus, dan melihat *detail* produk. Produk yang ditampilkan pada halaman *customer* harus berstatus ‘disetujui’ yang dilakukan oleh *manager*.

The image shows a web application interface for 'Arkatama Store'. On the left is a sidebar menu with the following items: Dashboard, Data Barang, Data Kategori, Setting Hero, Testimoni, Kelola Penjualan, and Laporan Penjualan. The main content area displays a modal window titled 'Tambah Produk/Layanan' with a close button 'x'. The form inside the modal contains the following fields: 'Nama Produk/Layanan' (text input), 'Keterangan' (text area), 'Kategori' (dropdown menu with 'Pilih Kategori'), 'Harga' (text input), 'Stok' (text input), 'Kondisi' (text input), 'Berat' (text input), and 'Gambar' (file upload button labeled 'Choose File'). At the bottom of the modal are two buttons: 'Batal' and 'Simpan'. In the background, a table with an 'Action' column is partially visible, showing buttons for 'tambah', 'edit', and 'hapus'.

Gambar 40. Desain *interface* tambah data barang - admin

Gambar 40 merupakan desain halaman *form* tambah produk di admin. Tampilan ini akan muncul ketika admin menekan tombol ‘tambah layanan’. Admin harus mengisi *form* agar produk dapat ditambahkan. Jika telah mengisi *form* admin dapat menekan tombol ‘simpan’ untuk menyimpan data produk ke *database*.

Arkata Store (nama admin)

ADMIN

- Dashboard
- Data Barang
- Data Kategori
- Setting Hero
- Testimoni
- Kelola Penjualan
- Laporan Penjualan

Edit Data Barang

Nama Produk/Layanan: {nama produk}

Keterangan: {keterangan}

Kategori: {kategori}

Harga: {harga}

Stok: {stok}

Kondisi: {kondisi}

Berat: {berat}

Gambar: Choose File

Kembali Simpan

Gambar 41. Desain *interface* edit data barang - admin

Gambar 41 merupakan desain halaman *form* edit produk di admin. Tampilan ini akan muncul ketika admin menekan tombol 'edit'. Pada halaman ini, admin dapat mengedit data produk. Jika telah mengedit di *form* admin dapat menekan tombol 'simpan' untuk menyimpan data produk ke *database*.

18) Desain *Interface Detail* Produk - Admin

Arkata Store (nama admin)

ADMIN

- Dashboard
- Data Barang
- Data Kategori
- Setting Hero
- Testimoni
- Kelola Penjualan
- Laporan Penjualan

Detail Barang

Detail Produk

Image

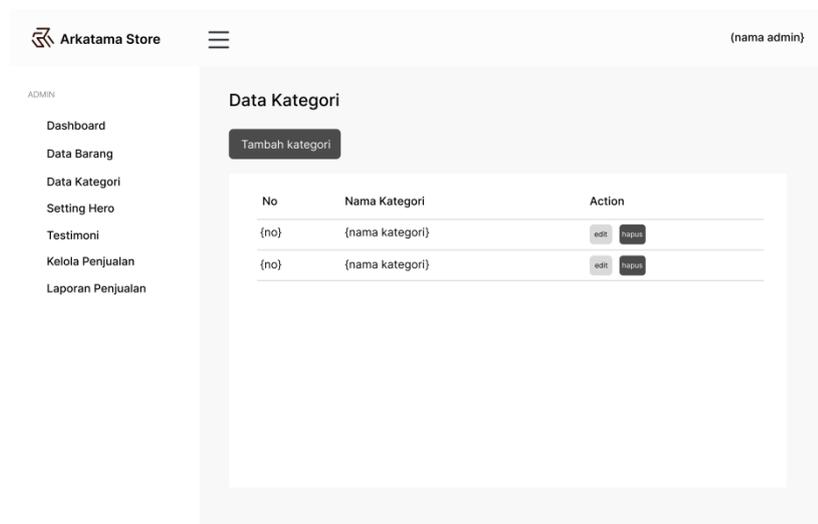
Nama Produk	{nama produk}
Keterangan	{keterangan}
Kategori	{kategori}
Stok	{stok}
Kondisi	{kondisi}
Berat	{berat}
Harga	{harga}
Status	{status}

