

SKRIPSI

PENGEMBANGAN SISTEM WEB API INTEGRATOR MARKETPLACE UNTUK SINKRONISASI STOK PRODUK PADA APOTEK 99

Ditulis untuk memenuhi sebagian persyaratan akademik
guna memperoleh gelar Sarjana Komputer

Oleh:

NAMA : DAVID ANTHONY NATHANAEL

NPM : 01081190015



**PROGRAM STUDI SISTEM INFORMASI
FAKULTAS ILMU KOMPUTER
UNIVERSITAS PELITA HARAPAN
JAKARTA
2023**



PERNYATAAN DAN PERSETUJUAN UNGGAH

TUGAS AKHIR

Saya yang bertanda tangan di bawah ini:

Nama - NPM : David Anthony Nathanael - 01081190015
Fakultas : Ilmu Komputer
Program Studi : Sistem Informasi
Lokasi Kampus : Tangerang
Jenis Tugas Akhir : Skripsi

Dengan ini menyatakan bahwa tugas akhir dengan judul:

“PENGEMBANGAN SISTEM WEB API INTEGRATOR MARKETPLACE UNTUK SINKRONISASI STOK PRODUK PADA APOTEK 99”

Menyatakan bahwa:

1. Tugas akhir tersebut adalah benar karya saya dengan arahan dari dosen pembimbing dan bukan merupakan duplikasi karya tulis yang sudah dipublikasikan atau yang pernah dipakai untuk mendapatkan gelar akademik di perguruan tinggi manapun;
2. Tugas akhir tersebut bukan merupakan plagiat dari hasil karya pihak lain, dan apabila saya mengutip dari karya orang lain maka akan dicantumkan sebagai referensi sesuai dengan ketentuan yang berlaku.
3. Saya memberikan Hak Noneksklusif Tanpa Royalti kepada Universitas Pelita Harapan atas Tugas Akhir tersebut untuk diunggah ke dalam Repositori UPH.

Apabila di kemudian hari ditemukan pelanggaran Hak Cipta dan Kekayaan Intelektual atau Peraturan Perundang-undangan Republik Indonesia lainnya dan integritas akademik dalam karya saya tersebut, maka saya bersedia menanggung secara pribadi segala bentuk tuntutan hukum dan sanksi akademis yang timbul serta membebaskan Universitas Pelita Harapan dari segala tuntutan hukum yang berlaku.

Dibuat di : Tangerang
Pada Tanggal : 3 Februari 2023
Yang menyatakan,

Tanda Tangan	
Nama	(David Anthony Nathanael)



UNIVERSITAS PELITA HARAPAN
FAKULTAS ILMU KOMPUTER

PERSETUJUAN DOSEN PEMBIMBING TUGAS AKHIR

**PENGEMBANGAN SISTEM WEB API INTEGRATOR MARKETPLACE
UNTUK SINKRONISASI STOK PRODUK PADA APOTEK 99**

Oleh :

Nama : David Anthony Nathanael
NPM : 01081190015
Program Studi : Sistem Informasi
Peminatan : Pemrograman Seluler dan Web

Telah diperiksa dan disetujui untuk diajukan dan dipertahankan dalam Sidang Tugas Akhir guna memperoleh gelar Strata Satu Sarjana Komputer (S.Kom.) pada Program Studi Sistem Informasi, Fakultas Ilmu Komputer, Universitas Pelita Harapan, Tangerang, Banten.

Tangerang, 03 Februari 2023

**Menyetujui :
Pembimbing Pertama**

(Hery, S.Kom, M.M.S.I.)

**Ketua Program Studi
Sistem Informasi**

**Dekan
Fakultas Ilmu Komputer**



(Arnold Aribowo, S.T., M.T.)



(Dr. Eng. Ir. Pujianto Yugopuspito, M.Sc.)



UNIVERSITAS PELITA HARAPAN
FAKULTAS ILMU KOMPUTER

PERSETUJUAN TIM PENGUJI TUGAS AKHIR

Pada hari Jumat, 20 Januari 2023 telah diselenggarakan Sidang Tugas Akhir mahasiswa/i dengan keterangan:

Nama : David Anthony Nathanael
NPM : 01081190015
Judul : **PENGEMBANGAN SISTEM WEB API INTEGRATOR MARKETPLACE UNTUK SINKRONISASI STOK PRODUK PADA APOTEK 99**

Sebagai persyaratan akademik untuk mencapai gelar Strata Satu Sarjana Komputer (S.Kom.) pada Program Studi Sistem Informasi, Fakultas Ilmu Komputer, Universitas Pelita Harapan, Tangerang, Banten, yang diuji oleh tim penguji yang terdiri dari:

Nama	Status	Tanda tangan
Hery, S.Kom., M.M.S.I.	sebagai ketua sidang	
Kusno Prasetya, Ph.D.	sebagai anggota 1	
Calandra Alencia Haryani, S.E., S.SI., M.TI.	sebagai anggota 2	

ABSTRAK

David Anthony Nathanael (01081190015)

PENGEMBANGAN SISTEM WEB API INTEGRATOR MARKETPLACE UNTUK SINKRONISASI STOK PRODUK PADA APOTEK 99

(xvi + 185 halaman: 146 gambar, 61 tabel, 5 lampiran)

Apotek 99 adalah bisnis apotek di Bogor sejak 2001 yang menjual berbagai obat dan produk medis. Saat ini, Apotek 99 sudah menggunakan sistem dalam mengelola produk dan transaksi, namun belum terintegrasi dengan *marketplace*. Hal ini membuat pemilik dan karyawan harus memeriksa dan memperbarui data stok produk secara manual. Permasalahan mulai timbul, ketika pihak apotek tidak selalu memeriksa dan memperbarui stok produk, yang dapat mengakibatkan stok produk di *marketplace* menjadi tidak *up-to-date*, dan berisiko terjadinya *overselling*, yang dapat merugikan apotek, seperti pembatalan pesanan atau pengeluaran usaha lebih dalam pencarian produk. Selain itu, apotek kesulitan dalam melacak dan menelusuri histori *expired* produk, karena sistem apotek belum dapat membedakan tanggal *expired*, sehingga dapat merugikan finansial akibat barang *expired* yang belum terjual.

Untuk menjawab permasalahan, dilakukan penelitian tugas akhir untuk mengembangkan *web API integrator* yang dapat membantu Apotek 99 dalam mengelola data produk, dan mengelompokkan produk berdasarkan tanggal *expired*, dan mempermudah proses sinkronisasi data stok produk secara *up-to-date* pada konvensional dan *marketplace*. Sistem yang diusulkan dikembangkan dengan metode *prototyping*, menggunakan *Unified Modelling Language* (UML) yang meliputi *use case*, *activity diagram*, *class diagram* dan *table relationship diagram* sebagai rancangan pemodelan sistem, sistem dibangun dengan bahasa *PHP* sebagai bahasa pemrograman, dan *framework CodeIgniter 3*, database *MySQL* dalam *PHPMyAdmin*, serta pembaruan stok produk menggunakan *API SmartSeller*.

Tugas akhir ini dapat menghasilkan sebuah sistem *web API integrator*, untuk mempermudah proses pembaruan data stok produk pada Apotek 99, mengelompokkan produk berdasarkan tanggal *expired*, dan pembuatan transaksi pada konvensional, yang akan mempengaruhi stok produk pada konvensional dan *marketplace*.

Kata kunci: Sistem Informasi, *PHP*, *CodeIgniter*, apotek, *web API integrator*.

Referensi: 21 (2010 – 2022)

ABSTRACT

David Anthony Nathanael (01081190015)

PENGEMBANGAN SISTEM WEB API INTEGRATOR MARKETPLACE UNTUK SINKRONISASI STOK PRODUK PADA APOTEK 99

(xvi + 185 pages: 146 figures, 61 tables, 5 appendices)

Apotek 99 is pharmacy store in Bogor since 2001 that sells various kinds of medicine products. Currently, the store has using system for manage products and transactions but didn't integrate with marketplace. This makes store must renewing product stock manually. The problems arise when store didn't always check and update product stock, which resulting an outdated stock in marketplace, which cause risk of overselling, which can inflict losses, such as cancelling orders or spending more effort in products searching. Store also having difficulties of tracking and tracing the product's expired, because their own system cannot distinguish between expiration dates, resulting financial losses due to unsold expired products.

To address the problems, a final project research was conducted to develop a web API integrator that can assist store in managing product data, and grouping products based on expiration date, and simplifying the process of synchronizing up-to-date product stock data on conventional and marketplaces. The proposed system was developed using the prototyping method and the system modeling is designed using the Unified Modeling Language (UML) which includes use cases, activity diagrams, class diagrams, and table relationship diagrams, the programming language used for system is PHP and CodeIgniter 3 framework, uses MySQL in PHPMyAdmin as database, as well as product stock updates with *SmartSeller API*.

This final project can produce an API integrator web system, to simplify the process of updating product stock data, grouping products based on expiration dates, and making transactions on conventional, which will affect stock on conventional and marketplaces.

Keywords: *Information System, PHP, CodeIgniter, pharmacy store, web API integrator.*

References: 21 (2010 – 2022)

KATA PENGANTAR

Segala puji dan syukur atas kehadiran Tuhan Yang Maha Esa dan juga hikmatNya yang senantiasa diberikan kepada penulis, sehingga penulis dapat menyelesaikan tugas akhir dengan judul "PENGEMBANGAN SISTEM WEB API INTEGRATOR MARKETPLACE UNTUK SINKRONISASI STOK PRODUK PADA APOTEK 99" sebagai syarat dalam menyelesaikan Program Studi Sistem Informasi (S1) dalam Program Sarjana Fakultas Ilmu Komputer Universitas Pelita Harapan.

Dalam penyusunan tugas akhir ini, banyak sekali hambatan dan rintangan yang penulis hadapi, namun akhirnya penulis dapat melalui hal ini berkat bantuan dan bimbingan dari berbagai pihak, baik secara moral maupun spiritual. Untuk itu, penulis menyampaikan ucapan terima kasih kepada:

1. Dr. Eng. Ir. Pujiyanto Yugospito, M.Sc. selaku Dekan Fakultas Ilmu Komputer Universitas Pelita Harapan.
2. Arnold Aribowo, S.T., M.T. selaku Ketua Program Studi Sistem Informasi Universitas Pelita Harapan.
3. Hery, S.Kom., M.M.S.I. selaku Dosen Pembimbing yang sudah berkenan meluangkan waktunya demi memberikan bimbingan dan arahan selama penyusunan tugas akhir.
4. Seluruh Dosen dan Staf Fakultas Ilmu Komputer Universitas Pelita Harapan.
5. Pemilik dan seluruh staf Apotek 99 yang telah memberikan izin penelitian dan juga membantu kelancaran penulis dalam penelitian ini.
6. Seluruh pihak yang tidak bisa disebutkan satu per satu yang sudah membantu memberikan dukungan.

Penulis meminta maaf atas seluruh kesalahan yang terjadi selama penelitian. Semoga tugas akhir ini dapat memberikan manfaat untuk penelitian-penelitian berikutnya.

Tangerang, 3 Februari 2023

Penulis

DAFTAR ISI

PERNYATAAN DAN PERSETUJUAN UNGGAH TUGAS AKHIR.....	ii
PERSETUJUAN DOSEN PEMBIMBING TUGAS AKHIR	iii
PERSETUJUAN TIM PENGUJI TUGAS AKHIR.....	iv
ABSTRAK	v
ABSTRACT	vi
KATA PENGANTAR.....	vii
DAFTAR ISI.....	viii
DAFTAR GAMBAR.....	xi
DAFTAR TABEL.....	xiv
DAFTAR LAMPIRAN	xvi
BAB I PENDAHULUAN.....	1
1.1 Latar Belakang	1
1.2 Perumusan Masalah	2
1.3 Pembatasan Masalah.....	2
1.4 Tujuan dan Manfaat Penelitian	3
1.4.1 Tujuan Penelitian	3
1.4.2 Manfaat Penelitian	3
1.5 Metodologi Penelitian.....	3
1.5.1 Metode Pengumpulan Data.....	3
1.5.2 Metodologi Pengembangan Sistem.....	4
1.6 Sistematika Penulisan.....	5
BAB II LANDASAN TEORI	7
2.1 <i>Website</i>	7
2.2 <i>Application Programming Interface (API)</i>	7
2.3 <i>API Integrator Marketplace</i>	7
2.4 <i>HyperText Markup Language (HTML)</i>	7
2.5 <i>Cascading Style Sheets (CSS)</i>	8
2.6 <i>HyperText Processor (PHP)</i>	8
2.7 <i>Javascript</i>	8
2.8 <i>CodeIgniter</i>	9
2.9 <i>Bootstrap</i>	9
2.10 <i>Marketplace</i>	9
2.10.1 Tokopedia.....	10
2.10.2 <i>Shopee</i>	11

2.11 <i>SmartSeller</i>	12
2.11.1 <i>SmartSeller API</i>	12
2.12 <i>Database Management System (DBMS)</i>	14
2.12.1 <i>Relational Database Management System (rDBMS)</i>	14
2.12.2 <i>MySQL</i>	15
2.13 <i>Software Development Life Cycle (SDLC)</i>	15
2.14 <i>Prototyping</i>	16
2.15 <i>Unified Modeling Language (UML)</i>	16
2.15.1 <i>Use Case Diagram</i>	16
2.15.2 <i>Activity Diagram</i>	18
2.15.3 <i>Class Diagram</i>	19
2.16 <i>Penelitian Terdahulu</i>	21
BAB III SISTEM SAAT INI	24
3.1 <i>Profil Perusahaan</i>	24
3.1.1 <i>Visi Perusahaan</i>	24
3.1.2 <i>Misi Perusahaan</i>	24
3.2 <i>Struktur Organisasi</i>	25
3.3 <i>Analisis Sistem Saat ini</i>	25
3.3.1 <i>Kendala Sistem Saat ini</i>	31
BAB IV ANALISIS DAN PERANCANGAN SISTEM USULAN	33
4.1 <i>Analisis Kelayakan Sistem</i>	33
4.1.1 <i>Kelayakan Teknis</i>	33
4.1.2 <i>Kelayakan Ekonomi</i>	34
4.1.3 <i>Kelayakan Organisasi</i>	36
4.2 <i>Tahap Analisis</i>	36
4.2.1 <i>Functional Requirement</i>	36
4.2.2 <i>Non-functional Requirement</i>	39
4.3 <i>Sistem Usulan</i>	40
4.3.1 <i>Pemodelan Proses Bisnis</i>	40
4.3.2 <i>Pemodelan Struktural</i>	66
4.3.3 <i>Perancangan Manajemen Data</i>	68
4.3.4 <i>Mockup Lapisan Antarmuka</i>	79
4.4 <i>Implementasi Sistem</i>	86
4.4.1 <i>Tahap Pemrograman</i>	86
4.4.2 <i>Perancangan Lapisan Antarmuka</i>	86
4.5 <i>Pengujian Sistem Usulan</i>	132
BAB V KESIMPULAN DAN SARAN	134

5.1 Kesimpulan	134
5.2 Saran.....	134
DAFTAR PUSTAKA	136
LAMPIRAN A HASIL WAWANCARA	A-1
LAMPIRAN B FORM LEMBAR MONITORING BIMBINGAN TUGAS AKHIR	A-8
LAMPIRAN C LINK VIDEO DEMO	A-11
LAMPIRAN D TAHAP PENGUJIAN.....	A-12
LAMPIRAN E <i>USE CASE DESCRIPTION</i>	A-29

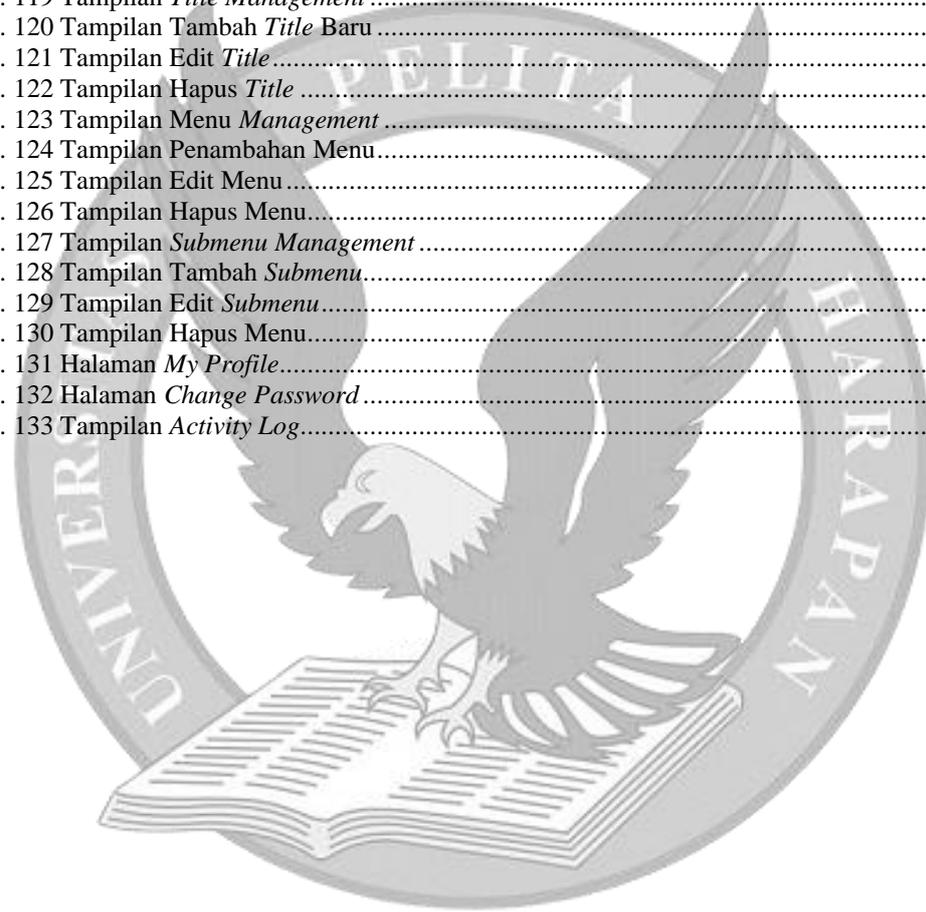


DAFTAR GAMBAR

Gambar 2. 1 Tahap-Tahap <i>Software Development Life Cycle</i> (SDLC)	15
Gambar 2. 2 Metodologi <i>Prototyping</i>	16
Gambar 3. 1 Struktur Organisasi Apotek 99.....	25
Gambar 3. 2 <i>Interface</i> Admin	25
Gambar 3. 3 <i>Interface</i> Kasir.....	26
Gambar 3. 4 Form Penjualan	26
Gambar 3. 5 <i>Activity Diagram</i> Form Pemesanan Konvensional	27
Gambar 3. 6 Form Master Obat	28
Gambar 3. 7 Form Isi Obat Baru.....	28
Gambar 3. 8 Pemilihan Obat.....	29
Gambar 3. 9 Perubahan Data Obat.....	29
Gambar 3. 10 Pemilihan Tanggal Laporan Penjualan.....	30
Gambar 3. 11 Data Laporan Penjualan.....	30
Gambar 4. 1 <i>Use Case Diagram</i> Sistem <i>Web API Integrator</i>	41
Gambar 4. 2 <i>Activity Diagram</i> Tambah dan Hapus Keranjang Transaksi Toko	43
Gambar 4. 3 <i>Activity Diagram</i> Ubah Keranjang Transaksi Toko	44
Gambar 4. 4 <i>Activity Diagram</i> Menyimpan Transaksi Toko.....	44
Gambar 4. 5 <i>Activity Diagram</i> Menghapus Transaksi Toko	45
Gambar 4. 6 <i>Activity Diagram</i> Mengelola Profil.....	47
Gambar 4. 7 <i>Activity Diagram</i> Mengelola Informasi Produk.....	49
Gambar 4. 8 <i>Activity Diagram</i> Mengelola <i>Batch</i> Produk.....	51
Gambar 4. 9 <i>Activity Diagram</i> Mengelola Akses Menu.....	52
Gambar 4. 10 <i>Activity Diagram</i> Mengelola Akses <i>Backdoor</i>	53
Gambar 4. 11 <i>Activity Diagram</i> Mengelola Akses Menu.....	54
Gambar 4. 12 <i>Activity Diagram</i> Mengelola Kategori Produk	55
Gambar 4. 13 <i>Activity Diagram</i> Mengelola Unit Produk	57
Gambar 4. 14 <i>Activity Diagram</i> Mencari Produk di Dashboard	58
Gambar 4. 15 <i>Activity Diagram</i> Mengakses Laporan Laba.....	59
Gambar 4. 16 <i>Activity Diagram</i> Menambah, Mengubah, dan Menghapus <i>API Key</i>	60
Gambar 4. 17 <i>Activity Diagram</i> Mengelola Aktivasi <i>API Key</i>	61
Gambar 4. 18 <i>Activity Diagram</i> Tambah, Edit, Hapus <i>User</i>	63
Gambar 4. 19 <i>Activity Diagram</i> Ubah <i>Password</i> dan Nonaktifkan <i>User</i>	64
Gambar 4. 20 <i>Activity Diagram</i> Mengelola <i>Role</i>	65
Gambar 4. 21 <i>Class Diagram</i>	67
Gambar 4. 22 <i>Table Relationship Diagram</i> Sistem <i>Web API Integrator</i>	69
Gambar 4. 23 Rancangan <i>Mockup Login</i>	79
Gambar 4. 24 Rancangan Lapisan Antarmuka Pengelolaan <i>User</i>	79
Gambar 4. 25 Rancangan Lapisan Antarmuka <i>Register User</i>	80
Gambar 4. 26 Rancangan Lapisan Antarmuka <i>Ubah Password</i>	80
Gambar 4. 27 Gambar Lapisan Antarmuka Mengelola <i>Role</i>	81
Gambar 4. 28 Gambar Lapisan Antarmuka Mengelola Produk.....	81
Gambar 4. 29 Gambar Lapisan Antarmuka Mengelola Kategori Produk	82
Gambar 4. 30 Gambar Lapisan Antarmuka Mengelola Unit Produk.....	82
Gambar 4. 31 Gambar Lapisan Antarmuka Mengelola <i>Batch</i> Produk.....	83
Gambar 4. 32 Gambar Lapisan Antarmuka Mengelola Menu.....	83
Gambar 4. 33 Gambar Lapisan Antarmuka Mengelola <i>Submenu</i>	84
Gambar 4. 34 Gambar Lapisan Antarmuka Menambah Transaksi Toko.....	84
Gambar 4. 35 Gambar Lapisan Antarmuka Aktivitas Transaksi	85
Gambar 4. 36 Gambar Lapisan Antarmuka Laporan Penjualan	85
Gambar 4. 37 Tampilan <i>Log In</i>	87
Gambar 4. 38 Tampilan <i>Log In</i> jika <i>User</i> Salah Memasukkan Email dan/atau <i>Password</i>	87
Gambar 4. 39 Tampilan Buka <i>Register</i> jika Halaman Dinonaktifkan Admin.....	87
Gambar 4. 40 Tampilan Register dengan Input Data Lengkap.....	88
Gambar 4. 41 Tampilan Registrasi Berhasil	88
Gambar 4. 42 Email Masuk dengan Link Aktivasi	89
Gambar 4. 43 Pesan Akun Berhasil Diaktifkan	89