Kembali

Gambar 42. Desain *interface detail* data barang - admin

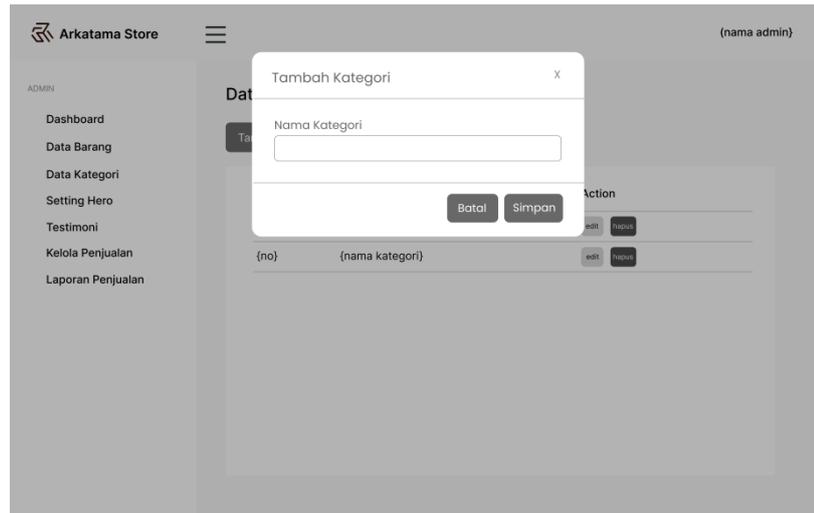
Pada Gambar 42 merupakan tampilan halaman *detail* produk di admin, tampilan ini akan muncul ketika admin memilih menu data barang dan menekan tombol '*detail*'. Halaman *detail* produk menampilkan seluruh informasi dari produk yang dipilih.

19) Desain *Interface* Kelola Data Kategori - Admin



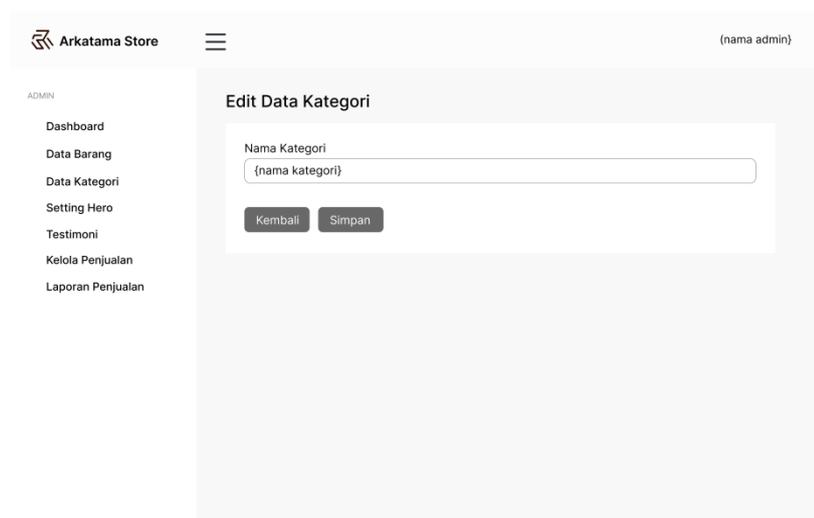
Gambar 43. Desain *interface* kelola data kategori - admin

Pada Gambar 43 merupakan tampilan halaman data kategori di admin, tampilan ini akan muncul ketika admin memilih menu data kategori. Halaman data kategori menampilkan seluruh data kategori. Pada halaman ini admin dapat menambah, mengedit, dan menghapus produk.