Gambar 4. 44 Tampilan Buka <i>Forgot Password</i> jika Dinonaktifkan Admin.....	89
Gambar 4. 45 Tampilan <i>Forgot Password</i> dengan Data Email Aktif.....	90
Gambar 4. 46 Tampilan <i>Forgot Password</i> Berhasil.....	90
Gambar 4. 47 Email Masuk dengan Link <i>Reset Password</i>	91
Gambar 4. 48 Tampilan <i>Reset Password</i>	91
Gambar 4. 49 Tampilan <i>Reset Password</i> Berhasil.....	91
Gambar 4. 50 Tampilan <i>Dashboard</i>	92
Gambar 4. 51 Tampilan Cari Produk dalam <i>Dashboard</i>	92
Gambar 4. 52 Tampilan Informasi Produk.....	93
Gambar 4. 53 Tampilan Produk Ditambah ke Transaksi Toko dari <i>Dashboard</i>	93
Gambar 4. 54 Tampilkan Kelola Produk.....	94
Gambar 4. 55 Tampilan Tambah Produk.....	94
Gambar 4. 56 Tampilan Pesan Sukses Penambahan Produk.....	95
Gambar 4. 57 Tampilan Informasi Produk.....	95
Gambar 4. 58 Tampilan Edit Informasi Produk.....	96
Gambar 4. 59 Tampilan Hapus Produk.....	96
Gambar 4. 60 Tampilan <i>Batch</i> Produk dalam Jendela <i>Pop-Up</i>	97
Gambar 4. 61 Tampilan Semua <i>Batch</i> dalam Satu Produk.....	97
Gambar 4. 62 Tampilan Tambah <i>Batch</i>	98
Gambar 4. 63 Pesan Sukses <i>Batch</i> Produk Berhasil Dibuat.....	98
Gambar 4. 64 Pengeditan Data <i>Batch</i> Produk.....	99
Gambar 4. 65 Pengeditan Data Stok <i>Batch</i> Produk.....	99
Gambar 4. 66 Tampilan Halaman Produk Pada <i>Shopee</i>	100
Gambar 4. 67 Tampilan Halaman Produk Pada Tokopedia.....	100
Gambar 4. 68 Konfirmasi Mengosongkan Stok Produk.....	100
Gambar 4. 69 Tampilan Kelola Kategori Produk.....	101
Gambar 4. 70 Tampilan Pesan Berhasil Impor Kategori dari <i>SmartSeller API</i>	101
Gambar 4. 71 Tampilan Tambah Kategori.....	102
Gambar 4. 72 Tampilan Edit Kategori.....	102
Gambar 4. 73 Tampilan Hapus Kategori.....	102
Gambar 4. 74 Tampilan Kelola Unit Produk.....	103
Gambar 4. 75 Tampilan Tambah Unit Produk.....	103
Gambar 4. 76 Tampilan Edit Unit Produk.....	104
Gambar 4. 77 Tampilan Hapus Unit Produk.....	104
Gambar 4. 78 Tampilan Produk <i>Expired</i> / Hampir <i>Expired</i>	105
Gambar 4. 79 Halaman Konfirmasi Pengosongan Stok <i>Batch</i> Produk.....	105
Gambar 4. 80 Tampilan Produk dengan <i>Batch</i> Kosong.....	106
Gambar 4. 81 Tampilan Form Penambahan <i>Batch</i> Produk.....	106
Gambar 4. 82 Tampilan Tambah Transaksi Toko.....	107
Gambar 4. 83 Tampilan <i>Dropdown</i> Pencarian Produk Dalam Transaksi.....	108
Gambar 4. 84 Tampilan Produk Berhasil Ditambah ke Keranjang Transaksi.....	108
Gambar 4. 85 Tampilan Keranjang Transaksi Berhasil Dihapus.....	108
Gambar 4. 86 Tampilan Ubah Keranjang Transaksi Toko.....	109
Gambar 4. 87 Tampilan Isi Kolom Input pada Transaksi Toko.....	109
Gambar 4. 88 Tampilan Pesan <i>Error</i> pada Penyimpanan Transaksi.....	110
Gambar 4. 89 Tampilan Pesan Sukses pada Penyimpanan Transaksi.....	110
Gambar 4. 90 Tampilan <i>History</i> Transaksi.....	111
Gambar 4. 91 Tampilan <i>Invoice</i> Transaksi.....	111
Gambar 4. 92 Tampilan Pesan Konfirmasi Hapus Transaksi.....	112
Gambar 4. 93 Tampilan Pesan Berhasil Hapus Transaksi.....	112
Gambar 4. 94 Halaman Stok Produk Bertambah setelah Hapus Transaksi.....	112
Gambar 4. 95 Grafik dan Laporan Penjualan dan Pembelian.....	113
Gambar 4. 96 Pemilihan Tahun Laporan Penjualan dan Pembelian.....	114
Gambar 4. 97 Laporan Penjualan dan Pembelian.....	114
Gambar 4. 98 Tampilan <i>User</i>	115
Gambar 4. 99 Tampilan Rincian <i>User</i>	115
Gambar 4. 100 Tampilan Tambah <i>User</i>	116
Gambar 4. 101 Tampilan Edit <i>User</i>	116
Gambar 4. 102 Tampilan Input Password Baru untuk <i>User</i>	117
Gambar 4. 103 Tampilan Hapus Akun <i>User</i>	117

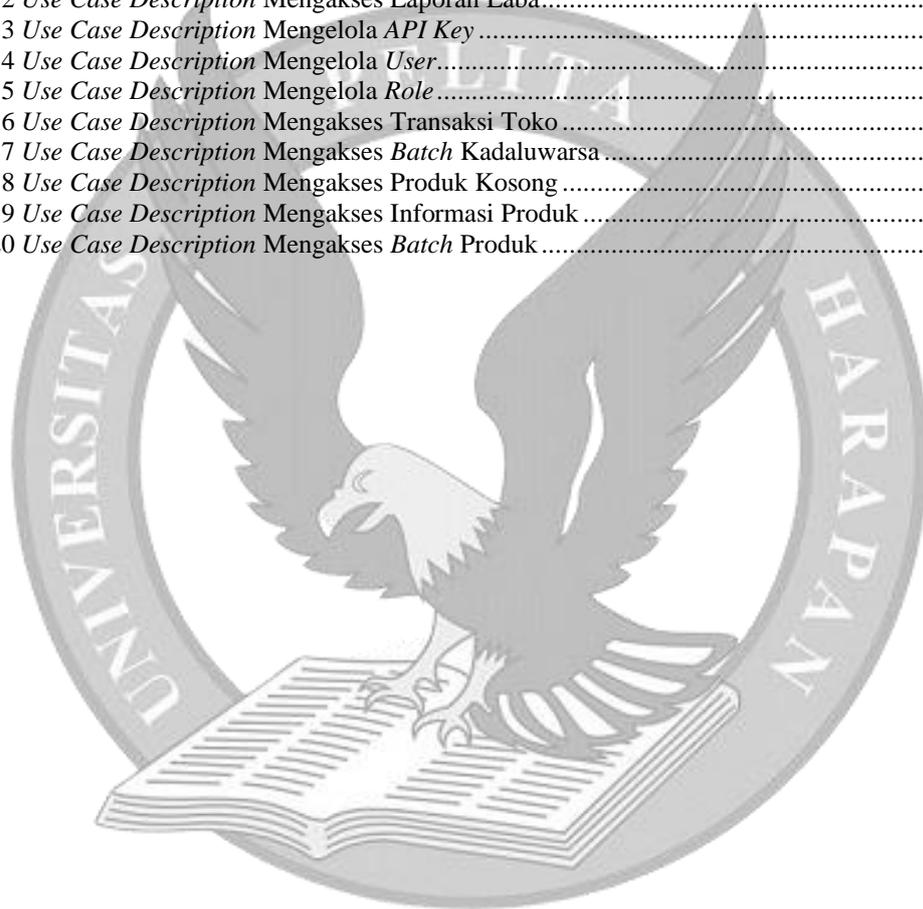
Gambar 4. 104 Tampilan <i>System Setting</i>	117
Gambar 4. 105 Tampilan <i>Login</i> Saat Fitur Aktif	118
Gambar 4. 106 Tampilan Fitur Registrasi saat Dinonaktifkan Admin	118
Gambar 4. 107 Tampilan Fitur Lupa <i>Password</i> saat Dinonaktifkan Admin	119
Gambar 4. 108 Tampilan Kelola <i>API Key</i>	119
Gambar 4. 109 Referensi SmartSeller untuk Mendapatkan <i>API Key</i>	120
Gambar 4. 110 Tampilan <i>API Key</i> Berhasil Ditambahkan	120
Gambar 4. 111 Tampilan Pesan <i>API Key</i> Gagal Diedit.....	121
Gambar 4. 112 Tampilan <i>Role Management</i>	121
Gambar 4. 113 Tampilan Penambahan Data <i>Role</i>	122
Gambar 4. 114 Tampilan Pengeditan Data <i>Role</i>	122
Gambar 4. 115 Tampilan Hapus <i>Role</i>	122
Gambar 4. 116 Tampilan <i>Role Access</i>	123
Gambar 4. 117 Tampilan Pesan Sukses Penambahan <i>Title</i> pada <i>Role</i>	123
Gambar 4. 118 Tampilan <i>Blocked</i> jika <i>User</i> Tidak Mendapat Akses	124
Gambar 4. 119 Tampilan <i>Title Management</i>	125
Gambar 4. 120 Tampilan Tambah <i>Title</i> Baru	125
Gambar 4. 121 Tampilan Edit <i>Title</i>	126
Gambar 4. 122 Tampilan Hapus <i>Title</i>	126
Gambar 4. 123 Tampilan Menu <i>Management</i>	127
Gambar 4. 124 Tampilan Penambahan Menu.....	127
Gambar 4. 125 Tampilan Edit Menu	128
Gambar 4. 126 Tampilan Hapus Menu.....	128
Gambar 4. 127 Tampilan <i>Submenu Management</i>	129
Gambar 4. 128 Tampilan Tambah <i>Submenu</i>	129
Gambar 4. 129 Tampilan Edit <i>Submenu</i>	130
Gambar 4. 130 Tampilan Hapus Menu.....	130
Gambar 4. 131 Halaman <i>My Profile</i>	131
Gambar 4. 132 Halaman <i>Change Password</i>	131
Gambar 4. 133 Tampilan <i>Activity Log</i>	132



DAFTAR TABEL

Tabel 2. 1 <i>Syntax Use Case Diagram</i>	17
Tabel 2. 1 <i>Syntax Use Case Diagram</i> (Lanjutan).....	18
Tabel 2. 2 <i>Syntax Activity Diagram</i>	18
Tabel 2. 2 <i>Syntax Activity Diagram</i> (Lanjutan)	19
Tabel 2. 3 <i>Syntax Class Diagram</i>	20
Tabel 2. 4 Penelitian Terdahulu Mengenai Pengembangan Sistem <i>Web API Integrator</i>	22
Tabel 2. 4 Penelitian Terdahulu Mengenai Pengembangan Sistem <i>Web API Integrator</i> (Lanjutan)	23
Tabel 4. 1 Spesifikasi Minimum Perangkat Sistem Usulan untuk Pengembang	35
Tabel 4. 2 Spesifikasi Minimum Perangkat Sistem Usulan untuk <i>Server</i>	35
Tabel 4. 3 <i>Table Description</i> pada tabel <i>User</i>	70
Tabel 4. 4 <i>Table Description</i> pada tabel <i>User_Role</i>	70
Tabel 4. 5 <i>Table Description</i> pada tabel <i>Access_Menu</i>	71
Tabel 4. 6 <i>Table Description</i> pada tabel <i>Access_Sub</i>	71
Tabel 4. 7 <i>Table Description</i> pada tabel <i>Menu</i>	71
Tabel 4. 7 <i>Table Description</i> pada tabel <i>Menu</i> (Lanjutan).....	72
Tabel 4. 8 <i>Table Description</i> pada tabel <i>Sub_Menu</i>	72
Tabel 4. 9 <i>Table Description</i> pada tabel <i>Cart</i>	72
Tabel 4. 9 <i>Table Description</i> pada tabel <i>Cart</i> (Lanjutan).....	73
Tabel 4. 10 <i>Table Description</i> pada tabel <i>Sales</i>	73
Tabel 4. 11 <i>Table Description</i> pada tabel <i>Sales_Detail</i>	73
Tabel 4. 11 <i>Table Description</i> pada tabel <i>Sales_Detail</i> (Lanjutan)	74
Tabel 4. 12 <i>Table Description</i> pada tabel <i>Product</i>	74
Tabel 4. 13 <i>Table Description</i> pada tabel <i>Product_Category</i>	75
Tabel 4. 14 <i>Table Description</i> pada tabel <i>Product_Batch</i>	75
Tabel 4. 14 <i>Table Description</i> pada tabel <i>Product_Batch</i> (Lanjutan).....	76
Tabel 4. 15 <i>Table Description</i> pada tabel <i>Product_Unit</i>	76
Tabel 4. 16 <i>Table Description</i> pada tabel <i>Month</i>	76
Tabel 4. 17 <i>Table Description</i> pada tabel <i>Setting</i>	76
Tabel 4. 17 <i>Table Description</i> pada tabel <i>Setting</i> (Lanjutan)	77
Tabel 4. 18 <i>Table Description</i> pada tabel <i>Login_History</i>	77
Tabel 4. 19 <i>Table Description</i> pada tabel <i>User_Token</i>	77
Tabel 4. 20 <i>Table Description</i> pada tabel <i>Title</i>	78
Tabel 4. 21 <i>Table Description</i> pada tabel <i>Access_Title</i>	78
Tabel 4. 22 <i>Table Description</i> pada tabel <i>Api</i>	78
Tabel 4. 23 <i>Test Case</i> Fitur Mengelola Informasi Produk oleh Admin sebagai Aktor.....	132
Tabel 4. 23 <i>Test Case</i> Fitur Mengelola Informasi Produk oleh Admin sebagai Aktor (Lanjutan).....	133
Tabel D. 1 <i>Test Case</i> Fitur Mengelola Informasi Produk oleh Admin sebagai Aktor	A-12
Tabel D. 1 <i>Test Case</i> Fitur Mengelola Informasi Produk oleh Admin sebagai Aktor (Lanjutan).....	A-13
Tabel D. 2 <i>Test Case</i> Fitur Mengelola <i>Batch</i> Produk oleh Admin sebagai Aktor	A-13
Tabel D. 2 <i>Test Case</i> Fitur Mengelola <i>Batch</i> Produk oleh Admin sebagai Aktor (Lanjutan).....	A-14
Tabel D. 3 <i>Test Case</i> Fitur Mengelola Kategori Produk oleh Admin sebagai Aktor.....	A-14
Tabel D. 3 <i>Test Case</i> Fitur Mengelola Kategori Produk oleh Admin sebagai Aktor (Lanjutan)	A-15
Tabel D. 4 <i>Test Case</i> Fitur Mengelola Unit Produk oleh Admin sebagai Aktor	A-15
Tabel D. 4 <i>Test Case</i> Fitur Mengelola Unit Produk oleh Admin sebagai Aktor (Lanjutan).....	A-16
Tabel D. 5 <i>Test Case</i> Fitur Mengelola Laporan oleh Admin sebagai Aktor	A-17
Tabel D. 6 <i>Test Case</i> Fitur Mengelola <i>User</i> oleh Admin sebagai Aktor.....	A-17
Tabel D. 6 <i>Test Case</i> Fitur Mengelola <i>User</i> oleh Admin sebagai Aktor (Lanjutan)	A-18
Tabel D. 7 <i>Test Case</i> Fitur Mengelola <i>Role</i> oleh Admin sebagai Aktor.....	A-19
Tabel D. 8 <i>Test Case</i> Fitur Mengelola <i>Menu</i> oleh Admin sebagai Aktor	A-20
Tabel D. 8 <i>Test Case</i> Fitur Mengelola <i>Menu</i> oleh Admin sebagai Aktor (Lanjutan).....	A-21
Tabel D. 9 <i>Test Case</i> Fitur Mengelola <i>Submenu</i> oleh Admin sebagai Aktor.....	A-21
Tabel D. 9 <i>Test Case</i> Fitur Mengelola <i>Submenu</i> oleh Admin sebagai Aktor (Lanjutan)	A-22
Tabel D. 10 <i>Test Case</i> Fitur Mengelola Profil Oleh Admin Sebagai Aktor.....	A-23
Tabel D. 10 <i>Test Case</i> Fitur Mengelola Profil Oleh Admin Sebagai Aktor (Lanjutan).....	A-24
Tabel D. 11 <i>Test Case</i> Fitur Menambah Transaksi Toko oleh Admin sebagai Aktor	A-24
Tabel D. 11 <i>Test Case</i> Fitur Menambah Transaksi Toko oleh Admin sebagai Aktor (Lanjutan)	A-25

Tabel D. 12 <i>Test Case</i> Fitur Melihat Transaksi Toko oleh Admin sebagai Aktor.....	A-26
Tabel D. 13 <i>Test Case</i> Fitur Melihat Produk Kadaluwarsa oleh Admin sebagai Aktor.....	A-27
Tabel D. 14 <i>Test Case</i> Fitur Melihat Produk Kosong oleh Admin sebagai Aktor.....	A-28
Tabel E. 1 <i>Use Case Description</i> Mengelola Keranjang Transaksi Toko.....	A-29
Tabel E. 1 <i>Use Case Description</i> Mengelola Keranjang Transaksi Toko (Lanjutan).....	A-30
Tabel E. 2 <i>Use Case Description</i> Menghapus Transaksi Toko	A-30
Tabel E. 3 <i>Use Case Description</i> Mengelola Profil	A-31
Tabel E. 4 <i>Use Case Description</i> Mengelola Informasi Produk.....	A-32
Tabel E. 5 <i>Use Case Description</i> Mengelola <i>Batch</i> Produk.....	A-33
Tabel E. 6 <i>Use Case Description</i> Mengakses Rangkuman Data <i>Dashboard</i>	A-34
Tabel E. 7 <i>Use Case Description</i> Mengelola Akses <i>Backdoor</i>	A-35
Tabel E. 8 <i>Use Case Description</i> Mengelola Akses Menu.....	A-36
Tabel E. 9 <i>Use Case Description</i> Mengelola Kategori Produk	A-37
Tabel E. 10 <i>Use Case Description</i> Mengelola Unit Produk	A-38
Tabel E. 11 <i>Use Case Description</i> Mencari Produk di <i>Dashboard</i>	A-39
Tabel E. 12 <i>Use Case Description</i> Mengakses Laporan Laba.....	A-40
Tabel E. 13 <i>Use Case Description</i> Mengelola <i>API Key</i>	A-41
Tabel E. 14 <i>Use Case Description</i> Mengelola <i>User</i>	A-42
Tabel E. 15 <i>Use Case Description</i> Mengelola <i>Role</i>	A-43
Tabel E. 16 <i>Use Case Description</i> Mengakses Transaksi Toko	A-44
Tabel E. 17 <i>Use Case Description</i> Mengakses <i>Batch</i> Kadaluwarsa	A-45
Tabel E. 18 <i>Use Case Description</i> Mengakses Produk Kosong	A-46
Tabel E. 19 <i>Use Case Description</i> Mengakses Informasi Produk	A-47
Tabel E. 20 <i>Use Case Description</i> Mengakses <i>Batch</i> Produk	A-48



DAFTAR LAMPIRAN

LAMPIRAN A HASIL WAWANCARA.....	A-1
LAMPIRAN B FORM LEMBAR MONITORING BIMBINGAN TUGAS AKHIR.....	A-8
LAMPIRAN C LINK VIDEO DEMO.....	A-11
LAMPIRAN D TAHAP PENGUJIAN	A-12
LAMPIRAN E <i>USE CASE DESCRIPTION</i>	A-29



BAB I

PENDAHULUAN

1.1 Latar Belakang

Perkembangan teknologi telah menjadi kebutuhan dasar masyarakat, terutama dalam bisnis transaksi jual beli. Salah satu kebutuhan teknologi informasi adalah penggunaan situs jual beli *online*, atau biasa dikenal sebagai *online marketplace*. *Online marketplace* adalah sebuah *platform* penghubung untuk penjual dan pembeli dapat saling berinteraksi dan melakukan transaksi dari jarak jauh. *Online marketplace* juga berkontribusi besar dalam mengoptimalkan upaya-upaya UMKM untuk bertahan dan mengembangkan bisnis selama pandemi COVID-19. [1]

Online marketplace telah berkembang sangat pesat, berkat kemudahannya yang ditawarkan. Jika pada sistem konvensional, *customer* harus datang ke tempat untuk memesan produk yang dibutuhkan, sementara kemudahan *online marketplace* sekarang membuat *customer* cukup memesan produk yang dibutuhkan, dan produk akan diantarkan ke lokasi *customer* yang dituju. Tempat usaha juga terdorong untuk berekspansi bisnis pada *online marketplace*, agar dapat menarik *customer* untuk dapat bertransaksi secara jarak jauh, sehingga dapat meningkatkan penjualan. Mengingat mulai banyaknya *online marketplace*, dengan perbedaan layanan harga dan promo yang ditawarkan, maka tempat usaha juga beradaptasi untuk memakai berbagai layanan *online marketplace*, agar dapat menjangkau berbagai segmentasi pasar, sesuai dengan minat pasar. Hal ini dapat menjadi peluang bisnis yang menguntungkan, sekaligus menjadi tantangan bagi perusahaan, dalam mengatur ketersediaan stok produk, saat perusahaan memilih untuk menggunakan sistem dari pihak lain, dan sistem tersebut belum terintegrasi dengan sistem milik perusahaan, yang membuat perusahaan harus memeriksa dan memperbarui data stok produk pada *marketplace*, secara manual. Hal ini dianggap kurang efisien dan memiliki risiko apabila perusahaan tidak selalu memeriksa dan memperbarui stok produk, yang dapat mengakibatkan stok produk di *marketplace* menjadi tidak *up-to-date* dengan stok yang sebenarnya. Ini dapat

mengakibatkan terjadinya *overselling*, yaitu memperoleh pesanan melebihi jumlah ketersediaan. Hal tersebut mengakibatkan kerugian pada toko, seperti pengeluaran usaha lebih oleh pihak toko untuk mencari produk tersebut, atau toko terpaksa membatalkan pesanan, sehingga dapat mengalami penurunan rating, dan menurunnya *insight* toko pada kolom pencarian *marketplace*. Selain itu, perusahaan dapat mengalami kesulitan dalam melacak dan menelusuri histori produk yang mempunyai tanggal *expired* yang berbeda-beda, yang dapat membuat produk dapat dibiarkan tidak terjual, yang membuat perusahaan mengalami kerugian secara finansial. Perusahaan yang dimaksud adalah sebuah tempat usaha yang bergerak di bidang apotek, bernama Apotek 99.

Berdasarkan latar belakang yang ada, maka dibutuhkan sebuah solusi yang mampu memperbarui data stok produk pada konvensional dan *marketplace* secara *up-to-date*, sesuai pengelolaan toko dan aktivitas transaksi penjualan, dengan menggunakan *API Integrator* antara sistem Apotek 99 dan sistem *marketplace*, kemudian dapat mencatat dan membagi produk berdasarkan tanggal *expired*.

1.2 Perumusan Masalah

Berdasarkan latar belakang yang telah diuraikan sebelumnya, maka dapat dirumuskan permasalahannya sebagai berikut:

1. Perusahaan mengalami kesulitan dalam mengatur ketersediaan stok produk pada *marketplace*, karena perusahaan menggunakan sistem dari pihak lain, yang belum terintegrasi dengan sistem perusahaan.
2. Risiko *overselling* yang dapat menyebabkan kerugian pada toko, seperti pengeluaran usaha lebih oleh pihak toko untuk mencari produk tersebut, atau toko terpaksa membatalkan pesanan.
3. Kesulitan dalam melacak dan membedakan produk yang mempunyai tanggal *expired* yang berbeda-beda.

1.3 Pembatasan Masalah

Berdasarkan perumusan masalah sebelumnya, pada tugas akhir ini didapati pembatasan masalah sebagai berikut:

1. Sistem hanya digunakan oleh pihak Apotek 99.

2. Fitur transaksi penjualan hanya digunakan untuk mengelola transaksi konvensional saja.
3. Sistem hanya mensinkronisasikan dan memperbarui data stok produk.

1.4 Tujuan dan Manfaat Penelitian

Berdasarkan latar belakang yang telah dijabarkan, maka tugas akhir ini memiliki tujuan dan manfaat sebagai berikut:

1.4.1 Tujuan Penelitian

Tujuan penelitian ini adalah untuk mengembangkan sistem Apotek 99 yang dapat mensinkronisasikan dan memperbarui data stok produk pada konvensional dan *marketplace*, untuk mempersingkat proses dalam mengatur ketersediaan stok produk, dan mengelompokkan stok produk berdasarkan tanggal *expired*, untuk mengatur umur simpan produk yang akan dijual.

1.4.2 Manfaat Penelitian

Manfaat penelitian ini adalah untuk memberikan kemudahan bagi pemilik. Sistem yang dirancang dapat mengurangi waktu dan upaya dalam mengelola ketersediaan produk, lalu pengelompokkan produk berdasarkan tanggal *expired* dapat meningkatkan efisiensi dan kemudahan dalam melacak dan menelusuri histori produk.

1.5 Metodologi Penelitian

1.5.1 Metode Pengumpulan Data

1. Studi literatur

Metode pengumpulan data yang dilakukan pada penelitian ini adalah *literature review*. Metode ini merupakan metode umum yang sering digunakan oleh para peneliti untuk membantu dalam mencari gambaran terkait topik yang akan dibahas. Cara kerja dari metode ini ialah dengan mengumpulkan beberapa karya penelitian dari para

praktisi dan peneliti lainnya yang pembahasannya berkaitan dengan topik penelitian.

2. Observasi

Observasi merupakan jenis metode pengumpulan data dengan melakukan pengamatan langsung terhadap objek penelitian. Pada penelitian ini, peneliti akan melakukan pengamatan terhadap proses yang berlangsung pada Apotek 99 dan mencatat informasi yang didapatkan sebagai hasil observasi.

3. Wawancara

Wawancara merupakan jenis pengumpulan data dengan melakukan tanya jawab langsung terhadap subjek penelitian. Pada penelitian ini, peneliti akan menanyakan aktifitas sistem yang digunakan Apotek 99 dan kendala apa saja yang dialami saat sedang sistem digunakan.

1.5.2 Metodologi Pengembangan Sistem

Metode yang digunakan dalam pengembangan sistem adalah model *prototyping*. Dalam model ini terdapat 7 fase, berdasarkan sumber pada [2], yaitu: *planning, analysis, design, implementation, system prototype, implementation, dan system*. Ketujuh tahapan tersebut dilakukan secara berurutan dari langkah 1 hingga langkah 7. Proses yang terjadi pada langkah 2, langkah 3, dan langkah 4 dapat dibuat jadi satu. Ketiga langkah tersebut dilakukan setelah melakukan langkah 1, dan akan menghasilkan langkah 5, yang masih dapat diubah dan diulur, sebelum langkah 6 dan langkah 7. Pada tahap awal, dilakukan perencanaan dan analisis kebutuhan pada Apotek 99. Menganalisis masalah apa yang terjadi, perangkat lunak seperti apa yang digunakan, dan apa saja yang perlu ditingkatkan sesuai keperluan apotek. Mengetahui tujuan dan manfaat dari sistem yang akan direkomendasikan, serta melakukan pengumpulan data lapangan lainnya. Dari informasi yang diperoleh, kemudian diolah dan di analisa sehingga didapatkan spesifikasi kebutuhan dari Apotek 99. Selanjutnya adalah tahap

desain, yakni proses perancangan sistem yang akan dibangun. Dari informasi spesifikasi yang didapatkan, kemudian diimplementasikan pada desain pengembangan sehingga membantu dalam memberikan gambaran mengenai apa yang harus dikerjakan. Adapun tahap perancangan pada penelitian ini meliputi perancangan basis data dengan menggunakan tabel ERD (*Entity Relationship Diagram*), representasi antar muka, dan arsitektur perangkat lunak. Setelah selesai melakukan desain, langkah selanjutnya adalah tahap coding yaitu membuat algoritma atau kode *logic* untuk menghasilkan sebuah program. Pada penelitian ini bahasa pemrograman yang akan digunakan adalah bahasa pemrograman PHP. Setelah selesai melakukan pemrograman maka langkah selanjutnya adalah testing. Pada langkah ini program yang sudah dibuat akan diuji coba terlebih dahulu sebelum diimplementasikan apakah berjalan sesuai yang diharapkan dan memastikan tidak ada bug yang terjadi. Jika program lolos dalam tahap testing maka langkah berikutnya adalah memberikan prototipe sistem kepada Apotek. Apabila kebutuhan dari prototipe sistem tersebut telah terpenuhi, maka langkah yang terakhir adalah implementasi. Pada tahap ini, sistem sudah siap untuk digunakan oleh objek penelitian.