Gambar 44. Desain *interface* tambah data kategori - admin

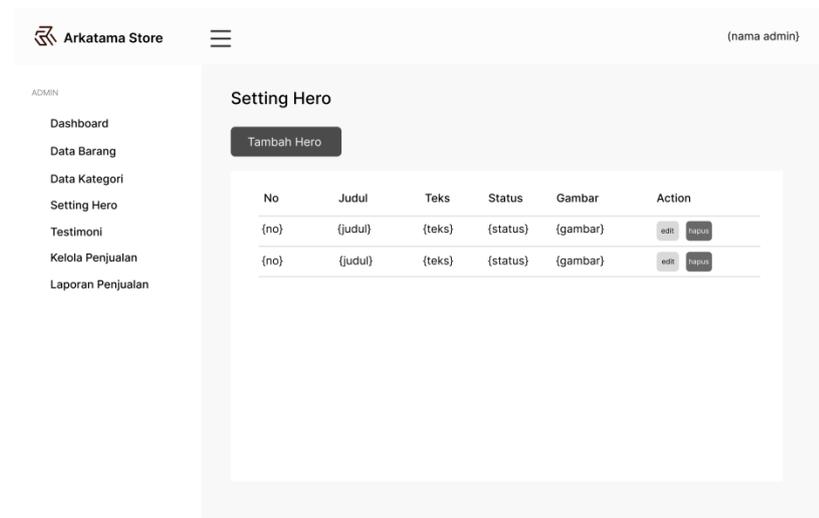
Gambar 44 merupakan desain halaman *form* tambah kategori di admin. Tampilan ini akan muncul ketika admin menekan tombol ‘tambah kategori’. Admin harus mengisi *form* agar kategori dapat ditambahkan. Jika telah mengisi *form* admin dapat menekan tombol ‘simpan’ untuk menyimpan data kategori ke *database*.



Gambar 45. Desain *interface* edit data kategori - admin

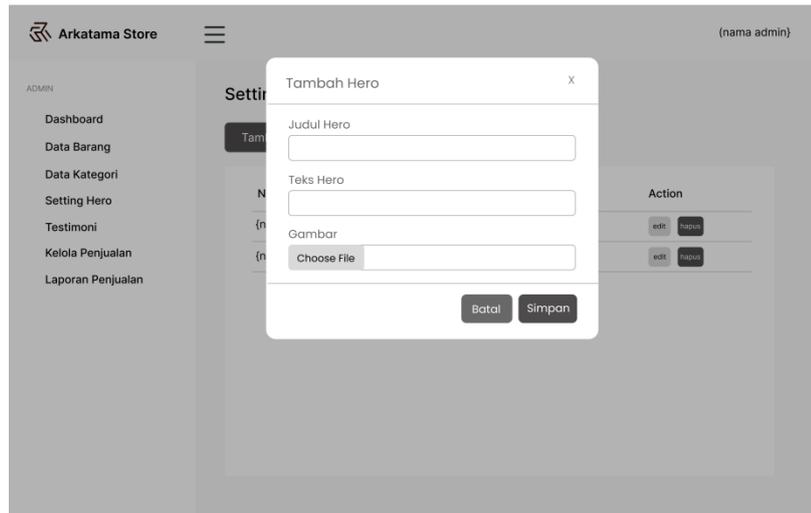
Gambar 45 merupakan desain halaman *form* edit kategori di admin. Tampilan ini akan muncul ketika admin menekan tombol 'edit'. Pada halaman ini, admin dapat mengedit data kategori. Jika telah mengedit di *form* admin dapat menekan tombol 'simpan' untuk menyimpan data kategori ke *database*.

20) Desain *Interface* Kelola Data *Hero* - Admin



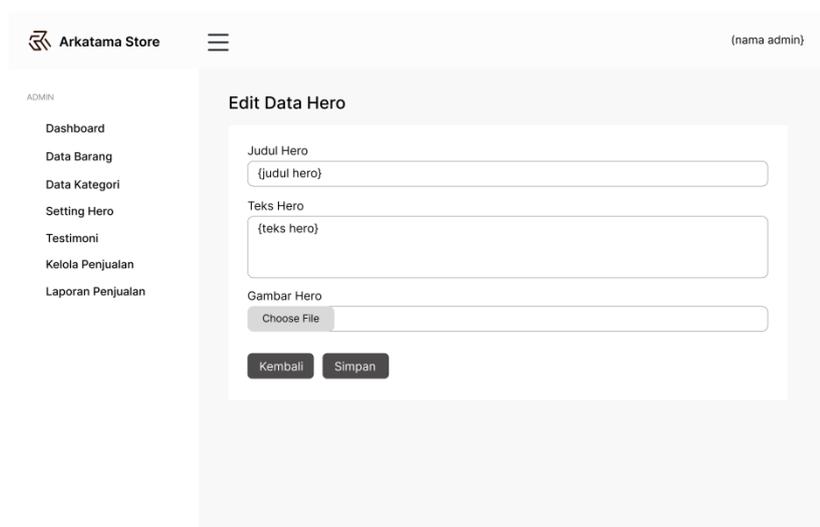
Gambar 46. Desain *interface setting hero* - admin

Pada Gambar 46 merupakan tampilan halaman data *hero* di admin, tampilan ini akan muncul ketika admin memilih menu *setting hero*. Halaman data *hero* menampilkan seluruh data *hero*. Pada halaman ini admin dapat menambah, mengedit, dan menghapus *hero*. *Hero* yang ditampilkan pada halaman *customer* harus berstatus 'disetujui' yang dilakukan oleh *manager*.



Gambar 47. Desain *interface* tambah *hero* - admin

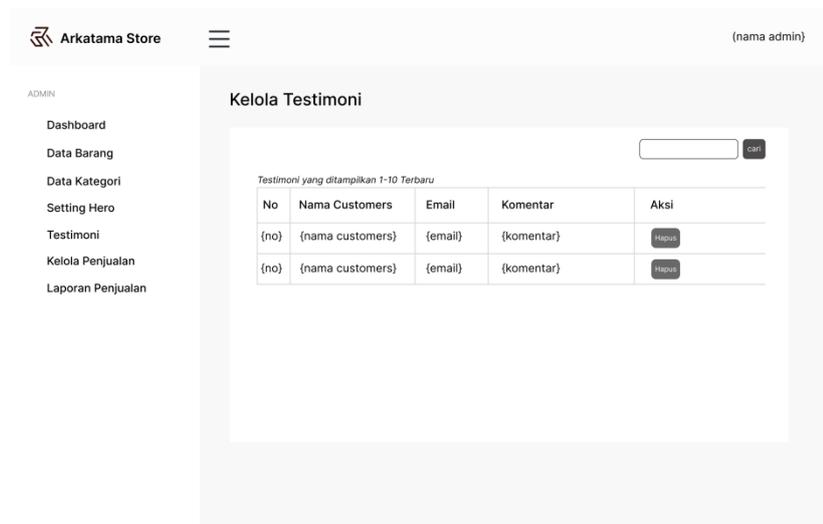
Gambar 47 merupakan desain halaman *form* tambah *hero* di admin. Tampilan ini akan muncul ketika admin menekan tombol ‘tambah *hero*’. Admin harus mengisi *form* agar *hero* dapat ditambahkan. Jika telah mengisi *form* admin dapat menekan tombol ‘simpan’ untuk menyimpan data *hero* ke *database*.



Gambar 48. Desain *interface* edit *hero* - admin

Gambar 48 merupakan desain halaman *form* edit *hero* di admin. Tampilan ini akan muncul ketika admin menekan tombol ‘edit’. Pada halaman ini, admin dapat mengedit data *hero*. Jika telah mengedit di *form* admin dapat menekan tombol ‘simpan’ untuk menyimpan data *hero* ke *database*.