1.6 Sistematika Penulisan

Berikut adalah sistematika penulisan yang digunakan pada tugas akhir ini:

BAB I: PENDAHULUAN

Bab ini berisi tentang latar belakang, perumusan masalah, pembatasan masalah, tujuan penelitian, manfaat penelitian, metodologi penelitian, dan metode pengumpulan data.

BAB II: LANDASAN TEORI

Bab ini berisi teori-teori yang mendasari penelitian ini, yang berupa teori yang dikutip dari buku maupun jurnal.

BAB III: SISTEM SAAT INI

Bab ini berisi profil perusahaan, struktur organisasi, sistem saat ini dan kendala yang dihadapi oleh Apotek 99.

BAB IV: ANALISIS DAN PERANCANGAN SISTEM

Bab ini berisi analisa dan perancangan sistem berbasis web untuk mengatasi masalah yang dihadapi oleh Apotek 99.

BAB V: KESIMPULAN DAN SARAN

Bab ini berisi kesimpulan dari penelitian yang dilakukan dan saran untuk penelitian-penelitian selanjutnya.



BAB II

LANDASAN TEORI

2.1 Website

Website merupakan sebuah program yang berjalan menggunakan *server* dan protokol HTTP, dan menyajikan file yang membentuk halaman *web* bagi penggunaannya untuk memberikan konten berupa informasi, teks, gambar, dan video kepada pengguna akhir melalui internet. Server fisik, sistem operasi, dan *software* menjadi komponen penting dalam membangun server *web*. [2]

2.2 Application Programming Interface (API)

Application Programming Interface (API) adalah kumpulan kode yang memungkinkan perangkat, aplikasi software, dan data server dapat saling berinteraksi, dan menjadi penghubung antar layanan yang digunakan. API memiliki kesamaan dengan website, berupa mengembalikan data, namun API tidak mengembalikan semua data utuh seperti website, melainkan hanya data mentah, yang harus diolah kembali oleh sistem, untuk menjadi informasi, yang dapat dibaca oleh *user*. [3]

2.3 API Integrator Marketplace

API Integrator Marketplace adalah sistem penghubung antar marketplace dan tempat usaha. API Integrator Marketplace memungkinkan tempat usaha untuk dapat berinteraksi dengan sistem marketplace, melalui *interface* sistem milik perusahaan.

2.4 HyperText Markup Language (HTML)

HyperText Markup Language (HTML) adalah bahasa yang digunakan untuk merancang dan membangun sebuah halaman *web*. Standard Generalized Markup Language (SGML), yang digunakan dalam standar pemformatan untuk dokumen teks, menjadi dasar arah dalam pembuatan HTML. Tim Berners-Lee

menciptakan HTML, saat sedang bekerja untuk CERN, dan menjadi populer berkat penggunaannya pada *browser* bernama Mosaic. HTML terus berkembang hingga tahun 1990. Penggunaannya yang gratis, membuat HTML dapat digunakan oleh semua *developer web*, dan terus dikembangkan oleh komunitas dalam skala global. HTML dapat digunakan dan diubah oleh teks editor apapun. Terdapat beberapa elemen yang dimulai dengan simbol membuka “<” dan berakhir dengan ”>” pada dokumen HTML. [2]

2.5 *Cascading Style Sheets (CSS)*

Cascading Style Sheets (CSS) adalah bahasa yang digunakan dalam mengontrol tampilan dokumen, dengan penulisan bahasa *markup*. Secara umum, CSS adalah bahasa pemrograman yang digunakan untuk mengubah desain dan tampilan halaman *web*. CSS menjadi pelengkap dari Bahasa HTML dengan memberikan nuansa ke dalam halaman *web*, seperti warna, gambar, jenis *font*, ukuran *font*, hingga perubahan tata letak halaman. [2]

2.6 *HyperText Processor (PHP)*

HyperText Processor (PHP) adalah bahasa pemrograman berbasis *web-server*, yang memproses data di sisi *server*, dan pemrograman ini bersifat *open-source*. Pembuatan sebuah halaman *website* yang dinamis dapat dilakukan, dengan mengintegrasikan PHP dan HTML. Artinya, halaman yang akan ditampilkan akan dibuat sebagai tanggapan atas permintaan *client*. *Client* akan menerima informasi terbaru, dengan cara ini. Setiap *script* PHP dijalankan di server tempatnya ditulis. Perbedaan kode PHP dan HTML terletak pada pemberian tag pembuka di depan kode PHP, dan tag penutup di akhir kode PHP. [4]

2.7 *Javascript*

JavaScript adalah bahasa pemrograman tingkat tinggi untuk membuat *website* yang dinamis. *JavaScript* dapat digunakan pada sebagian besar *web browser*, seperti Internet Explorer, Mozilla Firefox, Google Chrome, dan sebagainya. *JavaScript* dapat dideskripsikan dalam bentuk fungsi yang diletakkan pada bagian dalam tag <head> yang dibuka dengan tag <script

`language="javascript">`. Script yang terdapat pada *JavaScript* mempunyai konsep yang sama seperti PHP, seperti deklarasi *variable*, penggunaan operator, *looping*, percabangan, dan fungsi. Selain itu, *JavaScript* juga mempunyai *Alert* untuk menampilkan kotak pesan pada *browser* pada saat fungsinya dijalankan. [2]

2.8 CodeIgniter

CodeIgniter adalah sebuah framework dengan bahasa pemrograman *PHP* dengan konsep *MVC(model-view-controller)*, dan bersifat *open-source* dan *object-oriented programming (OOP)*. *MVC* membuat web dapat memisahkan pengembangan sistem ke dalam 3 jenis komponen, yakni *view*, *model*, *controller*. *View* adalah bagian untuk menangani tampilan file *HTML*, untuk menerima dan mempresentasikan data kepada pengguna. *Model* adalah bagian untuk menangani database, yakni memanipulasi data. *Controller* adalah bagian untuk mengatur hubungan *view* dan *model*, yaitu menerima data pengguna dan menentukan proses selanjutnya pada sistem. [5]

2.9 Bootstrap

Bootstrap adalah kerangka kerja(*framework tools*) yang dapat digunakan secara gratis, untuk membuat aplikasi *web* yang *responsive*, dengan cepat dan mudah. *Bootstrap* dapat membuat *Grids*, *Layouts*, *Typography*, *Tables*, *Forms*, *Navigation*, dan lain-lain, dengan menggunakan *HTML* dan *CSS*. *Modals*, *Transitions*, *Dropdowns*, *Scrollspy*, *Tooltips*, *Tabs*, *Alerts*, *Buttons*, dan *Carousels* hanyalah beberapa komponen *user interface* yang dapat dibuat oleh *Bootstrap*. *Bootstrap* memungkinkan pengguna untuk dapat membuat sistem *web* *responsif* dengan cepat dan mudah. [6]

2.10 Marketplace

Marketplace adalah media *online* berbasis internet untuk melakukan kegiatan bisnis dan transaksi antara pembeli dan penjual. Menurut Brunn, Jensen, & Skovgaard, *marketplace* adalah wadah komunitas bisnis interaktif secara elektronik yang menyediakan pasar dimana perusahaan dapat ambil andil dalam kegiatan jual beli e-commerce. Sehingga, *marketplace* adalah

wadah pemasaran produk secara elektronik yang mempertemukan banyak penjual dan pembeli untuk saling bertransaksi. [7]

2.10.1 Tokopedia

Tokopedia adalah *marketplace* yang berdiri pada tahun 2009 oleh William Tanuwijaya dan Leontinus Alpha Edison, dengan tujuan untuk pemerataan ekonomi secara digital, dan menjangkau kebutuhan konsumen secara merata, baik di kota besar dan kota kecil di seluruh Indonesia. [8]

Di masa sekarang, per kuartal kedua tahun 2021, Tokopedia telah memiliki jumlah pengguna aktif sebanyak 147 juta [9]. Menurut data survei dari Tempo, Tokopedia dinyatakan sebagai e-commerce dengan penghasilan omzet penjualan dan transaksi penjualan terbesar, sebesar 35-36% dari seluruh responden. Selain itu, Tokopedia juga memiliki brand-awareness yang sangat tinggi, dengan 35% responden pada level paling ingat, dan 62% pada level ingat. [1]

2.10.1.1 Syarat Dan Ketentuan Tokopedia Pada Jenis Barang Obat

Berikut syarat dan ketentuan Tokopedia pada bagian J nomor 1 tentang obat-obatan: Segala jenis obat-obatan maupun zat-zat lain yang dilarang ataupun dibatasi peredarannya menurut ketentuan hukum yang berlaku, termasuk namun tidak terbatas pada ketentuan Undang-Undang Narkotika, Undang-Undang Psikotropika, dan Undang-Undang Kesehatan. Termasuk pula dalam ketentuan ini adalah obat keras, obat-obatan yang memerlukan resep dokter, obat bius dan sejenisnya, atau obat yang tidak memiliki izin edar dari Badan Pengawas Obat dan Makanan (BPOM). [10]

Dari syarat dan ketentuan tersebut, dapat diambil informasi bahwa Tokopedia hanya mengizinkan penjualan obat

bebas pada toko, dan melarang adanya penjualan obat keras, yang memerlukan resep dokter.

2.10.2 Shopee

Shopee adalah marketplace yang berdiri pada tahun 2015 oleh *Sea Group*, perusahaan asal Singapura, dengan tujuan untuk menyediakan platform untuk menghubungkan pembeli dan penjual dalam satu komunitas untuk dapat mengubah dunia menjadi lebih baik, dengan transformatif teknologi. *Shopee* diluncurkan di 7 negara pada Asia Tenggara, termasuk Indonesia. [11]

Di masa sekarang, per kuartal kedua tahun 2021, *Shopee* telah memiliki jumlah pengguna aktif sebanyak 126 juta [12]. Menurut data survei dari Tempo, *Shopee* dinyatakan sebagai e-commerce dengan penghasilan omzet penjualan dan transaksi penjualan kedua terbesar, sebesar 32-33% dari seluruh responden. Selain itu, *Shopee* juga memiliki brand-awareness yang sangat tinggi, dengan 34% responden pada level paling ingat, dan 60% pada level ingat. [1]

2.10.2.1 Syarat Dan Ketentuan *Shopee* Pada Jenis Barang Obat

Berikut syarat dan ketentuan *Shopee* pada nomor 2 bagian ii tentang obat-obatan: Segala jenis obat-obatan maupun zat-zat lain yang dilarang ataupun dibatasi peredarannya menurut ketentuan hukum yang berlaku, termasuk namun tidak terbatas pada ketentuan Undang-Undang Narkotika, Undang-Undang Psikotropika, dan Undang-Undang Kesehatan di Republik Indonesia. Termasuk pula dalam ketentuan ini adalah obat keras, obat-obatan yang memerlukan resep dokter, obat bius, obat pelangsing dan sejenisnya, atau obat yang tidak memiliki izin edar resmi dari Badan Pengawas Obat dan Makanan (BPOM). [13]

Dari syarat dan ketentuan tersebut, dapat diambil informasi bahwa *Shopee* hanya mengizinkan penjualan obat bebas pada toko, dan melarang adanya penjualan obat keras, yang memerlukan resep dokter.

2.11 *SmartSeller*

SmartSeller adalah aplikasi yang dapat mengelola order, produk, *customer*, pengiriman, dan analisa bisnis. *SmartSeller* didirikan pada tahun 2016, oleh dua *founder*, yakni Agus E Setiyono, dan Alif Akbar Fitrawan. Pada awalnya, *SmartSeller* hadir sebagai aplikasi manajemen *order* sederhana, untuk mengatasi masalah pada pebisnis *online* yang mengalami kesulitan dalam melakukan *recap* pesanan pada *online platform*. Per tahun 2020, *SmartSeller* telah mengembangkan bekerjasama dengan *Shopee* dan *Tokopedia* untuk mendukung fitur *omnichannel*, dalam mengelola produk dan pesanan. [14]

2.11.1 *SmartSeller API*

SmartSeller API adalah *API* yang disediakan oleh *SmartSeller*, untuk dapat mengintegrasikan sistem milik perusahaan dengan platform *SmartSeller*. *API* dibangun berdasarkan konsep *REST API*, dan akan mengembalikan *request data* dalam format *JSON*. Diperlukan satu toko aktif di layanan *SmartSeller*, untuk dapat menggunakan layanan *SmartSeller API* [15]. Terdapat beberapa jenis lokasi parameter yang harus dikirimkan pada *API*, seperti:

1. *Body*

Sebuah *request* yang harus menyertakan data dalam format *JSON*, yang biasanya digunakan untuk *request POST* atau *PATCH* [15]. Berikut contoh data *JSON* yang dikirimkan dalam melakukan *generate product SKU*:

```
{
  "products": [
    {
      "name": "Sepatu",
    }
  ]
}
```

Dan berikut data sukses yang dihasilkan dalam format JSON:

```
{
  "response": "Success",
  "data": [
    {
      "name": "Sepatu",
      "sku": "SEP-C1E",
    }
  ]
}
```

2. Path

Sebuah *request* yang berlokasi dalam URL yang akan dipanggil, yang biasanya digunakan untuk *request GET* [15]. Berikut contoh *Path* yang dikirimkan untuk melakukan pemanggilan data kategori:

```
https://api.smartseller.co.id/api/open/product-category/116289
```

Dan berikut data sukses yang dihasilkan dalam format JSON:

```
{
  "response": "Success",
  "data": [
    {
      "id": 116289,
      "name": "Alergi",
      "root_categories_id": null,
      "root_categories_name": "",
      "updated_at": null
    }
  ]
}
```

3. Query

Sebuah *request* yang berlokasi dalam URL yang ditulis setelah tanda tanya, dan *query* ini bersifat opsional untuk digunakan dalam *request GET* [15]. Berikut contoh *query* yang dikirimkan untuk melakukan pengecekan produk stock:

```
https://api.smartseller.co.id/api/open/product/check-stock?variation_id=15311186
```

Dan berikut data sukses yang dihasilkan dalam format JSON:

```
{
  "response": "Success",
```

```
"data":
  {
    "stock": 3,
    "price": 95989,
    "price_dropship": 95989,
    "price1": 95989,
    "price2": 95989,
    "price3": 95989
  }
}
```

Berikut beberapa fungsi API yang disediakan oleh SmartSeller:

1. *Custom Product Category*:

- a. *Create Category*: Pembuatan data kategori produk.
- b. *Get Categories*: Mendapatkan data kategori produk.
- c. *Update Category*: Mengubah data kategori produk.
- d. *Delete Category*: Menghapus data kategori produk.

2. *Product*:

- a. *Generate Product SKU*: Pembuatan data SKU untuk produk.
- b. *Create Product*: Pembuatan data produk.
- c. *Get Products*: Mendapatkan data produk.
- d. *Check Product Stock*: Memeriksa data stok produk.
- e. *Update Product*: Mengubah data produk.
- f. *Update Stock*: Mengubah data stok produk.
- g. *Delete Products*: Menghapus data produk.

2.12 Database Management System (DBMS)

Database Management System (DBMS) merupakan seperangkat program yang dirancang untuk membuat pengguna lebih mudah untuk menambah, mengedit, menghapus, dan mengambil data dari basis data. Microsoft SQL, MySQL, Oracle, MS. Access, dan sebagainya adalah beberapa contoh DBMS. [16]

2.12.1 Relational Database Management System (rDBMS)

Relational Database Management System (RDBMS) adalah *software* pengelola data yang terdapat di dalam *database* relasional.

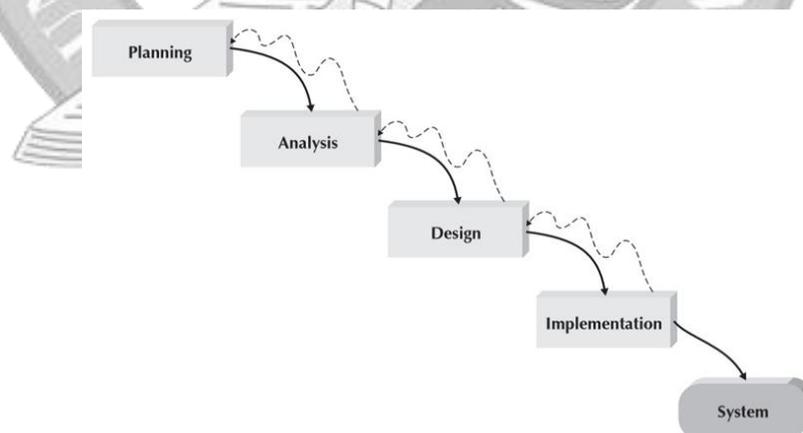
RDBMS dapat digunakan dalam pembuatan tabel, penginputan data, penghapusan data, dan manajemen data, seperti manajemen *user*, keamanan data, *backup* data, dan *recovery* data. Salah satu contoh *software* RDBMS adalah MySQL. [17]

2.12.2 MySQL

MySQL adalah sebuah *database server* atau *database engine* yang mengelola data menggunakan bahasa *database* SQL. MySQL adalah salah satu jenis RDBMS (*Relational Database Management System*), yang memanfaatkan baris, kolom, dan tabel. Karena *database server* ini bisa digunakan secara gratis, maka siapapun bisa menggunakan MySQL. [17]

2.13 Software Development Life Cycle (SDLC)

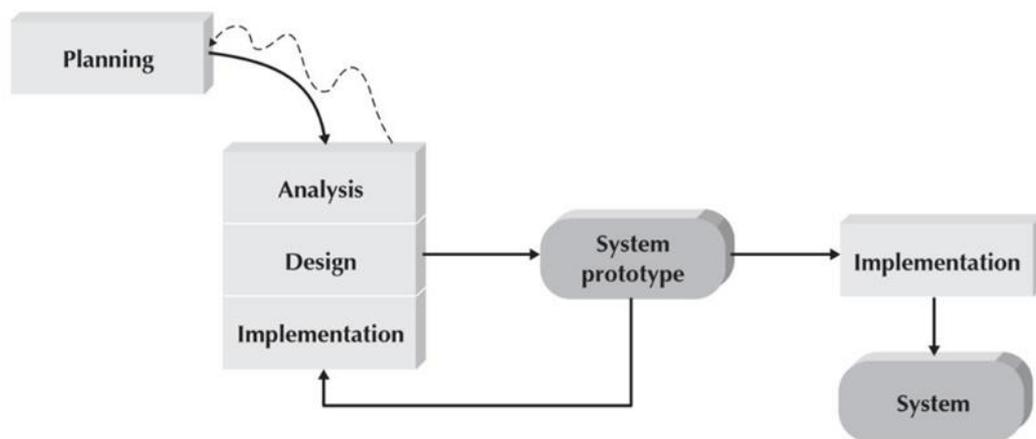
Software Development Life Cycle (SDLC) adalah proses dalam mencari tahu bagaimana sistem informasi dapat membantu dalam kebutuhan bisnis, dimulai dengan perancangan sistem, pembangunan sistem, dan pengiriman sistem ke pengguna. Terdapat 4 tahapan siklus dalam SDLC, yaitu fase perencanaan, fase analisis, fase desain, dan fase implementasi. [18]



Gambar 2. 1 Tahap-Tahap *Software Development Life Cycle* (SDLC)
Sumber: Dennis et al, 2015, p. 7.

2.14 Prototyping

Metodologi *prototyping* melakukan tiga fase secara bersamaan, yaitu fase analisis, desain, dan implementasi, dan ketiganya dilakukan berulang kali dalam satu siklus hingga suatu sistem selesai dibangun. Prototipe pertama biasanya ditujukan kepada pengguna dan sponsor proyek untuk memberikan komentar atau feedback. Komentar-komentar ini digunakan untuk melakukan *reanalyze*, *redesign*, dan *reimplementation* pada prototipe selanjutnya yang menyediakan beberapa fitur tambahan. Proses berlanjut dalam satu siklus hingga *stakeholder* menerima dan setuju bahwa prototipe tersebut sudah menyediakan fungsi yang layak untuk digunakan dalam organisasi. [18]



Gambar 2. 2 Metodologi *Prototyping*
Sumber: Dennis et al, 2013, p. 10.

2.15 Unified Modeling Language (UML)

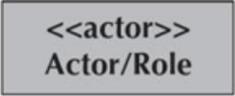
Unified Modelling Language (UML) adalah seperangkat teknik diagram yang digunakan dalam pengembangan sistem informasi. UML bertujuan untuk menyediakan teknik diagram, serta istilah *object oriented* yang beragam untuk membentuk model proyek pengembangan sistem, yang dimulai dari analisis hingga implementasi. UML dinyatakan sebagai standar pada seluruh pengembangan proyek pada November 1997 oleh Object Management Group (OMG) [18]. Saat ini, UML 2.5 adalah versi UML yang digunakan.

2.15.1 Use Case Diagram

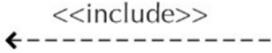
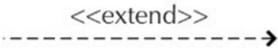
Use case adalah cara formal yang menggambarkan bagaimana sistem bisnis berinteraksi dengan aktor dan proses bisnis. Pada dasarnya,

use case diagram adalah tinjauan high-level dari proses bisnis dalam sistem informasi bisnis yang mewakili seluruh dasar untuk sistem berorientasi objek [18]. Dengan *use case diagram*, pengguna sistem dapat mengerti fungsi-fungsi sistem yang dimaksud tanpa harus mengetahui cara mengembangkan sistem.

Tabel 2. 1 *Syntax Use Case Diagram*

Keterangan	Simbol
<p>Actor</p> <ul style="list-style-type: none"> • Seseorang atau sistem yang memperoleh manfaat dan bersifat eksternal terhadap subjek. • Digambarkan sebagai stick figure atau persegi panjang (jika aktor bukan manusia) dengan tulisan <<actor>> di dalamnya. • Diberi label dengan perannya. • Dapat diasosiasikan dengan aktor lain menggunakan asosiasi spesialisasi/superclass, dilambangkan dengan panah khusus. • Ditempatkan di luar batas subjek 	 <p>Actor/Role</p> 
<p>Use Case</p> <ul style="list-style-type: none"> • Merepresentasikan bagian utama dari fungsi sistem. • Dapat extend ke use case lain. • Dapat include use case lain. • Ditempatkan di dalam batas sistem. • Diberi label dengan frasa kata kerja – kata benda deskriptif 	 <p>Use Case</p>
<p>Subject Boundary</p> <ul style="list-style-type: none"> • Nama subjek ditulis di dalam atau di atas simbol. • Merepresentasikan lingkup subjek, seperti sistem atau proses bisnis. 	 <p>Subject</p>
<p>Association Relationship</p> <ul style="list-style-type: none"> • Menghubungkan <i>actor</i> dengan <i>use case</i> yang berinteraksi dengannya. 	

Tabel 2. 1 *Syntax Use Case Diagram (Lanjutan)*

<p><i>Include Relationship</i></p> <ul style="list-style-type: none"> • Merepresentasikan penyertaan fungsi dari suatu <i>use case</i> di dalam <i>use case</i> yang lain. • Panah ditarik dari <i>use case</i> asal ke <i>use case</i> yang disertakan. 	
<p><i>Extend Relationship</i></p> <ul style="list-style-type: none"> • Merepresentasikan ekstensi dari suatu <i>use case</i> untuk memasukkan perilaku opsional. • Panah ditarik dari <i>use case</i> ekstensi ke <i>use case</i> dasar. 	
<p><i>Generalization Relationship</i></p> <ul style="list-style-type: none"> • Merepresentasikan <i>use case</i> khusus ke yang lebih umum. • Panah ditarik dari <i>use case</i> khusus ke <i>use case</i> dasar. 	

Sumber: A. Dennis, B. H. Wixom and D. Tegarden, System Analysis and Design; an Object Oriented Approach with UML fifth edition, Hoboken: John Wiley and Sons. Inc, 2015

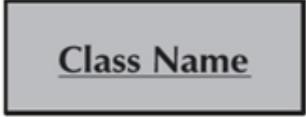
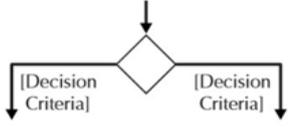
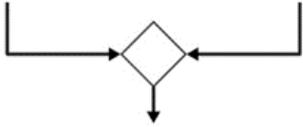
2.15.2 Activity Diagram

Activity diagram adalah model untuk memodelkan berbagai jenis proses, termasuk alur kerja bisnis tingkat *high-level* dengan *use case* berjumlah banyak, *use case* individual, dan perilaku dalam proses bisnis yang tidak terkait dengan objek. [18]

Tabel 2. 2 *Syntax Activity Diagram*

Keterangan	Simbol
<p><i>Action</i></p> <ul style="list-style-type: none"> • Perilaku yang sederhana dan tidak dapat diuraikan • Diberi label dengan namanya. 	
<p><i>Activity</i></p> <ul style="list-style-type: none"> • Merepresentasikan serangkaian action. • Diberi label dengan namanya. 	

Tabel 2. 2 *Syntax Activity Diagram* (Lanjutan)

<p>Object Node</p> <ul style="list-style-type: none"> • Merepresentasikan objek yang terhubung dengan serangkaian <i>object flow</i>. • Diberi label dengan nama kelasnya. 	
<p>Control Flow</p> <ul style="list-style-type: none"> • Memerlihatkan urutan eksekusi. 	
<p>Object Flow</p> <ul style="list-style-type: none"> • Memerlihatkan aliran dari suatu objek dari satu <i>activity/action</i>) ke <i>activity/action</i> lain. 	
<p>Initial Node</p> <ul style="list-style-type: none"> • Menggambarkan awal dari serangkaian <i>action</i> atau <i>activity</i>. 	
<p>Final-Activity Node</p> <ul style="list-style-type: none"> • Digunakan untuk menghentikan semua <i>control flow</i> dan <i>object flow</i> dalam suatu <i>activity</i> (atau <i>action</i>) 	
<p>Final-Flow Node</p> <ul style="list-style-type: none"> • Digunakan untuk menghentikan <i>control flow</i> atau <i>object flow</i> tertentu. 	
<p>Decision Node</p> <ul style="list-style-type: none"> • Merepresentasikan kondisi pengujian untuk memastikan bahwa <i>control flow</i> atau <i>object flow</i> hanya berada di satu jalur. • Diberi label dengan kriteria keputusan untuk melanjutkan ke jalur tertentu. 	
<p>Merge Node</p> <ul style="list-style-type: none"> • Digunakan untuk menyatukan kembali jalur decision yang berbeda yang dibuat menggunakan decision node. 	

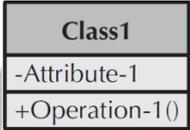
Sumber: A. Dennis, B. H. Wixom and D. Tegarden, *System Analysis and Design; an Object Oriented Approach with UML fifth edition*, Hoboken: John Wiley and Sons. Inc, 2015

2.15.3 Class Diagram

Class Diagram merupakan pemodelan untuk menggambarkan setiap objek atau data sistem. *Attributes* (*fields* dan *properties*), *events*,

dan *operation (methods)* adalah semua anggota pada setiap objek / data. *Class diagram* adalah model struktural yang menunjukkan struktur data. [18]

Tabel 2. 3 Syntax Class Diagram

Keterangan	Simbol
<p>Class: <i>Class</i> adalah mewakili jenis orang, tempat, atau informasi yang perlu ditangkap dan disimpan oleh sistem yang memiliki nama <i>class</i>, <i>attribute</i>, dan operasi.</p>	
<p>Attribute: <i>Attribute</i> merupakan sebuah gambaran keadaan dalam suatu objek. Dan dapat diturunkan dari <i>attribute</i> lain dengan menempatkan garis miring di depan nama <i>attribute</i>.</p>	<p>attribute name / derived attribute name</p>
<p>Operation: <i>Operation</i> dapat melambangkan aksi atau fungsi yang dapat dilakukan suatu <i>class</i>.</p>	<p>operation name ()</p>
<p>Association: <i>Association</i> merupakan apa yang menghubungkan antara beberapa <i>class</i> satu dengan <i>class</i> yang lainnya.</p>	<p>AssociatedWith</p> <hr/> <p>0..* 1</p>
<p>Generalization: <i>Generalization</i> merupakan perlambangan hubungan antar class yang di mana kelas anak (<i>descendent</i>) berbagi perilaku dan struktur data dari class yang ada di atasnya atau class induk(<i>ancestor</i>).</p>	
<p>Aggregation: <i>Aggregation</i> merupakan relasi antar kelas dengan makna semua bagian.</p>	<p>0..* IsPartOf ▶ 1 ◊</p>
<p>Composition: <i>Composition</i> merupakan bentuk khusus dari asosiasi yang melambangkan bagian fisik dari hubungan antar class atau class itu sendiri.</p>	<p>1..* IsPartOf ▶ 1 ◊</p>

Sumber: A. Dennis, B. H. Wixom and D. Tegarden, System Analysis and Design; an Object Oriented Approach with UML fifth edition, Hoboken: John Wiley and Sons. Inc, 2015

2.16 Penelitian Terdahulu

Terdapat tiga penelitian terdahulu yang berkaitan dengan topik penelitian yaitu oleh Kamariah, Dwi Yuli Prasetyo dengan judul “Sistem Informasi Transaksi Pada Apotek Kab. Inhil Riau Berbasis Web” [19], Abdul M. Hasyim, Yoyok S. Dwanoko, Abdul Aziz dengan judul “Rancang Bangun Sistem Informasi Manajemen Gudang Apotek Menggunakan Model *Software Development Life Cycle* (SDLC) Di Apotek Marifa” [20], Esti Astutik, Mustagfirin dengan judul “Sistem Informasi Ketersediaan Obat Menggunakan Framework Laravel di Apotek Mugi Sehat Limpung Batang” [21].

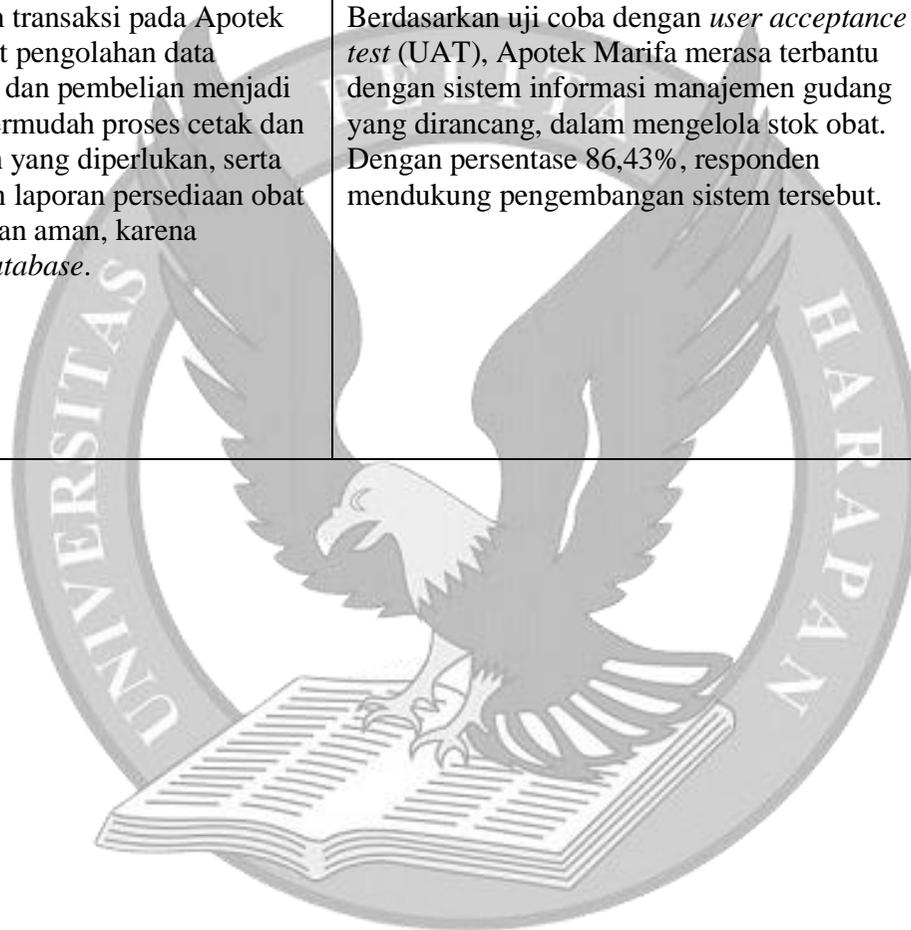


Tabel 2. 4 Penelitian Terdahulu Mengenai Pengembangan Sistem *Web API Integrator*

Keterangan	Penelitian I	Penelitian II	Penelitian III
Judul	Sistem Informasi Transaksi Pada Apotek Kab. Inhil Riau Berbasis Web	Rancang Bangun Sistem Informasi Manajemen Gudang Apotek Menggunakan Model <i>Software Development Life Cycle</i> (SDLC) Di Apotek Marifa	Sistem Informasi Ketersediaan Obat Menggunakan Framework Laravel di Apotek Mugi Sehat Limpung Batang
Penulis	Kamariah, Dwi Yuli Prasetyo	Abdul M. Hasyim, Yoyok S. Dwanoko, Abdul Aziz	Esti Astutik, Mustagfirin
Tahun	2016	2019	2020
Obyek Penelitian	Pada penelitian ini, Apotek Kab Inhil menjadi objek penelitian, untuk perancangan sistem informasi, untuk mempermudah transaksi. Sistem hanya berisi satu aktor yaitu admin.	Pada penelitian ini, Apotek Marifa menjadi objek penelitian, difokuskan untuk perancangan sistem informasi manajemen gudang, yang dirancang untuk membantu apotek dalam mengelola stok obat di gudang. Sistem ini memiliki <i>administrator</i> , operator gudang, pemilik, dan operator, sebagai keempat aktor.	Pada penelitian ini, Apotek Mugi Sehat menjadi objek penelitian, untuk perancangan sistem informasi ketersediaan obat yang dibuat untuk memudahkan pemilik apotek menyajikan data persediaan obat dan mengelola sistem secara efisien, sehingga mempercepat pendataan obat. Admin dan user adalah kedua aktor yang terdapat pada sistem.
Fungsi	Adapun sistem yang dibuat memiliki 4 menu utama, yaitu data master, pembelian, penjualan, dan laporan. Data master berisi atas pengolahan data obat, <i>supplier</i> , pelanggan, dan <i>user</i> . Fitur Pembelian dan penjualan merupakan menu untuk menampilkan dan mengelola transaksi jual beli di apotek. Fitur laporan merupakan menu berisi laporan berdasarkan pembelian, penjualan, dan stok produk.	Adapun sistem yang dibuat terbagi atas 9 fitur, yaitu login, master user, master supplier, master obat, transaksi permintaan dan penerimaan obat, transaksi untuk obat kadaluarsa, transaksi obat keluar, dan laporan. Fitur transaksi obat kadaluarsa, merupakan menu	Adapun sistem yang dibuat memiliki beberapa fitur, yaitu pengelolaan data user, kategori, supplier, barang, stok barang, pembayaran, dan cetak laporan.

Tabel 2. 4 Penelitian Terdahulu Mengenai Pengembangan Sistem *Web API Integrator* (Lanjutan)

<p>Kesimpulan</p>	<p>Perancangan sistem transaksi pada Apotek Kab. Inhil membuat pengolahan data transaksi penjualan dan pembelian menjadi efisien, dan mempermudah proses cetak dan pencarian dokumen yang diperlukan, serta dapat menghasilkan laporan persediaan obat yang lebih akurat dan aman, karena tersimpan dalam <i>database</i>.</p>	<p>Berdasarkan uji coba dengan <i>user acceptance test</i> (UAT), Apotek Marifa merasa terbantu dengan sistem informasi manajemen gudang yang dirancang, dalam mengelola stok obat. Dengan persentase 86,43%, responden mendukung pengembangan sistem tersebut.</p>	<p>Sistem ketersediaan obat pada Apotek Mugi Sehat dirancang agar pendataan obat menjadi lebih efisien, dengan memudahkan apotek menyajikan data persediaan dan mengelola sistem penyimpanan obat.</p>
--------------------------	--	---	--



BAB III

SISTEM SAAT INI

3.1 Profil Perusahaan

Apotek 99 merupakan bisnis UMKM (Usaha Mikro Kecil Menengah) yang bergerak di bidang penjualan kebutuhan kesehatan dan obat. Apotek 99 didirikan pada tahun 2001 dan berlokasi di Kota Bogor. Awal dibangunnya bisnis ini dikarenakan melihat adanya peluang yang cukup besar terhadap kebutuhan kesehatan sehari-hari, terkhususnya pada era pandemi ini. Selain itu adapun peluang lainnya yang membuat bisnis ini terbentuk adalah kurangnya apotek yang berada di Kota Bogor, pada saat itu. Apotek 99. Harga produk yang kompetitif, dan pelayanan yang baik pada *customer*, membuat Apotek 99 telah memiliki *brand awareness* yang baik di Kota Bogor. Produk yang dimiliki dan disediakan oleh Apotek 99 telah mencakup kebutuhan masyarakat akan kesehatan, seperti obat-obatan, suplemen, dan sebagainya.

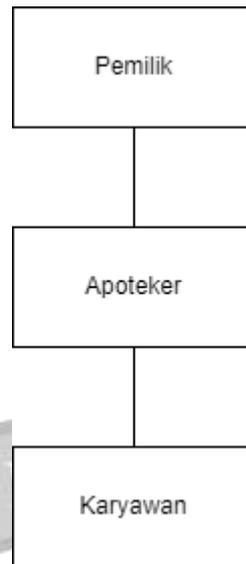
3.1.1 Visi Perusahaan

Visi dari Apotek 99 adalah menjadi Apotek dengan produk dan pelayanan yang terbaik, dan harga terjangkau, untuk meraih kebutuhan, kenyamanan, dan kepuasan pelanggan.

3.1.2 Misi Perusahaan

1. Menyediakan produk obat-obatan dan kesehatan yang mencakup kebutuhan masyarakat.
2. Memberikan informasi dan konsultasi obat terhadap penyakit *customer*.
3. Memberikan pelayanan terbaik dengan sikap kekeluargaan dan keramahtamahan kepada *customer*.
4. Menjamin produk obat-obatan yang bermutu dan berkualitas.

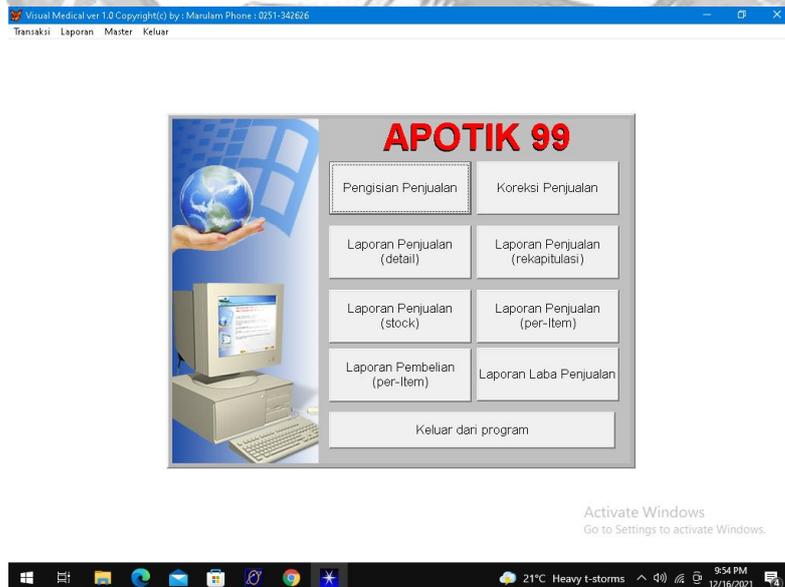
3.2 Struktur Organisasi



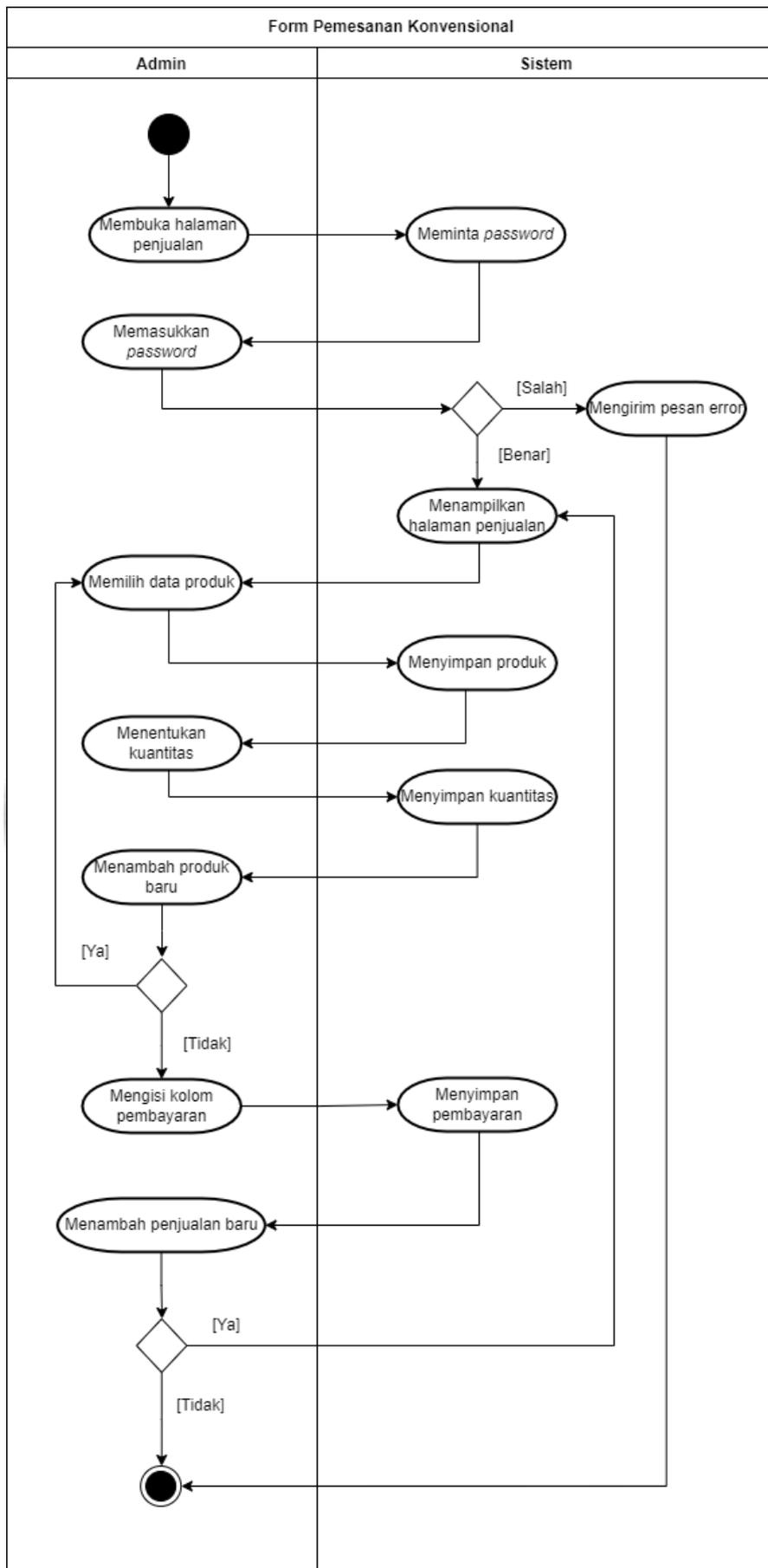
Gambar 3. 1 Struktur Organisasi Apotek 99

3.3 Analisis Sistem Saat ini

Sistem saat ini telah terkomputerisasi, menggunakan program bernama Visual Medical 1.0 oleh perusahaan penyedia software bernama Marulam. Program terbagi atas dua peran, yaitu admin dan kasir. Terdapat pada gambar 3.2 mengenai *interface* admin dan gambar 3.3 mengenai *interface* kasir.

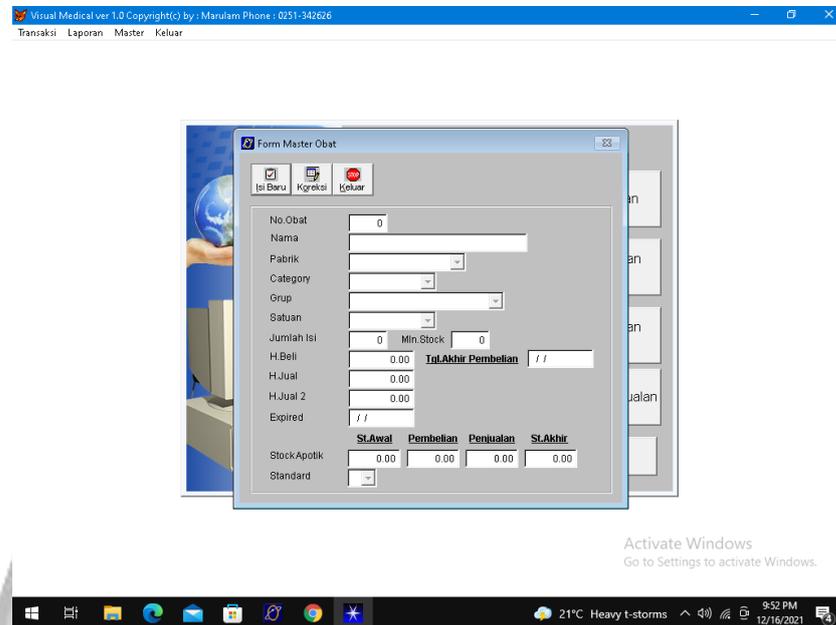


Gambar 3. 2 *Interface* Admin

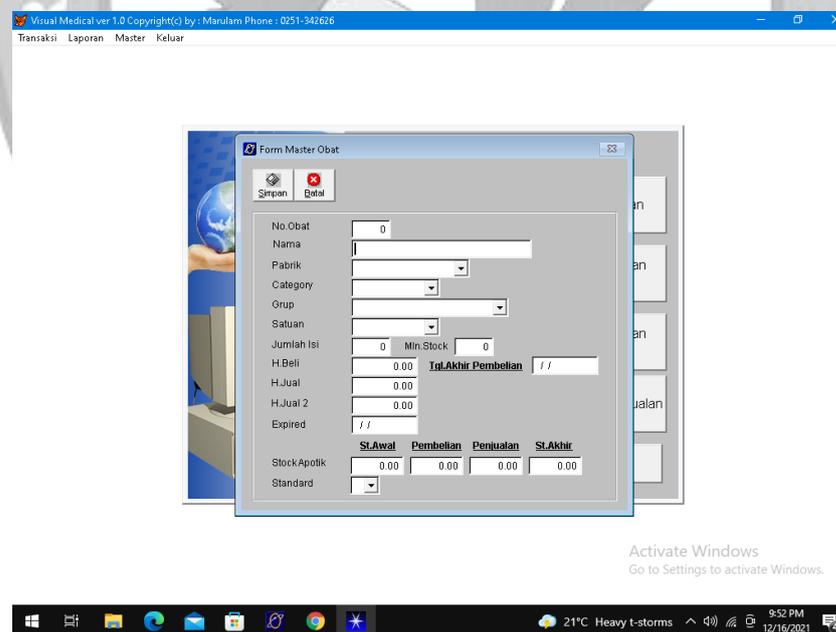


Gambar 3. 5 Activity Diagram Form Pemesanan Konvensional

Produk dapat diakses oleh admin, dengan mengakses form master obat. Terdapat pada Gambar 3.6 untuk *form* master obat, dan Gambar 3.7 untuk menambahkan obat baru.



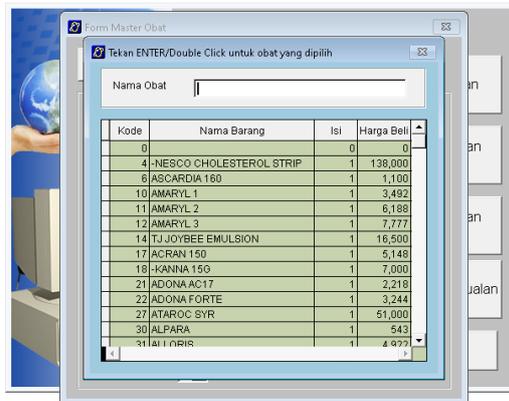
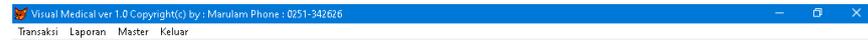
Gambar 3. 6 Form Master Obat



Gambar 3. 7 Form Isi Obat Baru

Admin dapat mengoreksi obat, dengan menekan tombol koreksi, kemudian memilih obat yang akan dikoreksi, dan mengisi form edit. Setelah itu,

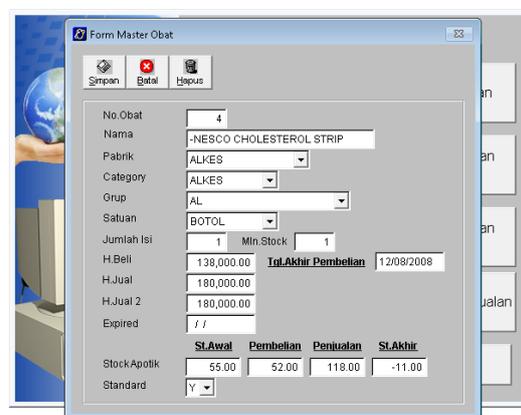
admin menekan tombol simpan. Terdapat pada Gambar 3.8 Pemilihan Obat, dan Gambar 3.9 Perubahan Data Obat.



Activate Windows
Go to Settings to activate Windows.



Gambar 3. 8 Pemilihan Obat



Activate Windows
Go to Settings to activate Windows.