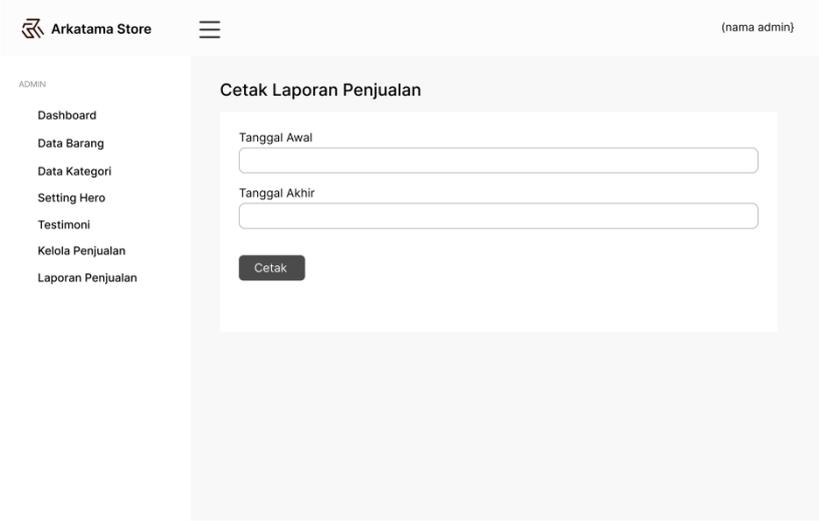
21) Desain *Interface* Data Testimoni - Admin



Gambar 49. Desain *interface* testimoni - admin

Pada Gambar 49 merupakan tampilan halaman data testimoni di admin, tampilan ini akan muncul ketika admin memilih menu kelola testimoni. Halaman data testimoni menampilkan seluruh data testimoni terbaru 1-10. Pada halaman ini admin dapat menghapus testimoni dengan menekan tombol ‘hapus’.

22) Desain *Interface* Cetak Laporan Penjualan - Admin & *manager*

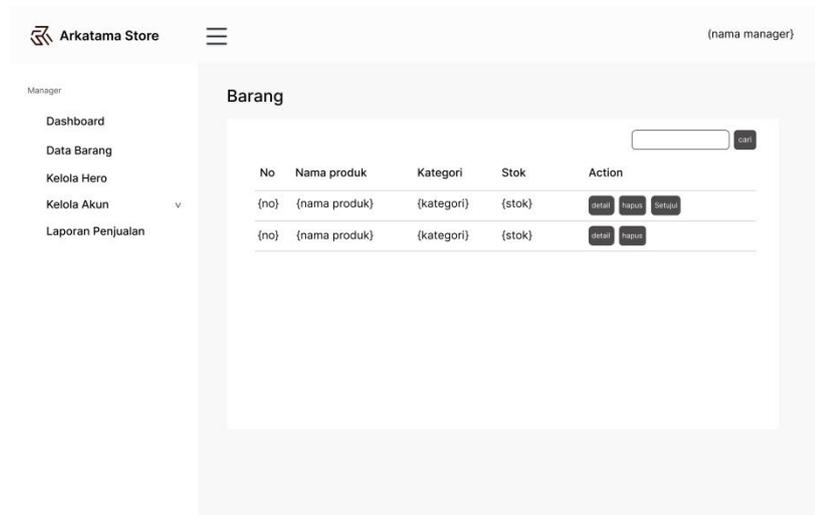


The screenshot displays the admin interface for 'Arkatama Store'. The top navigation bar includes the store logo, a hamburger menu icon, and the user name '(nama admin)'. A sidebar menu on the left lists various administrative options: Dashboard, Data Barang, Data Kategori, Setting Hero, Testimoni, Kelola Penjualan, and Laporan Penjualan. The main content area is titled 'Cetak Laporan Penjualan' and contains a form with two input fields: 'Tanggal Awal' and 'Tanggal Akhir'. Below these fields is a 'Cetak' button.

Gambar 50. Desain *interface* cetak laporan penjualan – admin dan *manager*

Pada Gambar 50 merupakan tampilan halaman laporan penjualan di admin dan *manager*, tampilan ini akan muncul ketika admin atau *manager* memilih menu laporan penjualan. Halaman laporan penjualan menampilkan *form* tanggal yang harus di isi oleh admin atau *manager*. *Form* tanggal ini berguna untuk membatasi data pesanan yang akan ditampilkan dan dicetak.