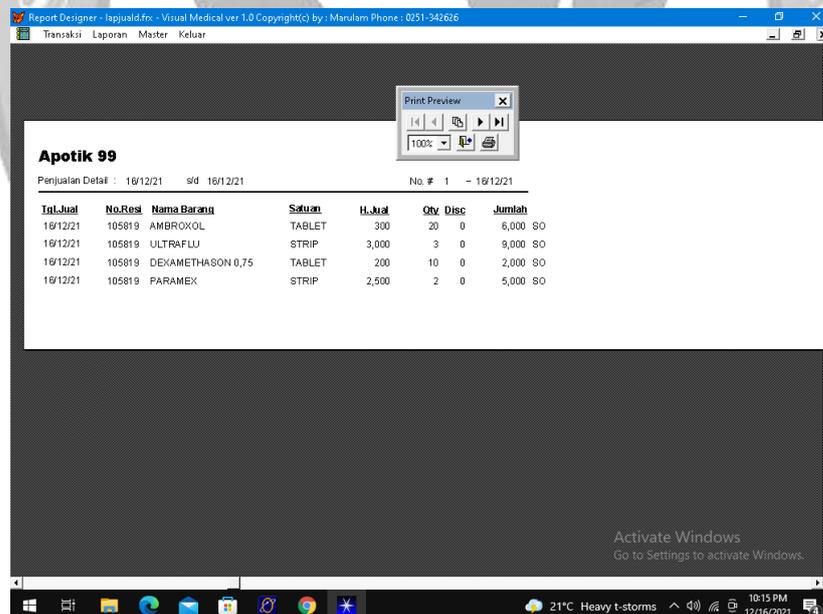
Gambar 3. 9 Perubahan Data Obat

Admin dapat mengakses laporan penjualan detail obat, dengan menentukan tanggal mulai dan tanggal akhir. Admin menekan tombol Layar, dan sistem akan menampilkan laporan dalam layar. Dapat dilihat pada Gambar

3.10 Pemilihan Tanggal Laporan Penjualan, dan Gambar 3.11 Data Laporan Penjualan.



Gambar 3. 10 Pemilihan Tanggal Laporan Penjualan



Gambar 3. 11 Data Laporan Penjualan

Aktivitas transaksi pada Apotek 99 dimulai dari kedatangan *customer* ke Apotek 99, setelah itu *customer* menanyakan produk yang ingin dibeli. Pemilik atau kasir akan menjawab ketersediaan produk yang ditanyakan *customer*, atau bila ragu,

produk akan dicek terlebih dahulu pada etalase toko. Setelah itu, pemilik atau kasir akan membuka sistem, untuk mengecek harga produk, kemudian menyampaikan informasi harga pada *customer*. Apabila *customer* setuju untuk membeli produk, maka *customer* akan membayar sejumlah uang yang sesuai dengan harga dan kuantitas produk, dan *customer* menerima produk dan uang kembali(bila ada). Halaman transaksi dapat dibiarkan dan ditambah/diubah/dihapus secara berkala sesuai kebutuhan *customer*. Apabila *customer* ingin mendapatkan struk bukti transaksi, maka pemilik atau kasir harus mengisi seluruh produk yang *customer* beli, kemudian menginput *discount* dan lain-lain, dan kolom Bayar. Sistem akan menghitung data jumlah, total, dan kembali. Sistem akan menanyakan apakah transaksi ingin dicetak atau tidak. Pemilik atau kasir menekan cetak, dan sistem akan mencetak struk. Struk diberikan kepada *customer* yang membutuhkan. Sistem akan menanyakan apakah ingin melakukan transaksi lagi. Jika Ya, maka halaman transaksi akan dibuat dari awal, dan bila Tidak, akan dialihkan ke Menu. Stok produk akan berkurang sesuai dengan produk yang dipesan.

Aktivitas transaksi pada *marketplace* milik Apotek 99 dimulai dengan *customer* membeli sejumlah produk yang dibutuhkan. Apabila pesanan sudah dibayar, maka pesanan akan masuk ke dalam sistem mitra. Apotek mempunyai waktu 1x24 jam(kurir biasa) atau 18 jam(kurir instan) untuk memutuskan terima pesanan atau tolak. Apabila diterima, apotek mempunyai waktu 2x24 jam(kurir biasa) atau H+1 dengan maksimal pukul 16.00 WIB(bila lewat, maka diproses di hari berikutnya) untuk konfirmasi pengiriman, mulai dari input resi dan/atau *request pickup*. Apotek bekerjasama dengan dua mitra, sehingga terdapat dua lapak Tokopedia yang berbeda.

3.3.1 Kendala Sistem Saat ini

Terdapat beberapa kendala yang saat ini dihadapi oleh Apotek 99 ;

1. Program VM 1.0 berjalan di perangkat dengan *local storage*, dan apabila perangkat mengalami kerusakan, maka seluruh data dalam program berpotensi hilang.

2. Program VM 1.0 belum terintegrasi dengan sistem lain, sehingga proses pembaruan data produk harus dilakukan secara manual ke sistem *marketplace*.
3. Pemilik dan karyawan tidak selalu memeriksa dan mengupdate stok produk di *marketplace*, sehingga dapat membuat stok produk pada *marketplace* menjadi tidak *up-to-date* dengan stok produk di gudang.
4. Terdapat risiko untuk apotek memperoleh pesanan *overselling*, apabila stok produk di *marketplace* tidak diperbarui dalam jangka waktu yang lama, membuat apotek harus membutuhkan waktu lebih untuk memproses pesanan, atau terpaksa membatalkan pesanan. Pembatalan pesanan *marketplace* dapat membuat reputasi apotek pada *marketplace* menurun.
5. Sistem VM 1.0 tidak dapat mengelompokkan produk berdasarkan tanggal *expired*, karena keterbatasan pada form penambahan atau perubahan data produk, yang hanya dapat mencantumkan tanggal terbaru *expired*, sehingga apotek dapat mengalami kerugian finansial, bila produk telah *expired* terlebih dahulu, sebelum terjual atau ditukar dengan keluaran terbaru.
6. Pemilik dan karyawan terkadang mengalami kesulitan dalam memahami dan menggunakan sistem dari mitra, sehingga tidak dapat memanfaatkan seluruh potensi fitur yang disediakan sistem.
7. Penambahan data produk di *marketplace* bergantung pada *database* produk milik mitra. Oleh karena itu, apotek tidak dapat menjual produk terbaru di *marketplace*, jika mitra belum menambahkan data produk tersebut ke dalam *database*. Selain itu, apotek juga tidak dapat menambahkan produk ke dalam *database* mitra.

BAB IV

ANALISIS DAN PERANCANGAN SISTEM USULAN

4.1 Analisis Kelayakan Sistem

Dalam analisis kelayakan sistem, terdapat tiga bagian, yang terdiri dari dari kelayakan teknis (*technical feasibility*), kelayakan operasional (*operational feasibility*), dan kelayakan organisasi (*organizational feasibility*).

4.1.1 Kelayakan Teknis

Analisis kelayakan teknis adalah proses pengidentifikasian risiko-risiko yang akan terjadi, untuk menentukan apakah sistem tersebut dapat dibangun atau tidak, secara teknis. Secara teknis, sistem web API Integrator yang akan dibangun pada Apotek 99 memiliki beberapa risiko yang akan dihadapi, dalam penerapannya, yaitu:

1) Risiko mengenai pengenalan aplikasi adalah sedang;

Apotek 99 saat ini menggunakan sistem yang belum terintegrasi dengan sistem atau teknologi lain, untuk mengelola data produk dan transaksi pada toko. Adanya sistem ini akan menjadi suatu hal yang baru bagi apotek tersebut, sehingga ada peluang risiko pemilik apotek tidak mengenal sebagian fitur-fitur pada aplikasi ini. Maka dari itu, risiko mengenai pengenalan aplikasi adalah sedang. Diperlukan pelatihan kepada pemilik apotek agar mengenal dan mengerti tentang penggunaan fitur-fitur yang disediakan, serta dapat menggunakan sistem yang dirancang secara optimal.

2) Risiko mengenai pengenalan teknologi adalah rendah;

Penggunaan dari sistem usulan ini adalah pemilik apotek, dimana pemilik apotek memiliki pengetahuan dan latar belakang yang cukup

tentang teknologi komputer dan perangkat lunak kasir, sehingga risiko mengenai pengenalan teknologi adalah rendah.

3) Risiko pembangunan aplikasi adalah sedang;

Sistem usulan akan digunakan pada apotek yang memiliki kebutuhan untuk mengelola stok produk pada konvensional dan *marketplace*, mengelompokkan produk berdasarkan tanggal *expired*, serta memperbarui stok *batch* produk secara otomatis, melalui *API*, setiap adanya transaksi penjualan baru. Maka dari itu, risiko pembangunan aplikasi adalah sedang karena keterbatasan waktu untuk memenuhi kebutuhan tersebut.

Berdasarkan analisis dari risiko-risiko tersebut, dapat disimpulkan bahwa kelayakan teknis dari usulan sistem *web API integrator* memiliki tingkat risiko sedang. Sistem usulan layak untuk dilaksanakan, karena manfaat yang diperoleh lebih besar dari risiko yang dihadapi.

4.1.2 Kelayakan Ekonomi

Analisis kelayakan ekonomi adalah proses identifikasi risiko keuangan terkait dengan pembangunan sistem, dengan cara perhitungan biaya yang dibutuhkan untuk membangun sistem yang diusulkan.

Sistem usulan yang dirancang diharapkan dapat membantu Apotek 99 dalam mempermudah beberapa proses bisnis seperti pencatatan data transaksi, melakukan perhitungan pesanan produk secara otomatis, dan proses lainnya. Berikut merupakan spesifikasi perangkat keras dan perangkat lunak minimal yang dibutuhkan dalam penerapan sistem usulan:

Tabel 4. 1 Spesifikasi Minimum Perangkat Sistem Usulan untuk Pengembang

Perangkat Keras	
<i>Processor</i>	Intel Core i3-5005U
<i>RAM</i>	4 GB
<i>Harddisk space</i>	100 GB
Perangkat Lunak	
<i>Sistem operasi</i>	Windows 11 build 22621
<i>Database</i>	MariaDB versi 10.3
<i>Web server</i>	Apache versi 2.4.54
<i>Browser</i>	Google Chrome versi 109

Tabel 4. 2 Spesifikasi Minimum Perangkat Sistem Usulan untuk *Server*

Perangkat Keras	
<i>RAM</i>	512 MB
<i>Harddisk space</i>	1 GB
Perangkat Lunak	
<i>Sistem operasi</i>	Linux versi 3.10
<i>Database</i>	MariaDB versi 10.3
<i>Web server</i>	Apache versi 2.4.54
<i>Web hosting</i>	cPanel versi 106

Sistem usulan yang dirancang berjalan dalam sebuah *web server* bernama IDCloudHost, dengan biaya *subscription* sebesar Rp. 15.000,- per bulan. Server mulai diaktifkan sejak tanggal 9 September 2022, dan telah menghabiskan biaya sebesar Rp. 75.000,- per bulan Januari 2023. Berdasarkan biaya yang dikeluarkan, sistem usulan ini layak untuk dibangun dalam aspek ekonomi, karena memberikan manfaat yang lebih besar daripada biaya yang dikeluarkan.

4.1.3 Kelayakan Organisasi

Analisis kelayakan organisasi adalah proses analisis yang bertujuan untuk mengetahui seberapa baik sistem, apakah sistem dapat diterima oleh pengguna dan digunakan dalam operasi organisasi yang sedang berlangsung. Dari sisi kelayakan organisasi, sistem usulan ini memiliki tingkat risiko rendah, dikarenakan sistem yang dirancang sesuai dengan kebutuhan Apotek 99 dan informasi terkait kebutuhan sistem di Apotek 99 didapatkan langsung dari pemilik apotek. Sistem *Web API Integrator* ini bertujuan dalam mensinkronisasikan stok produk antara toko dan *marketplace*, namun tetap mempertahankan fitur yang terdapat pada sistem sebelumnya, yakni penambahan transaksi pada konvensional.

4.2 Tahap Analisis

Tahap kedua adalah tahap analisis. Pada tahap ini, tercakup kebutuhan pengguna berdasarkan *requirement*. *Requirement* didapatkan dari hasil pengelolaan *system request* berdasarkan sudut pandang pengguna. *System requirement* akan menjadi gambaran terkait karakteristik yang harus dimiliki oleh sistem dan apa yang bisa dilakukan oleh sistem tersebut. Ada dua jenis *system requirement*, yaitu *functional requirement* dan *non-functional requirement*.

4.2.1 Functional Requirement

Functional requirement menyangkut proses yang harus dilakukan sistem atau informasi yang perlu disajikan, Sistem *Web API Integrator* pada Apotek 99 memiliki *functional requirement* sebagai berikut.

1. Mengelola informasi produk
Admin dapat menambah, mengubah, dan menghapus informasi produk. Kasir dapat mengakses informasi produk.
2. Mengelola *batch* produk
Admin dapat menambah, mengubah, dan menghapus *batch* produk. Kasir dapat mengakses *batch* produk.
3. Mengelola kategori produk
Admin dapat menambah, mengubah, dan menghapus kategori produk.
4. Mengelola unit produk
Admin dapat menambah, mengubah, dan menghapus unit produk.
5. Mengelola produk kadaluwarsa
Admin dan kasir dapat mengakses produk kadaluwarsa. Admin dapat mengosongkan produk yang kadaluwarsa.
6. Mengelola produk kosong
Admin dan kasir dapat mengakses produk kosong. Admin dapat menambahkan *batch* produk terbaru.
7. Mengelola transaksi toko
Admin dan kasir dapat mengakses halaman *history* transaksi toko. Admin dan kasir dapat menambahkan transaksi toko terbaru. Admin dapat menghapus transaksi toko.
8. Mencari produk di *dashboard*
Admin dapat mencari produk di *dashboard*.
9. Mengakses rangkuman data *dashboard*

Admin dapat mengakses rangkuman data dashboard, seperti data jumlah, misalnya total produk, produk kosong, dan produk *expired*, serta daftar informasi, misalnya daftar penjualan produk tertinggi, dan *batch* produk terbaru.

10. Mengakses laporan laba

Admin dapat mengakses laporan laba jual beli.

11. Mengelola akses *backdoor*

Admin dapat mengelola akses *backdoor* pada *register* dan *forgot password*.

12. Mengelola *API Key*

Admin dapat menambah, mengubah, dan menghapus *API Key*.

13. Mengelola *user*

Admin dapat menambah, mengubah data, mengubah *password* dan menghapus *user*.

14. Mengelola *role*

Admin dapat menambah, mengubah, dan menghapus *role*.

15. Mengelola pilihan menu

Admin dapat menambah, mengubah, dan menghapus *title*, menu, dan submenu. Ketiga bagian ini akan mempengaruhi tampilan pilihan menu pada *user*.

16. Mengelola profil

Admin dan kasir dapat mengelola profil, seperti mengubah nama, dan *profile picture*.

4.2.2 Non-functional Requirement

Non-functional requirement menyangkut sifat dan perilaku yang harus dimiliki sistem seperti kinerja dan kegunaannya. Sistem *Web API Integrator* pada Apotek 99 memiliki *non-functional requirement* sebagai berikut.

1) *Operational*

- a. Sistem dapat dioperasikan pada perangkat yang mendukung *web browser*.
- b. Sistem membutuhkan koneksi internet untuk dapat dijalankan.

2) *Performance*

- a. Sistem dapat diakses 24 jam.
- b. Sistem terkoneksi secara *realtime* dengan *database online* dan *SmartSeller API*.

3) *Security*

- a. Sistem hanya diakses oleh *user* yang telah terdaftar.
- b. Sistem dapat diakses oleh user dengan memasukkan e-mail dan password.
- c. Password yang disimpan di dalam database menggunakan hash md5.
- d. Password dapat diubah sesuai keinginan pengguna dengan minimum 8 karakter.
- e. Penggunaan kode autentikasi yang dikirimkan pada email pengguna untuk perubahan *password*.

4) *Cultural*

Bahasa yang digunakan pada sistem usulan ini adalah Bahasa Inggris dan Bahasa Indonesia. Bahasa Inggris digunakan pada elemen-elemen yang terdapat pada sistem seperti tombol, judul halaman, dan *field* formulir. Bahasa Indonesia digunakan dalam memasukkan data informasi produk, seperti deskripsi, kategori, unit.

4.3 Sistem Usulan

4.3.1 Pemodelan Proses Bisnis

Pada tahap ini, terdapat gambaran yang lebih terperinci, tentang bagaimana alur proses bisnis akan berjalan, dan bagaimana alur interaksi antar pengguna di dalam sistem berjalan. Hal ini dideskripsikan dalam bentuk *use case diagram* dan *activity diagram*.

4.3.1.1 Use Case Diagram

Diagram *use case* digunakan untuk mengilustrasikan fungsi-fungsi dan aktor-aktor utama dalam sistem usulan. Aktor dari sistem ini meliputi:

1. Admin

Admin merupakan *user* dari sistem usulan ini yang memiliki akses tertinggi, yang dapat melakukan tindakan, seperti:

Menghapus transaksi toko, mengelola informasi produk, mengelola *batch* produk, mengelola kategori produk, mengelola unit produk, mencari produk di *dashboard*, mengakses rangkuman data di *dashboard*, mengelola akses *backdoor*, mengelola akses menu, mengakses laporan laba, mengelola *API Key*, mengelola *user*, dan mengelola *role*. Pada sistem usulan ini, pemilik akan mendapatkan akses sebagai admin.

2. Kasir

Kasir merupakan *user* dari sistem usulan ini yang dapat melakukan beberapa tindakan, seperti:

Mengelola keranjang transaksi toko, mengelola profil, mengakses transaksi toko, mengakses *batch* kadaluwarsa, mengakses produk kosong, mengakses informasi produk, dan mengakses *batch* produk. Pada sistem usulan ini, karyawan akan mendapatkan akses sebagai kasir.

Berikut ini adalah *use case diagram* dari sistem yang diusulkan, pada Gambar 4.1.



Gambar 4. 1 Use Case Diagram Sistem Web API Integrator

4.3.1.2 Activity Diagram

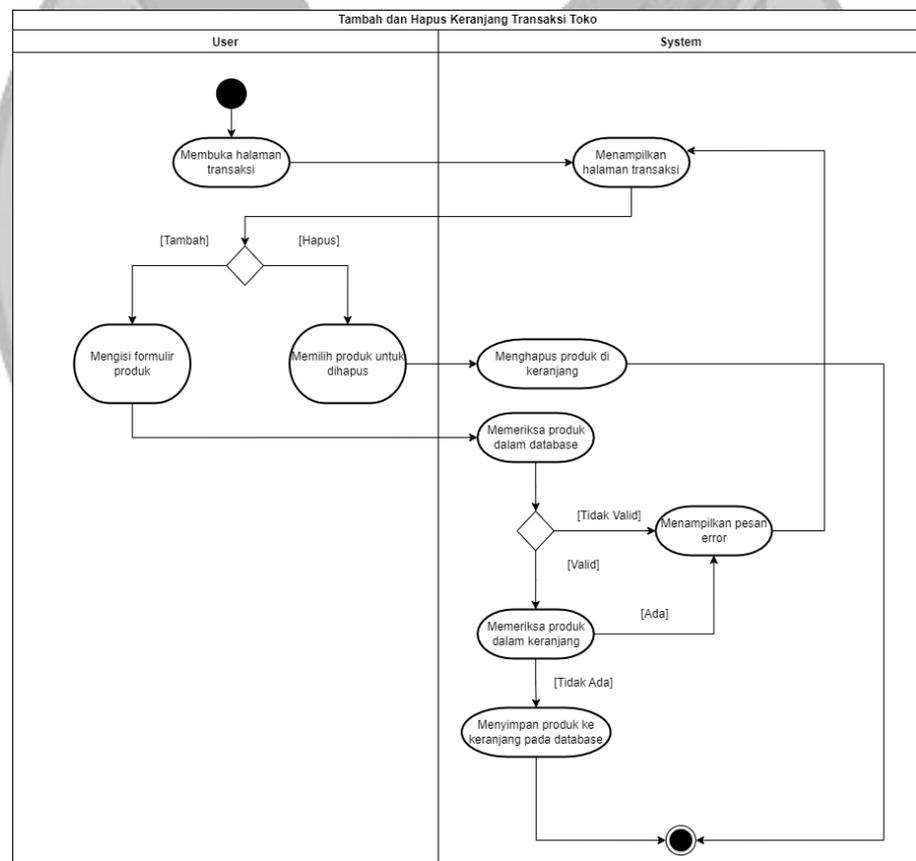
Activity diagram merupakan gambaran dari setiap proses yang berlangsung di dalam sistem. Berikut merupakan *activity diagram* dari sistem yang diusulkan:

1) Mengelola keranjang transaksi toko

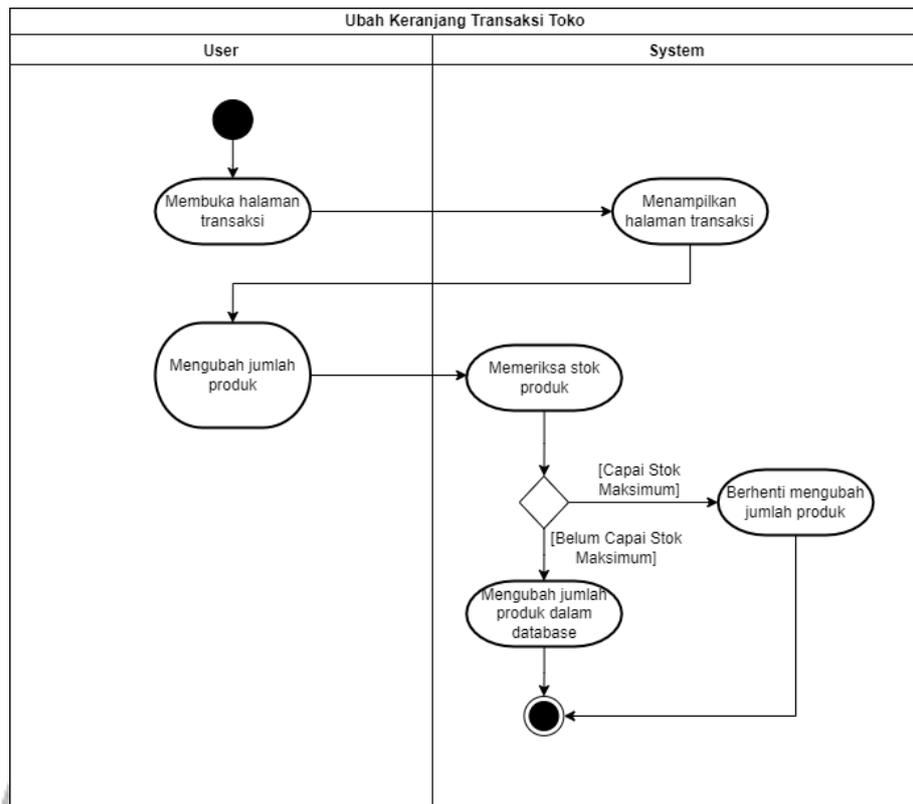
Gambar 4.2, Gambar 4.3, dan Gambar 4.4 menunjukkan *activity diagram* mengelola keranjang transaksi toko, adapun proses pengelolaan keranjang transaksi toko terdiri dari langkah-langkah berikut:

- a. *User* membuka halaman transaksi toko.
- b. Sistem menampilkan halaman transaksi toko.
- c. Terdapat 4 hal yang dapat dilakukan *user* pada fitur, yaitu tambah keranjang, ubah keranjang, hapus keranjang, dan simpan transaksi.
- d. Untuk menambah produk ke keranjang, *user* mengisi informasi produk.
- e. Sistem akan memeriksa informasi produk.
- f. Jika produk tidak ditemukan, maka sistem akan menampilkan pesan *error*.
- g. Jika produk ditemukan, maka sistem akan memeriksa produk di keranjang.
- h. Jika produk telah dimasukkan ke dalam keranjang, maka sistem akan menampilkan pesan *error*.
- i. Jika produk belum dimasukkan ke dalam keranjang, maka sistem akan menyimpan data produk ke dalam keranjang.
- j. Untuk mengubah keranjang, *user* dapat mengubah kuantitas produk di keranjang.
- k. Sistem akan memeriksa validasi stok.
- l. Jika kuantitas belum mencapai stok produk tersedia, maka sistem akan menambah kuantitas.

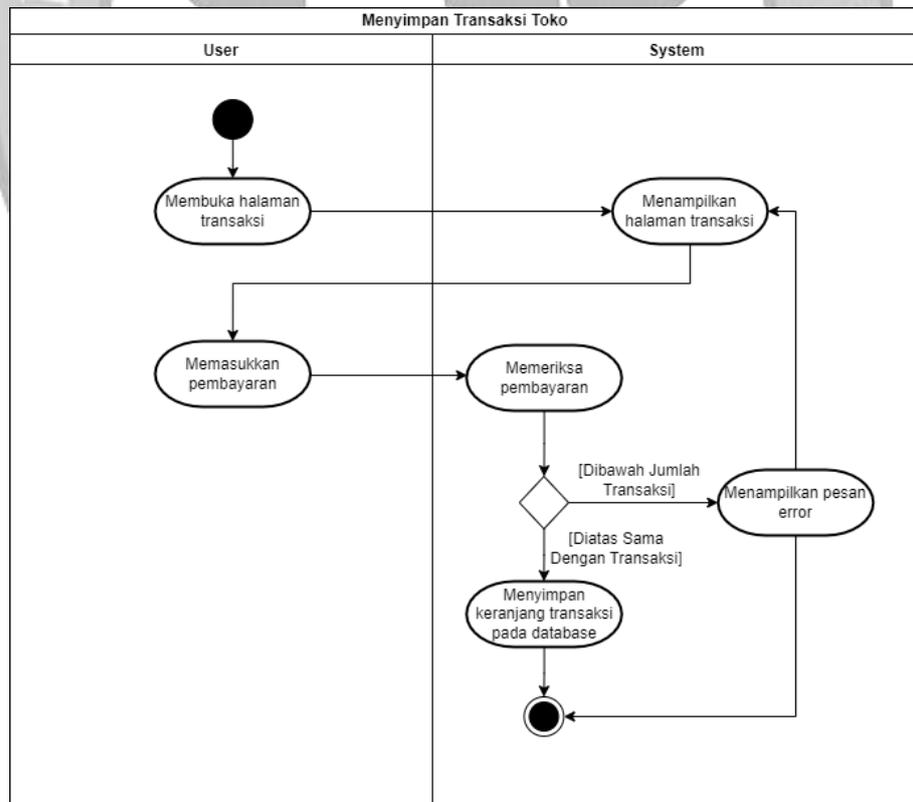
- m. Jika kuantitas telah mencapai stok produk tersedia, maka sistem akan berhenti menambah kuantitas.
- n. Untuk menghapus keranjang, *user* dapat menghapus produk di keranjang. Sistem akan menghapus produk dari keranjang.
- o. Jika *user* ingin menyimpan transaksi, *user* memasukkan pembayaran pada formulir transaksi.
- p. Sistem memeriksa pembayaran pada transaksi.
- q. Jika pembayaran dibawah dari jumlah transaksi, maka sistem akan menampilkan pesan error.
- r. Jika pembayaran diatas sama dengan jumlah transaksi, maka sistem akan menyimpan keranjang transaksi.



Gambar 4. 2 Activity Diagram Tambah dan Hapus Keranjang Transaksi Toko



Gambar 4. 3 Activity Diagram Ubah Keranjang Transaksi Toko

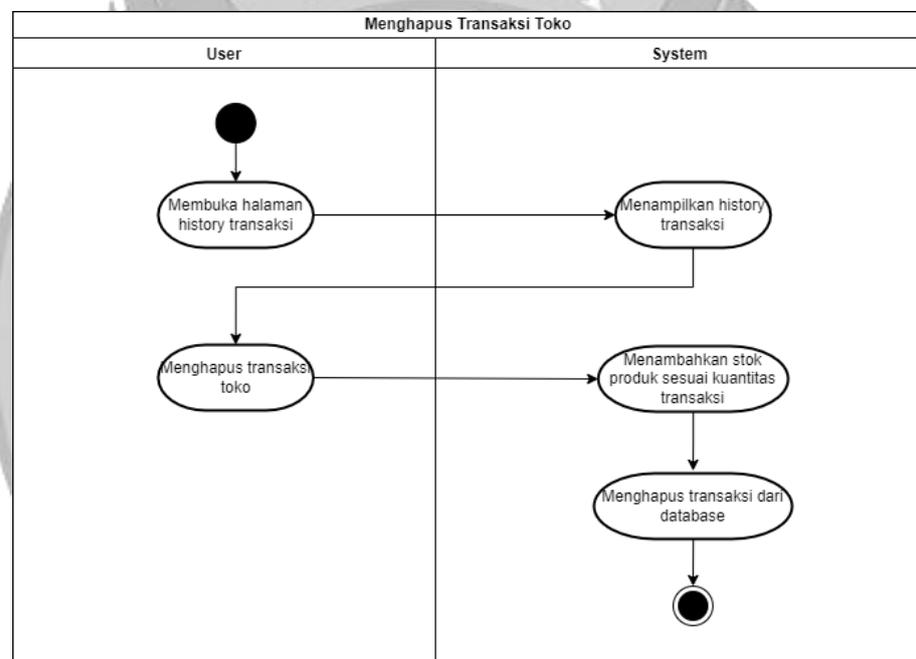


Gambar 4. 4 Activity Diagram Menyimpan Transaksi Toko

2) Menghapus transaksi toko

Gambar 4.5 menunjukkan *activity diagram* menghapus transaksi toko, adapun proses penghapusan transaksi toko terdiri dari langkah-langkah berikut:

- a. Admin membuka halaman *history* transaksi toko.
- b. Sistem menampilkan *history* transaksi toko.
- c. Admin menghapus transaksi toko.
- d. Sistem menambahkan stok produk sesuai kuantitas transaksi.
- e. Sistem menghapus transaksi.



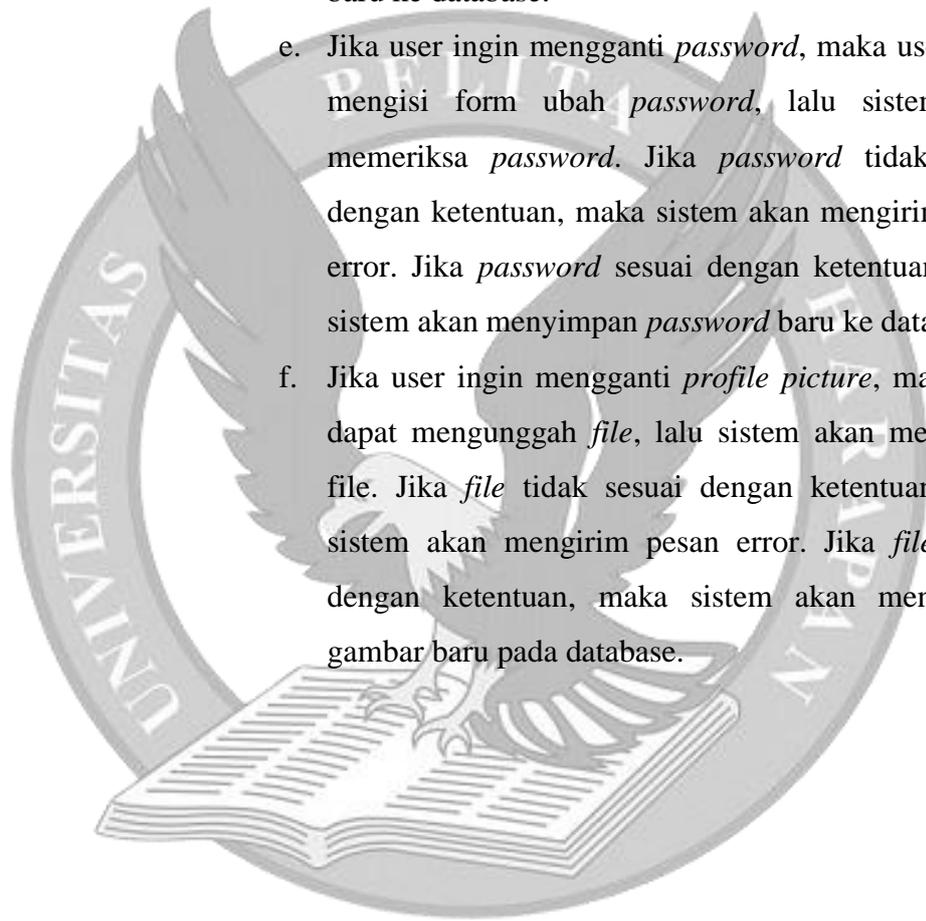
Gambar 4.5 Activity Diagram Menghapus Transaksi Toko

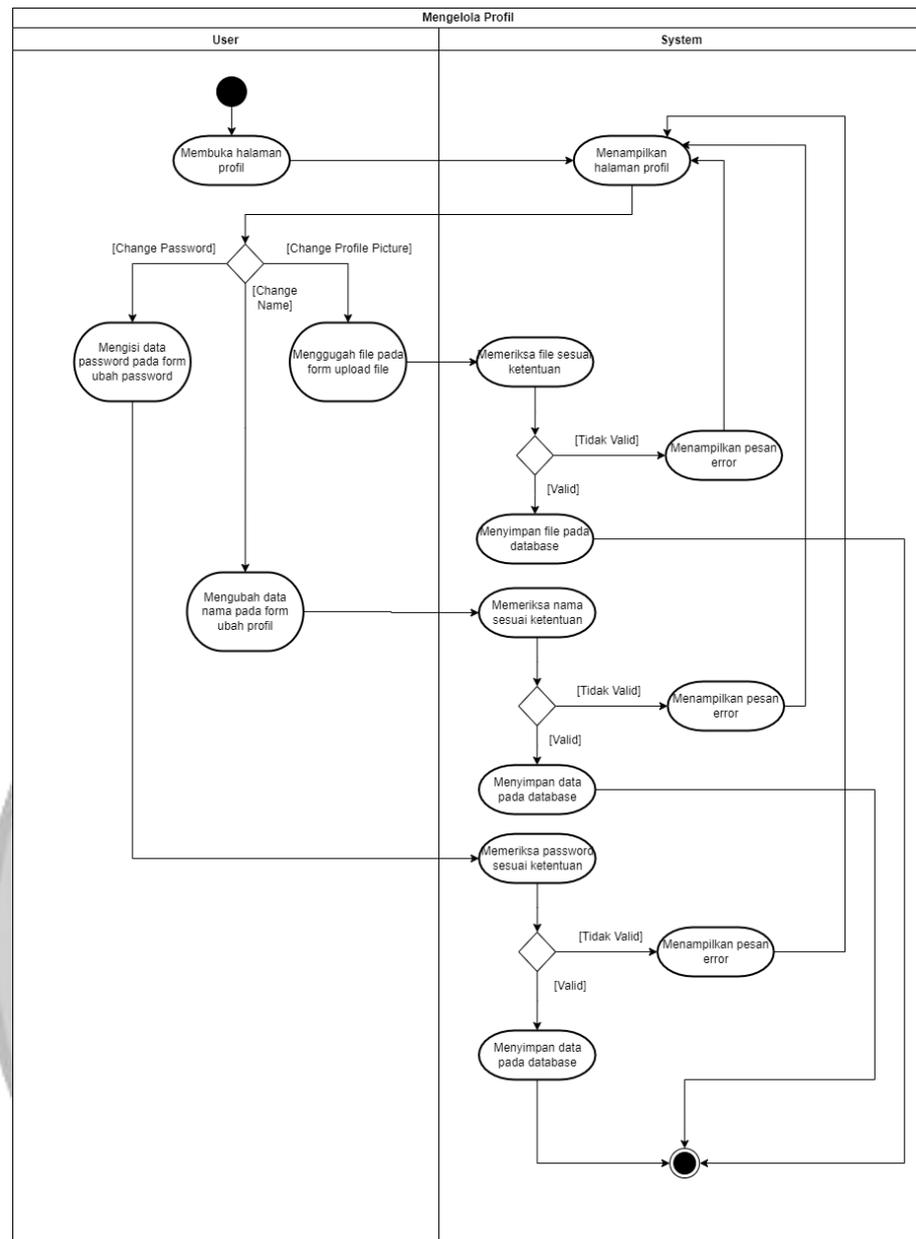
3) Mengelola profil

Gambar 4.6 menunjukkan *activity diagram* mengelola profil, adapun proses pengelolaan profil terdiri dari langkah-langkah berikut:

- a. *User* membuka halaman profil.
- b. Sistem menampilkan halaman profil.

- c. Terdapat 3 hal yang bisa dilakukan user pada profile, yaitu *change name*, *change password*, dan *change profile picture*.
- d. Jika user ingin mengganti nama, maka user dapat mengisi form ubah nama, lalu sistem akan memeriksa nama. Jika nama tidak sesuai dengan ketentuan, maka sistem akan mengirim pesan error. Jika nama sesuai dengan ketentuan, maka sistem akan menyimpan nama baru ke database.
- e. Jika user ingin mengganti *password*, maka user dapat mengisi form ubah *password*, lalu sistem akan memeriksa *password*. Jika *password* tidak sesuai dengan ketentuan, maka sistem akan mengirim pesan error. Jika *password* sesuai dengan ketentuan, maka sistem akan menyimpan *password* baru ke database.
- f. Jika user ingin mengganti *profile picture*, maka *user* dapat mengunggah *file*, lalu sistem akan memeriksa *file*. Jika *file* tidak sesuai dengan ketentuan, maka sistem akan mengirim pesan error. Jika *file* sesuai dengan ketentuan, maka sistem akan menyimpan gambar baru pada database.





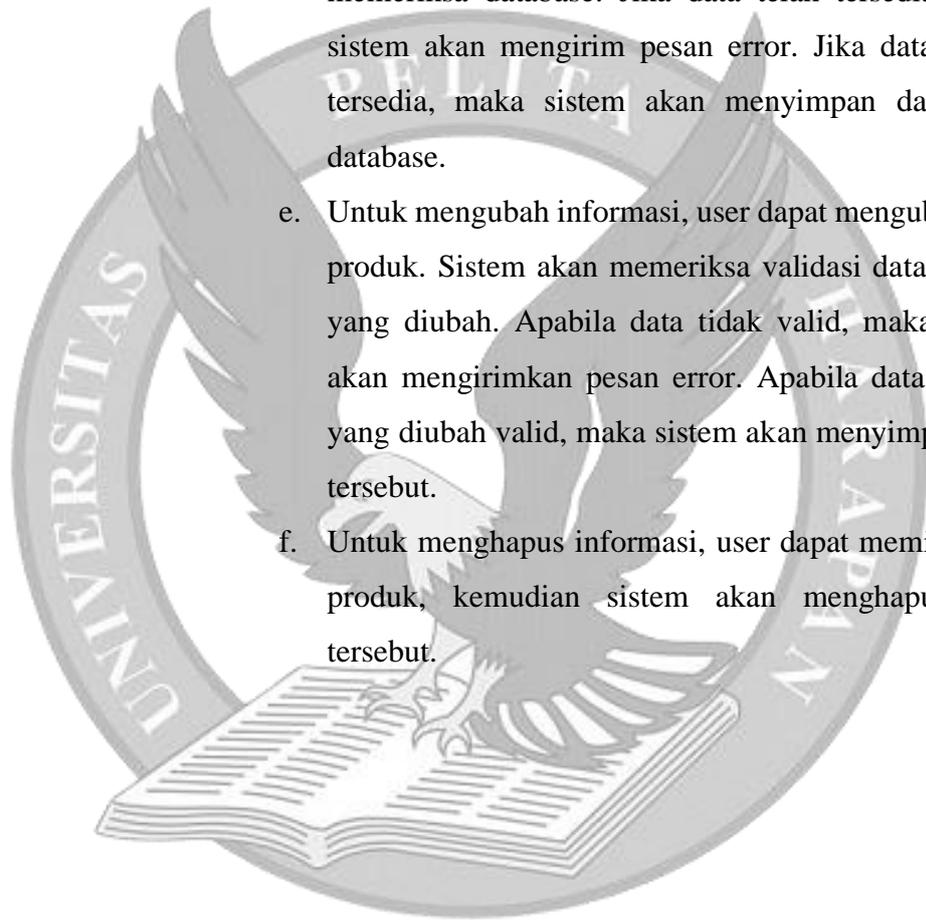
Gambar 4. 6 Activity Diagram Mengelola Profil

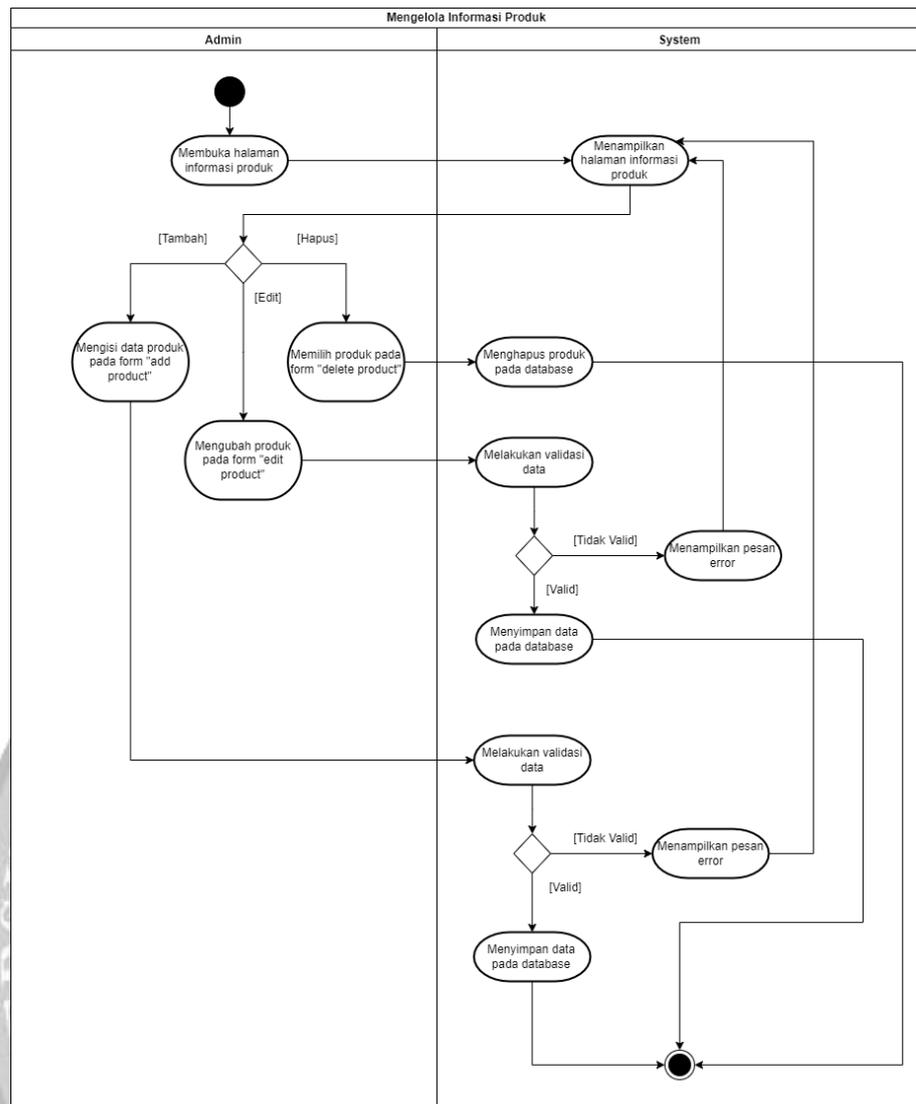
4) Mengelola informasi produk

Gambar 4.7 menunjukkan *activity diagram* untuk mengelola informasi produk, adapun proses pengelolaan informasi produk terdiri dari langkah-langkah berikut:

- a. User membuka halaman informasi produk.
- b. Sistem menampilkan halaman informasi produk.

- c. Terdapat 3 hal yang dapat dilakukan admin, yaitu menambah informasi, mengubah informasi, dan menghapus informasi.
- d. Untuk menambah produk, admin dapat mengisi formulir data produk. Sistem akan memeriksa data produk yang ditambah. Apabila data tidak sesuai dengan ketentuan, maka sistem akan mengirim pesan *error*. Apabila data sesuai ketentuan, maka sistem akan memeriksa database. Jika data telah tersedia, maka sistem akan mengirim pesan *error*. Jika data belum tersedia, maka sistem akan menyimpan data pada database.
- e. Untuk mengubah informasi, user dapat mengubah data produk. Sistem akan memeriksa validasi data produk yang diubah. Apabila data tidak valid, maka sistem akan mengirimkan pesan *error*. Apabila data produk yang diubah valid, maka sistem akan menyimpan data tersebut.
- f. Untuk menghapus informasi, user dapat memilih data produk, kemudian sistem akan menghapus data tersebut.





Gambar 4.7 Activity Diagram Mengelola Informasi Produk

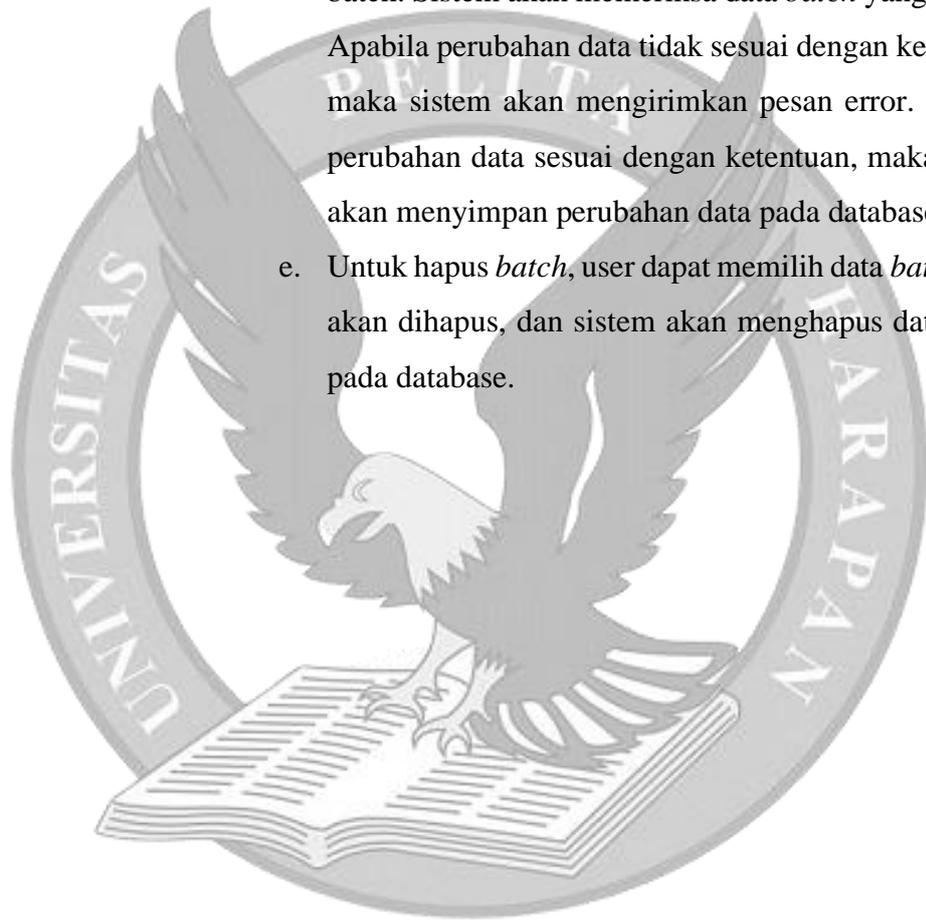
5) Mengelola *batch* produk

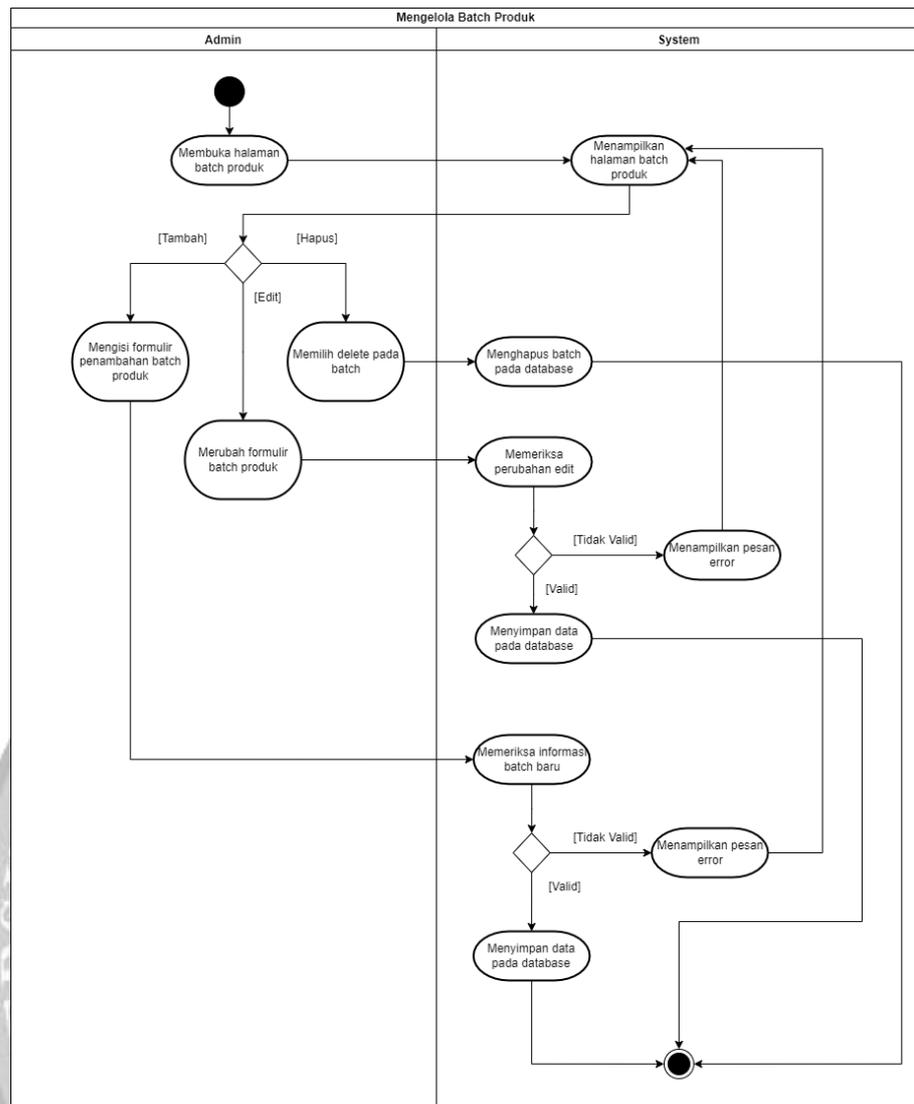
Gambar 4.8 menunjukkan *activity diagram* untuk mengelola *batch* produk, adapun proses pengelolaan produk terdiri dari langkah-langkah berikut:

- a. *User* membuka halaman *batch* produk.
- b. Terdapat 3 hal yang dapat dilakukan admin, yaitu menambah *batch*, mengubah *batch*, dan menghapus *batch*.
- c. Untuk menambah *batch*, admin dapat membuka formulir penambahan *batch*. Sistem menampilkan

formulir tambah *batch*, lalu admin mengisi form tambah *batch*. Sistem akan memeriksa data *batch*. Jika data *batch* tidak sesuai dengan ketentuan, maka sistem akan mengirim pesan error. Jika data *batch* sesuai dengan ketentuan, maka sistem akan menyimpan *batch* baru pada database.

- d. Untuk mengubah *batch*, *user* dapat memilih *batch* yang akan diubah, dan *user* dapat mengubah form data *batch*. Sistem akan memeriksa data *batch* yang diubah. Apabila perubahan data tidak sesuai dengan ketentuan, maka sistem akan mengirimkan pesan error. Apabila perubahan data sesuai dengan ketentuan, maka sistem akan menyimpan perubahan data pada database.
- e. Untuk hapus *batch*, *user* dapat memilih data *batch* yang akan dihapus, dan sistem akan menghapus data *batch* pada database.