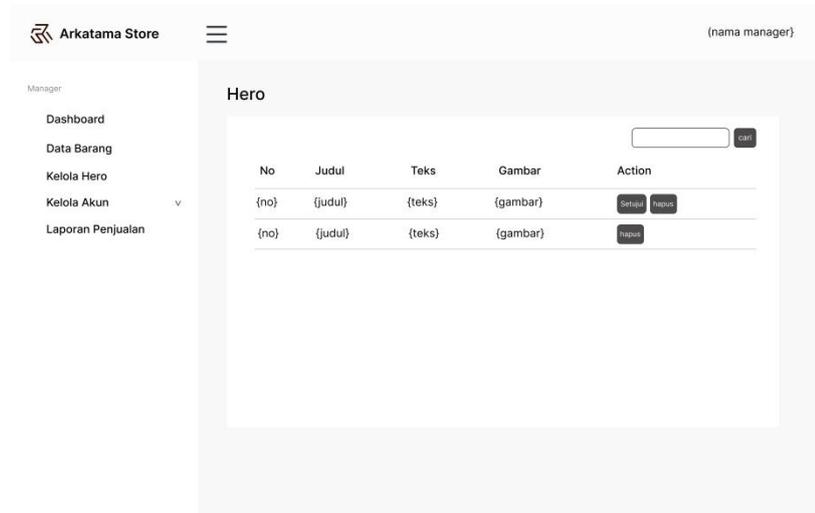
23) Desain *Interface* Data Barang - *Manager*



Gambar 51. Desain *interface* data barang - *manager*

Pada Gambar 51 merupakan tampilan halaman data barang di *manager*, tampilan ini akan muncul ketika *manager* memilih menu data barang. Halaman data barang menampilkan seluruh data barang/produk yang dijual. Pada halaman ini *manager* dapat menyetujui produk, batal menyetujui produk, melihat *detail* produk, dan menghapus produk. Apabila status produk 'disetujui' maka produk ditampilkan di halaman menu produk yang ada di *customer*. Apabila ingin menghapus produk, *manager* dapat menekan tombol 'hapus' maka produk akan terhapus dari *database*, dan jika *manager* ingin melihat *detail* produk dapat dengan menekan tombol 'detail' maka aplikasi menampilkan informasi yang lebih lengkap mengenai produk tersebut.

24) Desain *Interface* Data *Hero* - *Manager*



Gambar 52. Desain *interface* data *hero* - *manager*

Pada Gambar 52 merupakan tampilan halaman data *hero* di *manager*, tampilan ini akan muncul ketika *manager* memilih menu kelola *hero*. Halaman data *hero* menampilkan seluruh data *hero*. Pada halaman ini *manager* dapat menyetujui *hero*, batal menyetujui *hero*, dan menghapus *hero*. Apabila status *hero* 'disetujui' maka *hero* ditampilkan di halaman menu *home* yang ada di *customer*. Apabila ingin menghapus *hero*, *manager* dapat menekan tombol 'hapus' maka *hero* akan terhapus dari *database*.

V. SIMPULAN DAN SARAN

5.1. Simpulan

Berdasarkan hasil dan pembahasan mengenai perancangan dalam pembuatan Aplikasi Arkatama *Store* berbasis *Web* dapat disimpulkan sebagai berikut.

1. Telah berhasil dibuat Aplikasi Arkatama *Store* berbasis *Web*.
2. Aplikasi Arkatama *Store* berbasis *Web* sudah memiliki fungsi dan fitur pengelolaan jual beli produk sesuai yang dibutuhkan.
3. Aplikasi Arkatama *Store* berbasis *Web* yang dibuat telah mendapatkan penilaian yang sangat baik oleh PT Arkatama Multisolusindo.