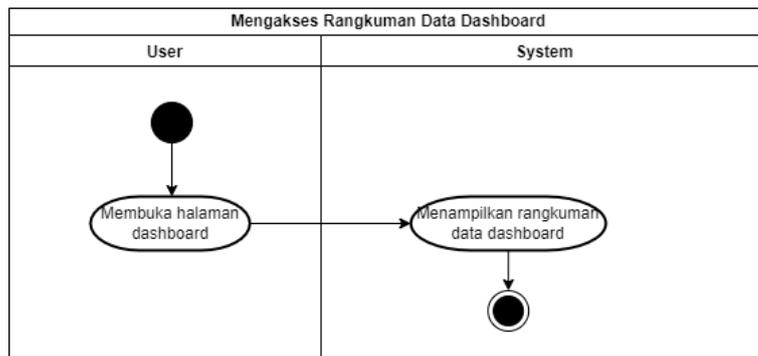


Gambar 4. 8 *Activity Diagram* Mengelola Batch Produk

6) Mengakses rangkuman data *dashboard*

Gambar 4.9 menunjukkan *activity diagram* untuk mengakses rangkuman data *dashboard*, adapun proses akses rangkuman data *dashboard* terdiri dari langkah-langkah berikut:

- a. *User* membuka halaman *dashboard*
- b. Sistem menampilkan rangkuman data *dashboard*

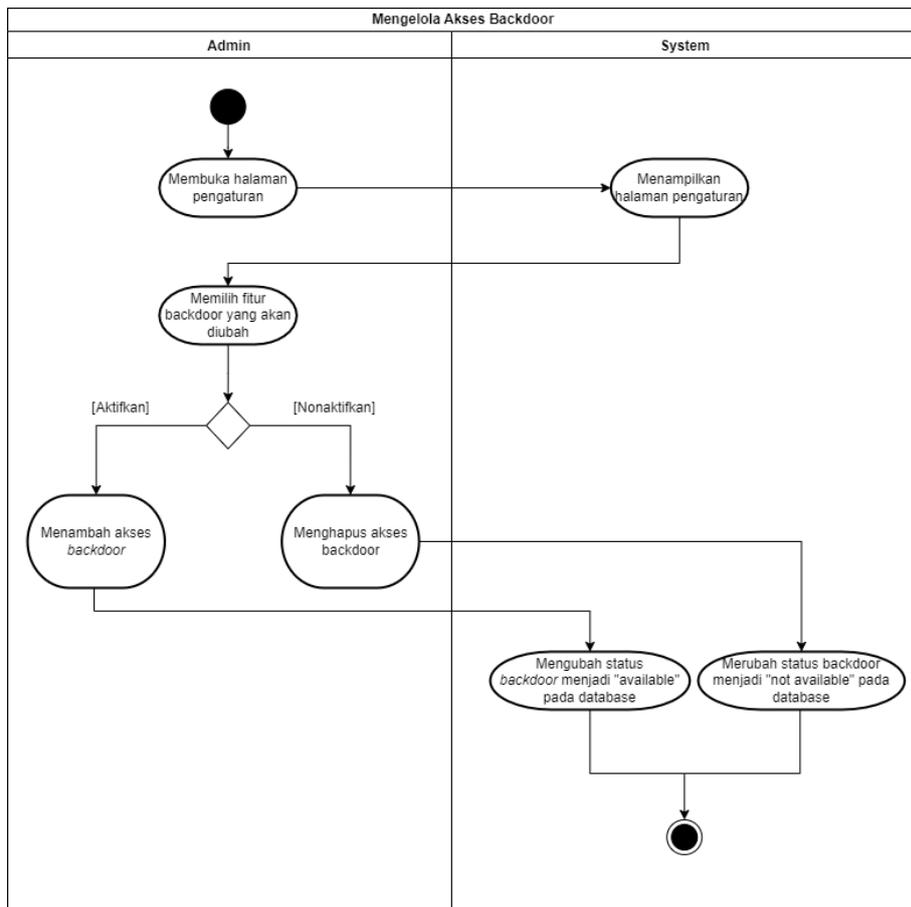


Gambar 4. 9 Activity Diagram Mengelola Akses Menu

7) Mengelola akses *backdoor*

Gambar 4.10 menunjukkan *activity diagram* untuk mengakses pengelolaan akses *backdoor* untuk *register* dan *forgot password*, adapun proses pengelolaan akses *backdoor* terdiri dari langkah-langkah berikut:

- a. Admin membuka halaman pengaturan.
- b. Sistem menampilkan halaman pengaturan.
- c. Admin memilih fitur *backdoor* yang akan diubah.
- d. Jika admin ingin mengaktifkan halaman, dapat menambah akses ke *backdoor*.
- e. Jika admin ingin menonaktifkan halaman, dapat menghapus akses ke *backdoor*.
- f. Sistem mengubah data akses halaman *backdoor*.



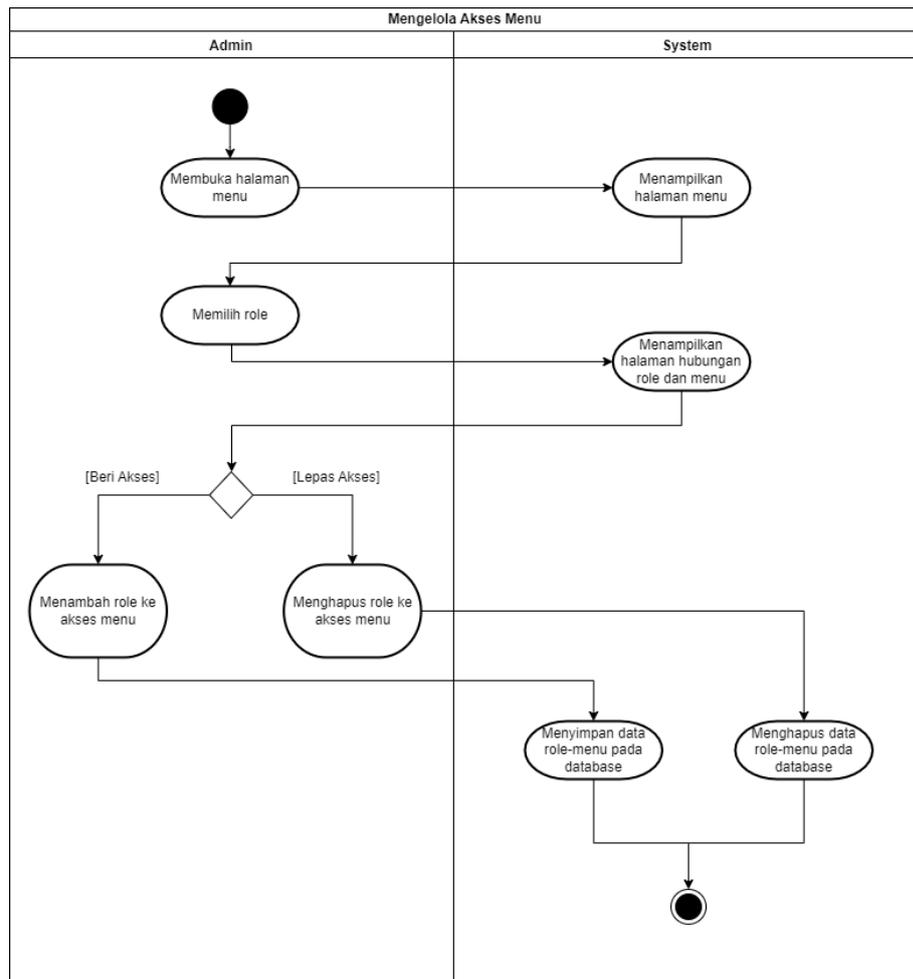
Gambar 4. 10 Activity Diagram Mengelola Akses Backdoor

8) Mengelola akses menu

Gambar 4.11 menunjukkan *activity diagram* untuk mengakses pengelolaan akses menu, adapun proses pengelolaan akses menu terdiri dari langkah-langkah berikut:

- a. Admin membuka halaman menu.
- b. Sistem menampilkan halaman menu.
- c. Admin memilih *role*.
- d. Sistem menampilkan halaman hubungan *role* dan menu.
- e. Jika admin ingin memberi akses ke *role*, dapat menambah *role* ke akses menu.
- f. Sistem menyimpan data *role-menu* pada database.
- g. Jika admin ingin melepas akses ke *role*, dapat menghapus *role* ke akses menu.

h. Sistem menghapus data *role-menu* pada database.



Gambar 4. 11 Activity Diagram Mengelola Akses Menu

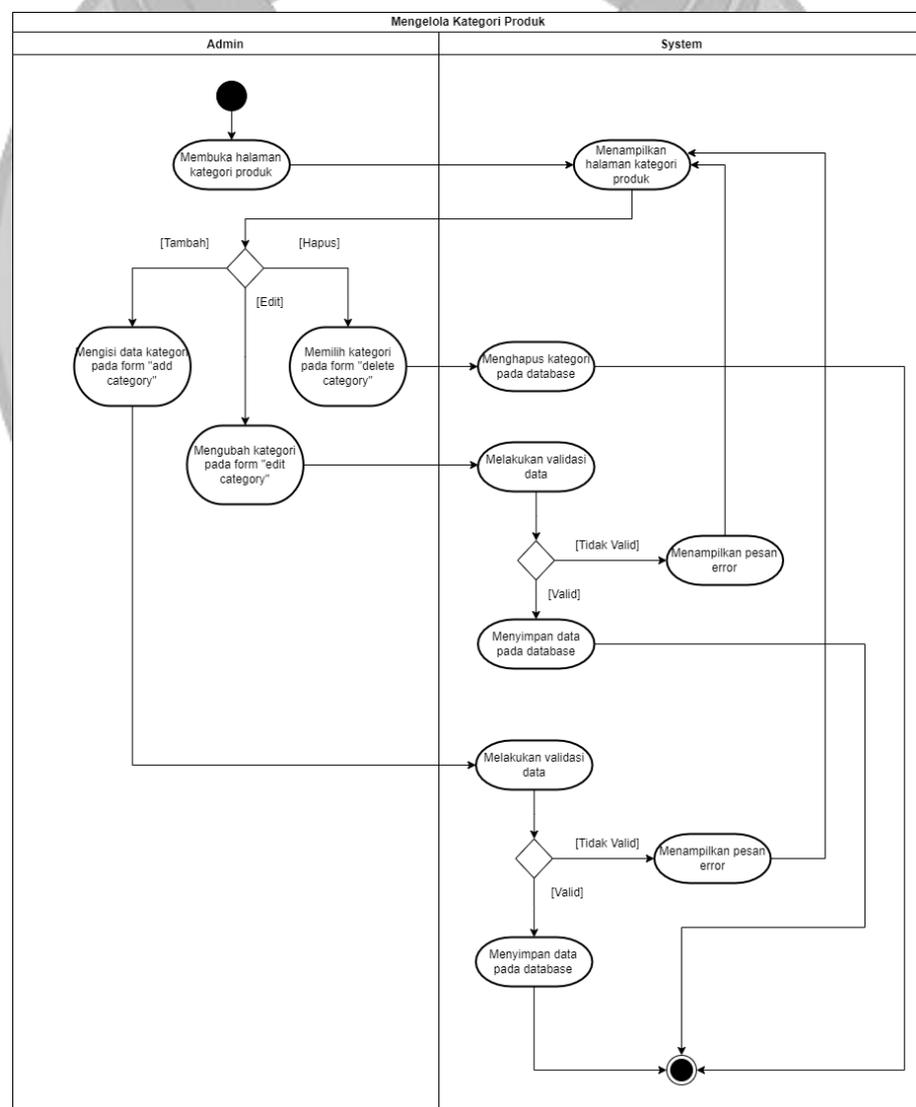
9) Mengelola kategori produk

Gambar 4.12 menunjukkan *activity diagram* mengelola kategori produk, adapun proses pengelolaan kategori terdiri dari langkah-langkah berikut:

- Admin membuka halaman kategori.
- Sistem menampilkan halaman kategori.
- Terdapat 3 hal yang dapat dilakukan admin pada fitur, yaitu *add data category*, *edit data category*, dan *delete data category*.
- Untuk menambah data kategori, admin dapat mengisi form tambah data kategori, lalu sistem akan melakukan validasi dengan data. Jika data tidak valid, maka sistem

akan menampilkan pesan error. Jika data valid, maka sistem akan menyimpan data tersebut.

- e. Untuk mengedit data kategori, admin dapat mengubah form edit data kategori, lalu sistem akan melakukan validasi dengan data. Jika data yang diubah tidak valid, maka sistem akan menampilkan pesan error. Jika data valid, maka sistem akan mengubah dan menyimpan data tersebut.
- f. Untuk menghapus data kategori, admin dapat memilih data kategori, kemudian sistem akan menghapus data kategori yang dipilih.

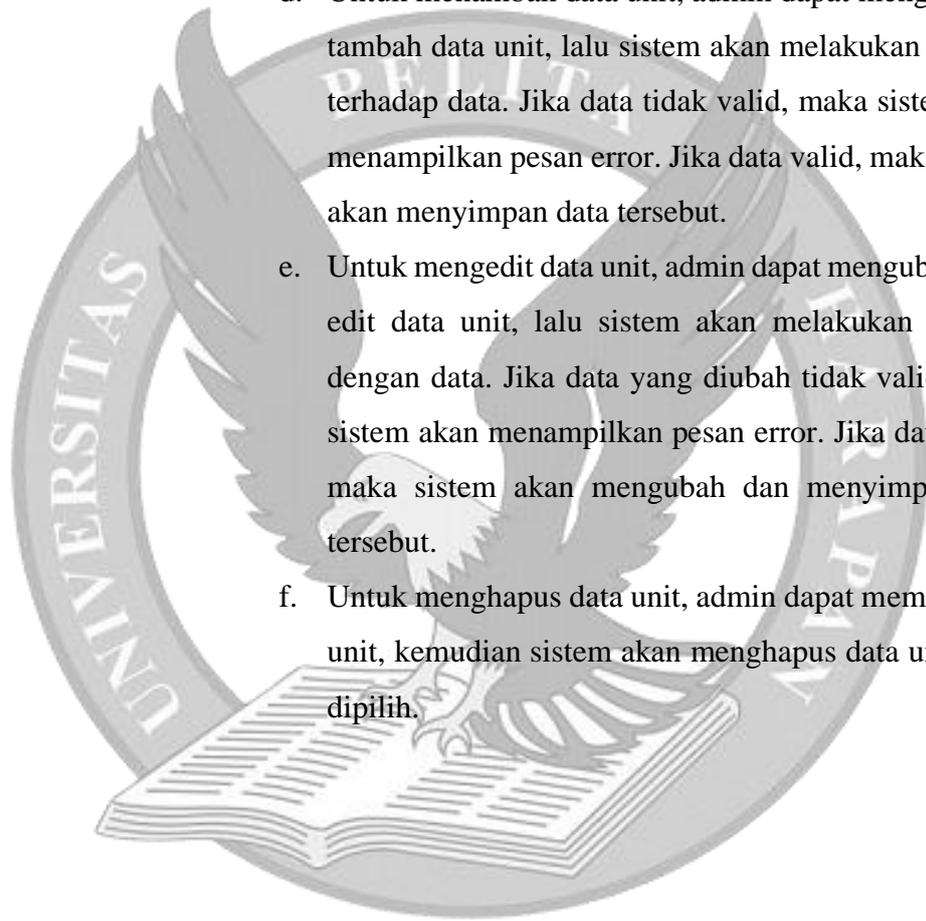


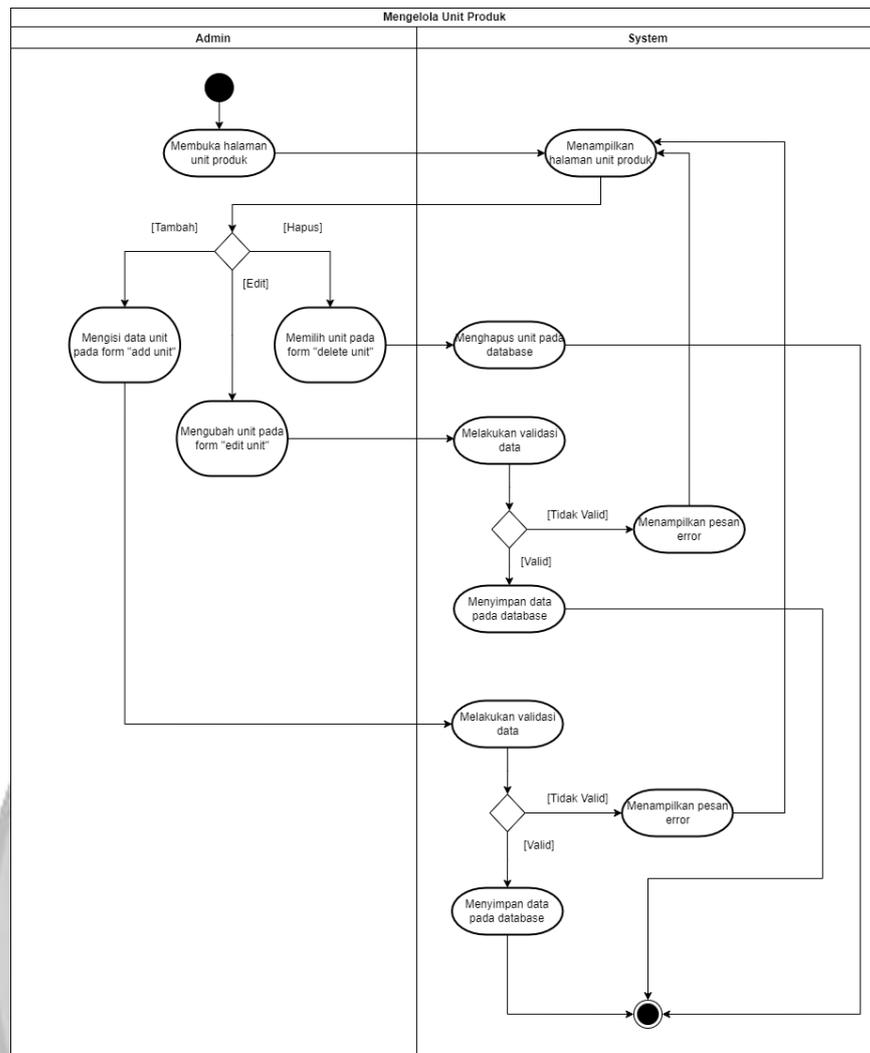
Gambar 4. 12 Activity Diagram Mengelola Kategori Produk

10) Mengelola unit produk

Gambar 4.13 menunjukkan *activity diagram* mengelola unit produk, adapun proses pengelolaan unit terdiri dari langkah-langkah berikut:

- a. Admin membuka halaman unit.
- b. Sistem menampilkan halaman unit.
- c. Terdapat 3 hal yang dapat dilakukan admin pada fitur, yaitu *add data unit*, *edit data unit*, dan *delete data unit*.
- d. Untuk menambah data unit, admin dapat mengisi form tambah data unit, lalu sistem akan melakukan validasi terhadap data. Jika data tidak valid, maka sistem akan menampilkan pesan error. Jika data valid, maka sistem akan menyimpan data tersebut.
- e. Untuk mengedit data unit, admin dapat mengubah form edit data unit, lalu sistem akan melakukan validasi dengan data. Jika data yang diubah tidak valid, maka sistem akan menampilkan pesan error. Jika data valid, maka sistem akan mengubah dan menyimpan data tersebut.
- f. Untuk menghapus data unit, admin dapat memilih data unit, kemudian sistem akan menghapus data unit yang dipilih.





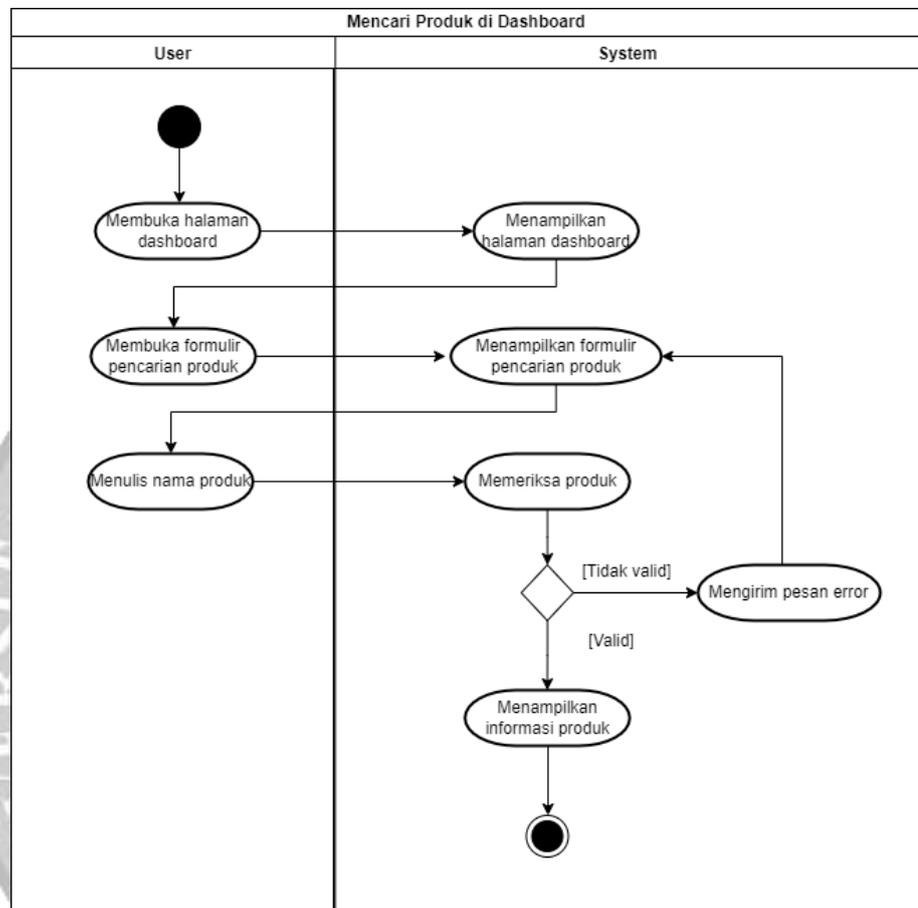
Gambar 4. 13 Activity Diagram Mengelola Unit Produk

11) Mencari produk di *dashboard*

Gambar 4.14 menunjukkan *activity diagram* mencari produk di *dashboard*, adapun proses pencarian produk pada *dashboard* terdiri dari langkah-langkah berikut:

- a. Admin membuka *dashboard*.
- b. Sistem menampilkan *dashboard*.
- c. Admin membuka formulir pencarian produk.
- d. Sistem menampilkan formulir pencarian produk.
- e. Admin menulis nama produk.
- f. Sistem akan memeriksa nama produk.

- g. Jika produk tidak tersedia, maka sistem mengirim pesan *error*.
- h. Jika produk tersedia, maka sistem menampilkan informasi produk.

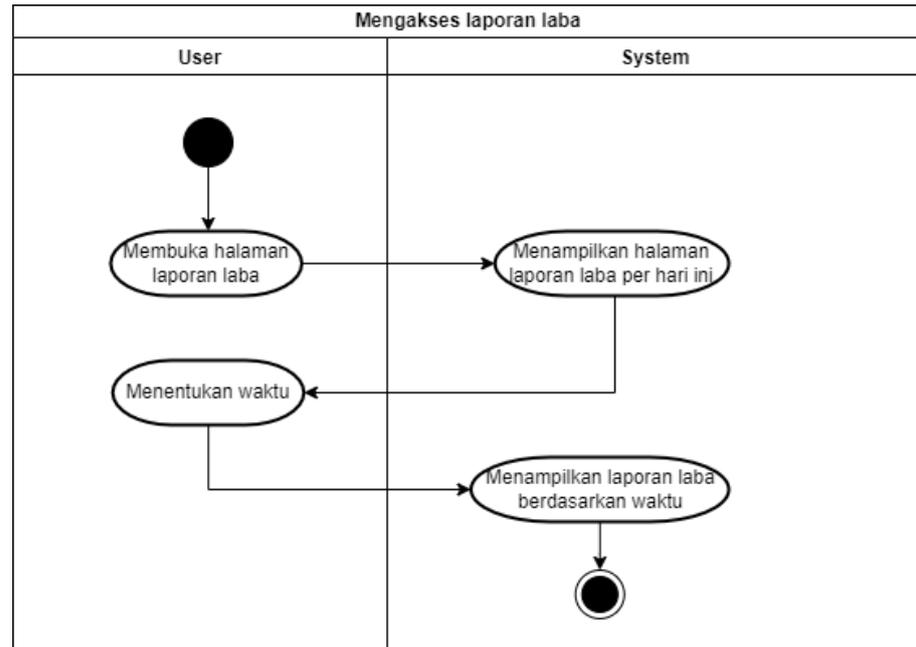


Gambar 4. 14 Activity Diagram Mencari Produk di Dashboard

12) Mengakses laporan laba

Gambar 4.15 menunjukkan *activity diagram* mengelola laporan laba, adapun proses pengelolaan laporan laba terdiri dari langkah-langkah berikut:

- a. Admin membuka halaman laporan laba.
- b. Sistem menampilkan halaman laporan laba per hari ini.
- c. Admin dapat menentukan waktu.
- d. Sistem menampilkan data laporan laba.