5.2. Saran

Berdasarkan aplikasi yang telah dibuat terdapat saran terhadap aplikasi yang dikembangkan agar menjadi lebih baik sebagai berikut:

1. Perlu dikembangkannya bagian fitur metode pembayaran *customer* seperti menggunakan fitur *payment gateway* yang variatif agar proses transaksi dapat berjalan secara otomatis, cepat, dan lebih aman.
2. Perlu dikembangkannya bagian fitur pesanan, yaitu dengan menambahkan fitur lacak pesanan agar *customer* dapat mengetahui posisi dari produk yang sedang dikirim.
3. Pada fitur *detail* produk, bagian foto produk perlu dikembangkan menjadi galeri foto produk agar *customer* dapat melihat foto-foto produk lebih banyak dan lebih jelas.

4. Ditambahkan fungsi otomatis untuk batas waktu pembayaran dan pengembalian jumlah stok produk apabila proses pemesanan ditolak oleh admin pada transaksi penjualan agar tidak merugikan penjual dan pembeli lain.
5. Pada menu laporan penjualan, dapat ditambahkan fitur pilihan cetakan laporan berdasarkan data produk-produk yang telah terjual dengan jumlah penjualan terbanyak selama periode tertentu agar admin dapat mengetahui data-data produk apa saja yang telah terjual diaplikasi.

DAFTAR PUSTAKA

- Abdulloh, R. 2015. *Web Programming Is Easy*. Jakarta: PT. Elex Media Komputindo.
- Alatas, H. 2013. *Responsive Web Design dengan PHP & BOOTSTRAP*. Yogyakarta: Lokomedia.
- Erinton, R., Negara, R. M., & Sanjoyo, D. D. 2017. Analisis Performasi Framework Code Igniter Dan Laravel Menggunakan Web Server Apache. *e-Proceeding of Engineering*, 3565-3572.
- Harmianto, Fuad, A., & Khairan, A. 2018. Sistem Informasi Penjualan Online Berbasis Web. *JIKO (Jurnal Informatika dan Komputer)*, 85 - 91.
- Irawan, M. D., & Simargolang, S. A. 2018. Implementasi E-Arsip Pada Program Studi Teknik Informatika. *Jurnal Teknologi Informasi*, 67-84.
- Jamil, M., & Bunyamin. 2015. Pengembangan Aplikasi Sistem Informasi. *Jurnal algoritma* , 358 - 363.
- Josi, A. 2017. Penerapan Metode Prototyping Dalam Pembangunan Website. *JTI*, 50-57.
- Kustiyahningsih, Y., & Anamisa, D. R. 2011. *Pemrograman basis data berbasis WEB menggunakan PHP & MySQL*. Yogyakarta: Graha Ilmu.
- Listianto, F., Fauzi, Irviani, R., Kasmi, & Garaika. 2017. Aplikasi E-Commerce Berbasis Web Mobile Pada Industri. *TAM (Technology Acceptance Model)*, 146-152.
- Prabowo, B. R., Arwani, I., & Pramono, D. 2021. Pengembangan Website Toko Online Baju Bekas. *Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer*, 5413-5420.
- Rosa, & Shalahuddin, M. 2015. *Rekayasa perangkat lunak : terstruktur dan berorientasi objek*. Bandung: Informatika Bandung.
- Simarmata, J., & Paryudi, I. 2006. *Basis data*. Yogyakarta: Andi Offset.

Suharyanto, C. E., Chandra, J. E., & Gunawan, F. E. 2017. Perancangan Sistem Informasi Penggajian Terintegrasi Berbasis Web (Studi Kasus di Rumah Sakit St. Elisabeth). *Jurnal Nasional Teknologi dan Sistem Informasi* , 225 - 232.

Sutarman. 2009. *Pengantar teknologi informasi*. Jakarta: PT Bumi Aksara.

Wahyuni, T., Indriyanti, Ermawati, E., Fatah, H., & Ichsan, N. 2021. Rancang Bangun Sistem Penyewaan Baju Dan Dekorasi Berbasis Web Pada Nita Wedding Organizer. *Jurnal Interkom: Jurnal Publikasi Ilmiah Bidang Teknologi Informasi dan Komunikasi*, 1-9.