Gambar 4. 15 Activity Diagram Mengakses Laporan Laba

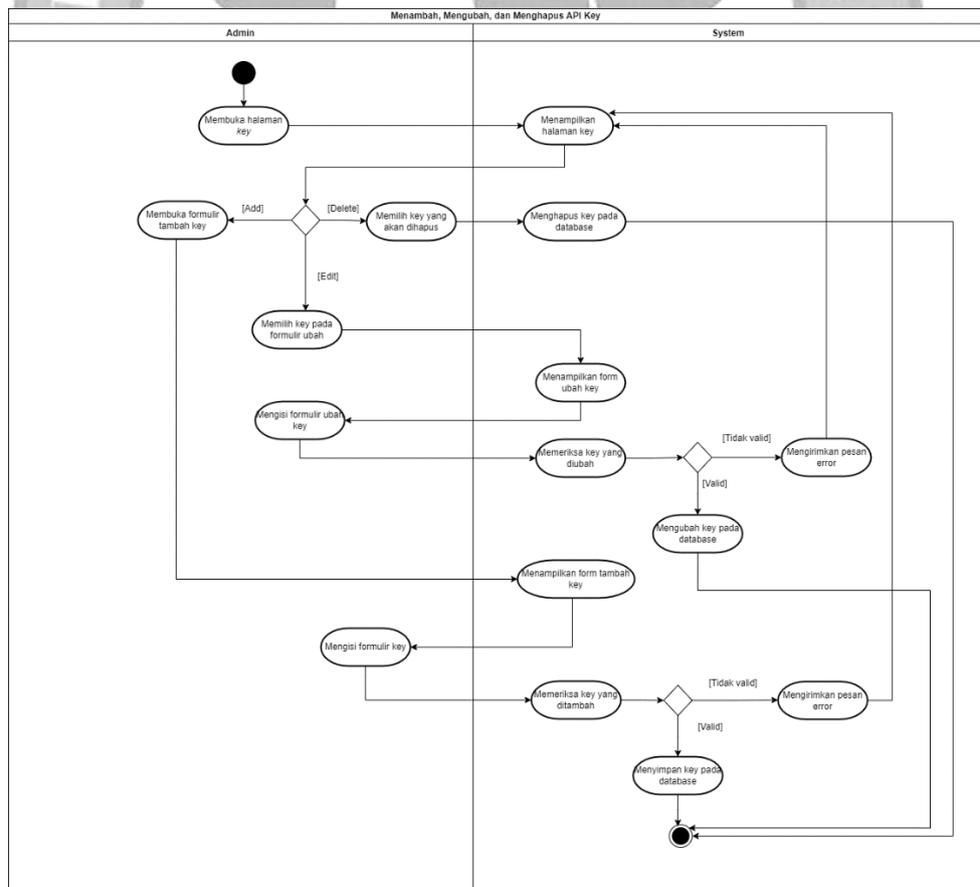
13) Mengelola API Key

Gambar 4.16 dan Gambar 4.17 menunjukkan *activity diagram* mengelola API key, adapun proses pengelolaan key terdiri dari langkah-langkah berikut:

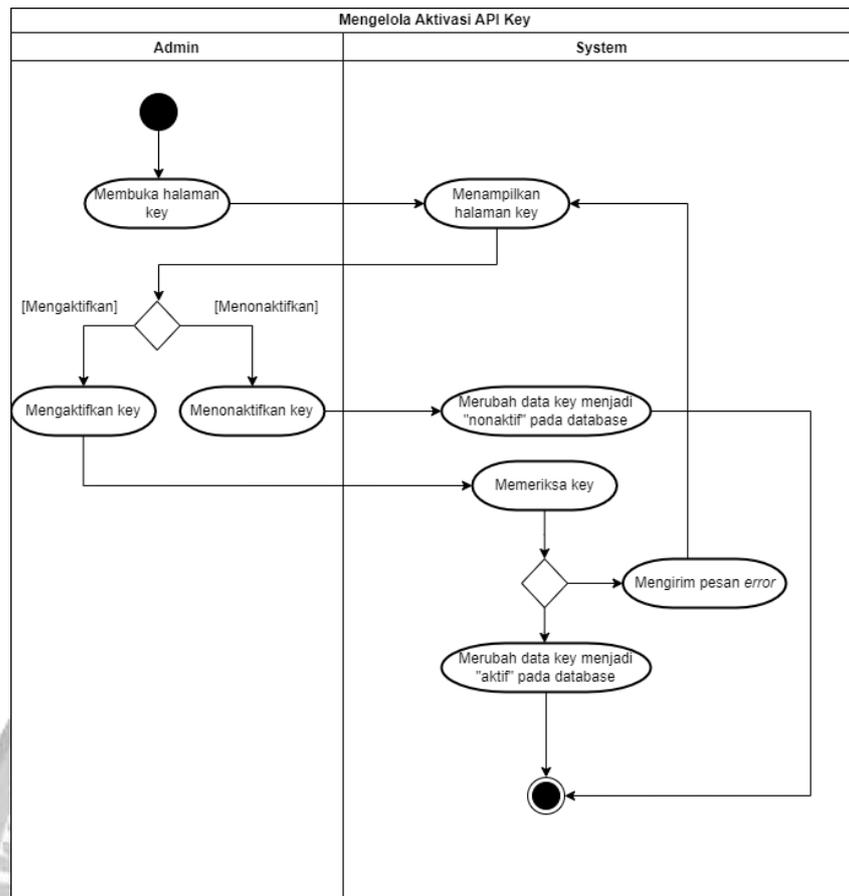
- a. Admin membuka halaman key.
- b. Sistem menampilkan halaman key.
- c. Terdapat 4 hal yang dapat dilakukan admin, yaitu menambah key, mengubah data key, mengelola status aktif key, dan menghapus key.
- d. Jika admin ingin menambah key, maka admin dapat membuka formulir penambahan key. Sistem menampilkan formulir penambahan key. Admin mengisi key. Sistem akan melakukan pengecekan key. Jika key tidak valid, maka sistem akan mengirim pesan *error*. Jika key valid, maka sistem akan menyimpan key pada *database*.
- e. Jika admin ingin mengubah key, maka admin dapat memilih key yang akan diubah. Sistem menampilkan

formulir perubahan *key*. Admin mengubah *key* pada formulir Sistem akan melakukan pengecekan pada perubahan *key*. Jika *key* tidak valid, maka sistem akan mengirim pesan *error*. Jika *key* valid, maka sistem akan menyimpan perubahan *key* pada *database*.

- f. Jika admin ingin menghapus *key*, maka admin dapat memilih *key* yang akan dihapus. Sistem akan menghapus *key* pada database.
- g. Jika admin ingin mengubah status aktif *key*, maka admin dapat memilih *key* yang akan diaktifkan.
- h. Sistem akan melakukan pengecekan pada *key*.
- i. Jika *key* tidak memenuhi ketentuan untuk terhubung ke *api*, maka sistem akan mengirim pesan *error*.
- j. Jika *key* memenuhi ketentuan untuk terhubung ke *api*, maka sistem akan mengubah data *key* menjadi aktif pada database.



Gambar 4. 16 Activity Diagram Menambah, Mengubah, dan Menghapus API Key



Gambar 4. 17 Activity Diagram Mengelola Aktivasi API Key

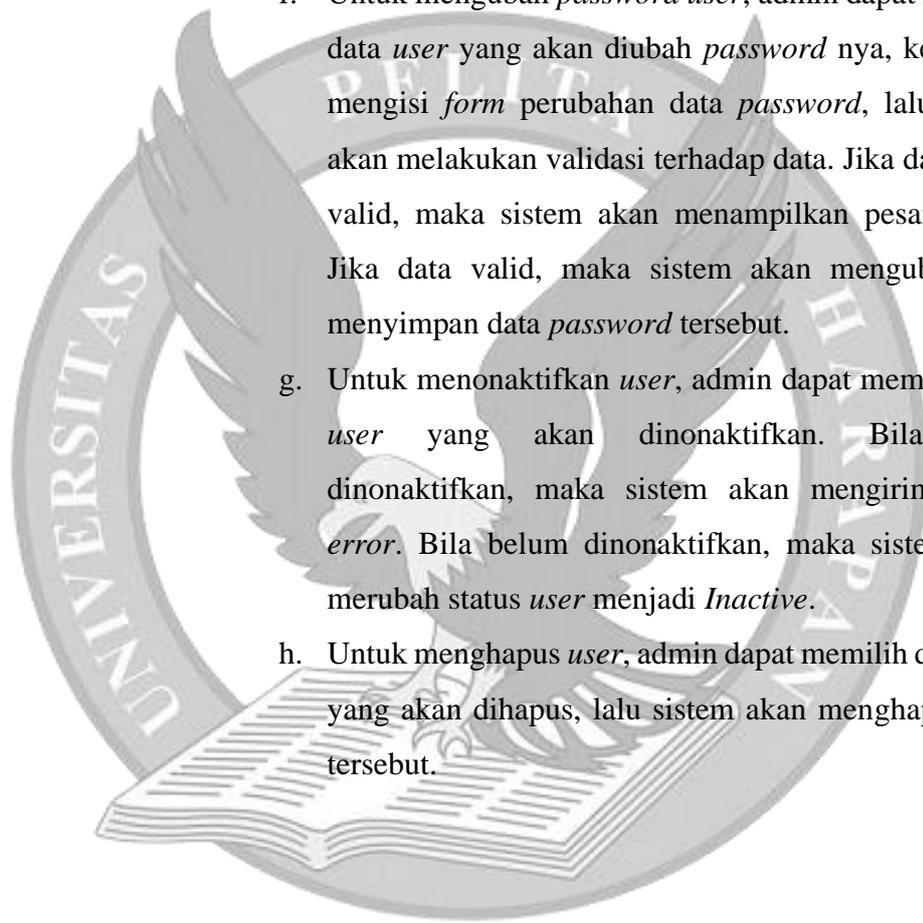
14) Mengelola *user*

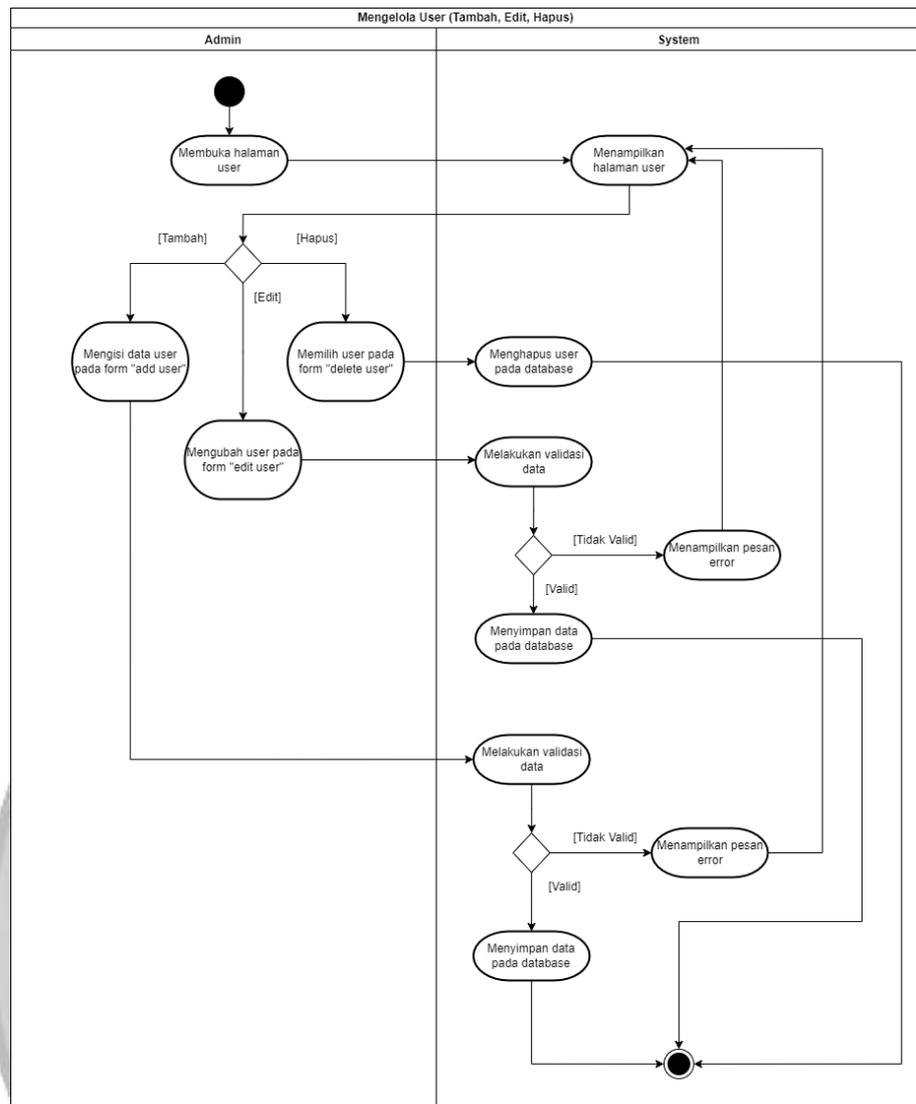
Gambar 4.18 dan Gambar 4.19 menunjukkan *activity diagram* mengelola *user*, adapun proses pengelolaan *user* terdiri dari langkah-langkah berikut:

- a. Admin membuka halaman *user*.
- b. Sistem menampilkan halaman *user*.
- c. Terdapat 5 hal yang dapat dilakukan admin pada fitur, yaitu menambah *user*, mengubah *user*, mengubah *password user*, menonaktifkan *user*, dan menghapus *user*.
- d. Untuk menambah *user*, admin dapat mengisi *form* penambahan data *user*, lalu sistem akan melakukan validasi terhadap data. Jika data tidak valid, maka

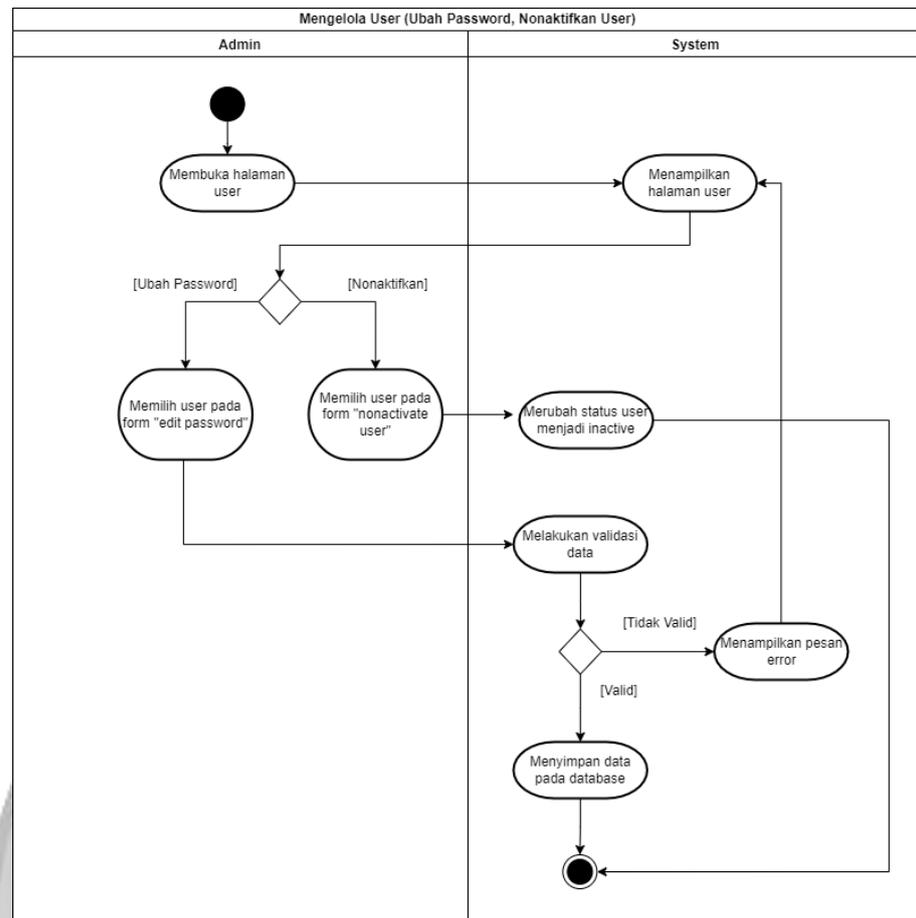
sistem akan menampilkan pesan *error*. Jika data valid, maka sistem akan menyimpan data tersebut.

- e. Untuk mengubah *user*, admin dapat memilih data *user* yang akan diubah, kemudian mengisi *form* perubahan data *user*, lalu sistem akan melakukan validasi terhadap data. Jika data tidak valid, maka sistem akan menampilkan pesan *error*. Jika data valid, maka sistem akan mengubah dan menyimpan data tersebut.
- f. Untuk mengubah *password user*, admin dapat memilih data *user* yang akan diubah *password* nya, kemudian mengisi *form* perubahan data *password*, lalu sistem akan melakukan validasi terhadap data. Jika data tidak valid, maka sistem akan menampilkan pesan *error*. Jika data valid, maka sistem akan mengubah dan menyimpan data *password* tersebut.
- g. Untuk menonaktifkan *user*, admin dapat memilih data *user* yang akan dinonaktifkan. Bila telah dinonaktifkan, maka sistem akan mengirim pesan *error*. Bila belum dinonaktifkan, maka sistem akan merubah status *user* menjadi *Inactive*.
- h. Untuk menghapus *user*, admin dapat memilih data *user* yang akan dihapus, lalu sistem akan menghapus data tersebut.





Gambar 4. 18 Activity Diagram Tambah, Edit, Hapus User



Gambar 4. 19 Activity Diagram Ubah Password dan Nonaktifkan User

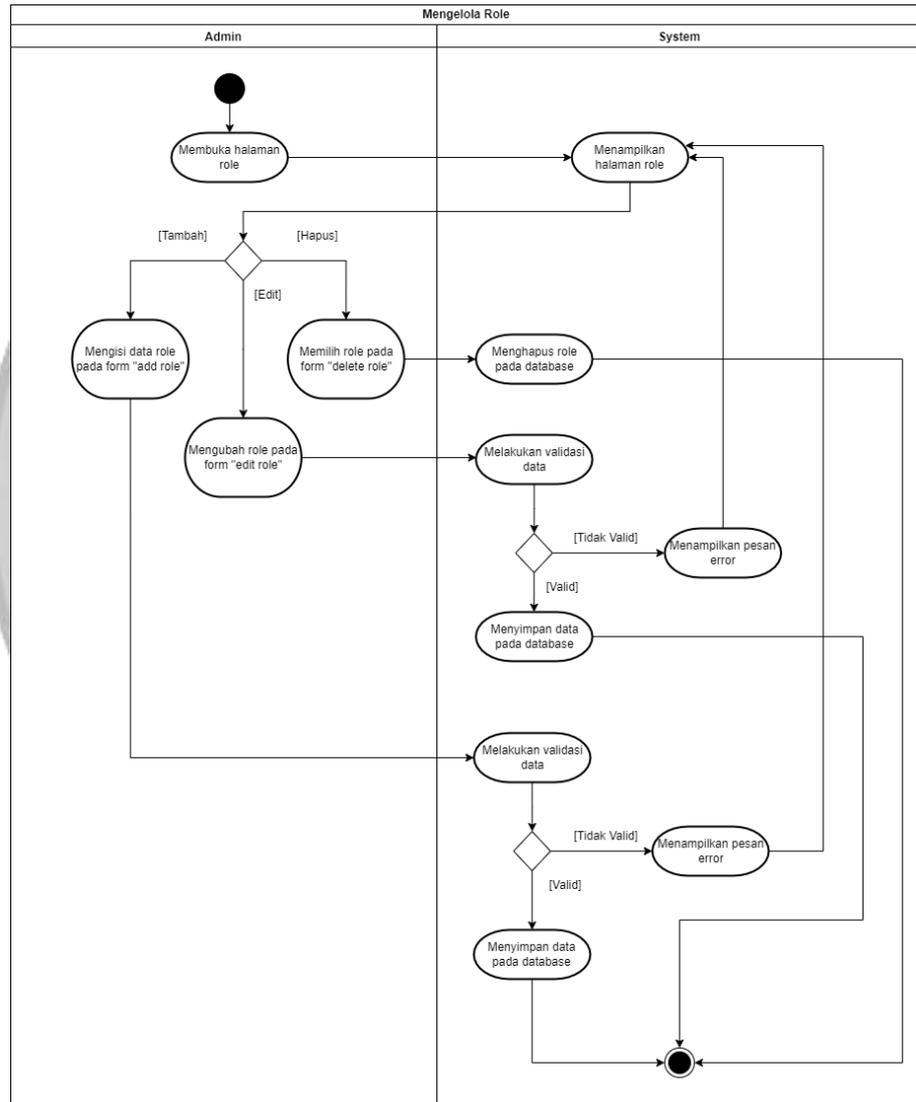
15) Mengelola role

Gambar 4.20 menunjukkan *activity diagram* mengelola *role*, adapun proses pengelolaan *role* terdiri dari langkah-langkah berikut:

- a. Admin membuka halaman *role*.
- b. Sistem menampilkan halaman *role*.
- c. Terdapat 3 hal yang dapat dilakukan admin pada fitur, yaitu menambah *role*, mengubah *role*, menghapus *role*.
- d. Untuk menambah *role*, admin dapat mengisi *form* penambahan data *role*, lalu sistem akan melakukan validasi terhadap data. Jika data tidak valid, maka sistem akan menampilkan pesan *error*. Jika data valid, maka sistem akan menyimpan data tersebut.
- e. Untuk mengubah *role*, admin dapat memilih data *role* yang akan diubah, kemudian mengisi *form* perubahan

data *role*, lalu sistem akan melakukan validasi terhadap data. Jika data tidak valid, maka sistem akan menampilkan pesan *error*. Jika data valid, maka sistem akan menyimpan data tersebut.

- f. Untuk menghapus *role*, admin dapat memilih data *role* yang akan dihapus, lalu sistem akan menghapus data tersebut.

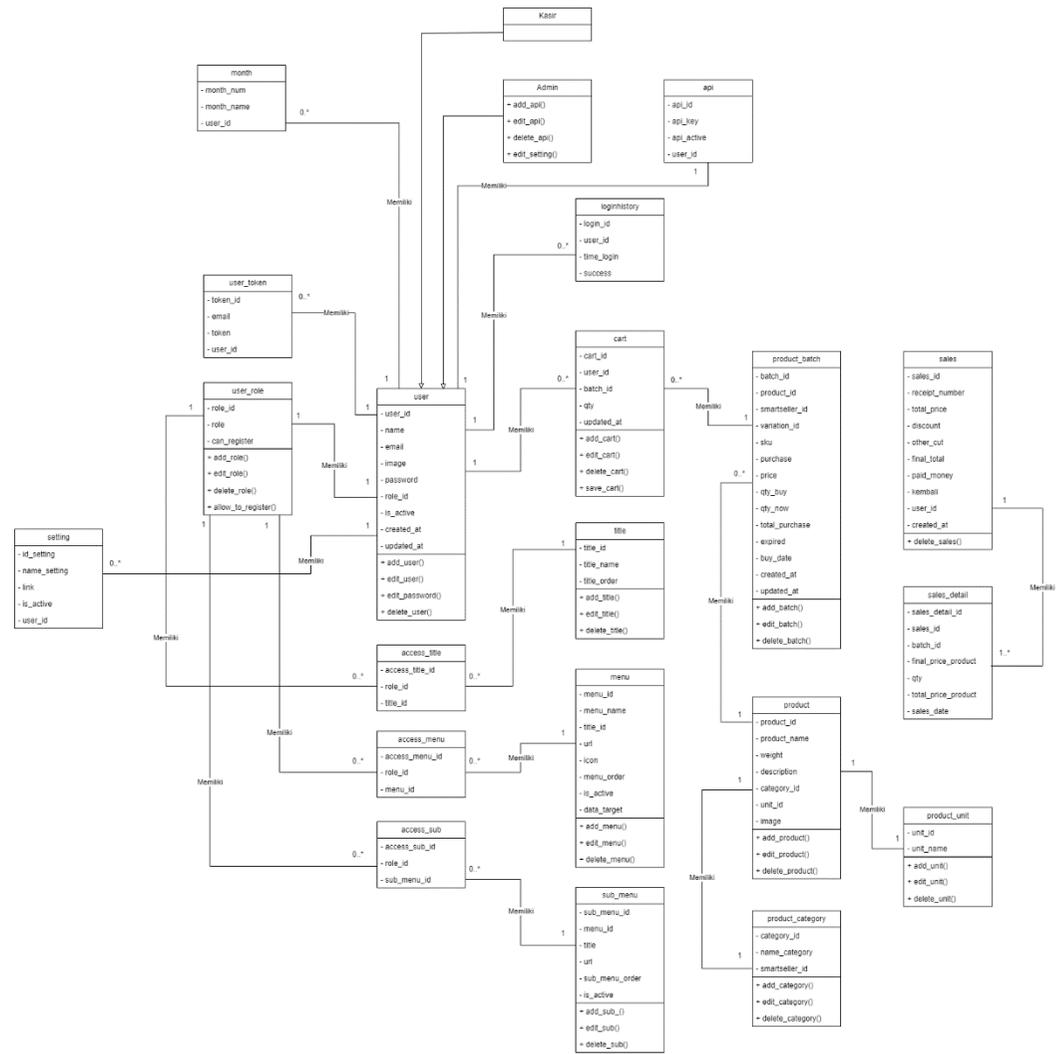


Gambar 4. 20 Activity Diagram Mengelola Role

4.3.2 Pemodelan Struktural

Pemodelan struktural dilakukan untuk memberi gambar terkait struktur objek yang mendukung proses bisnis dalam sebuah organisasi. Model ini mempresentasikan obyek, ide, atau konsep yang terdapat dalam permasalahannya. Pada sistem ini, digunakan *class diagram* dalam pemodelan struktur sistem, Gambar 4.21 berikut merupakan *class diagram* terhadap sistem.





Gambar 4. 21 Class Diagram

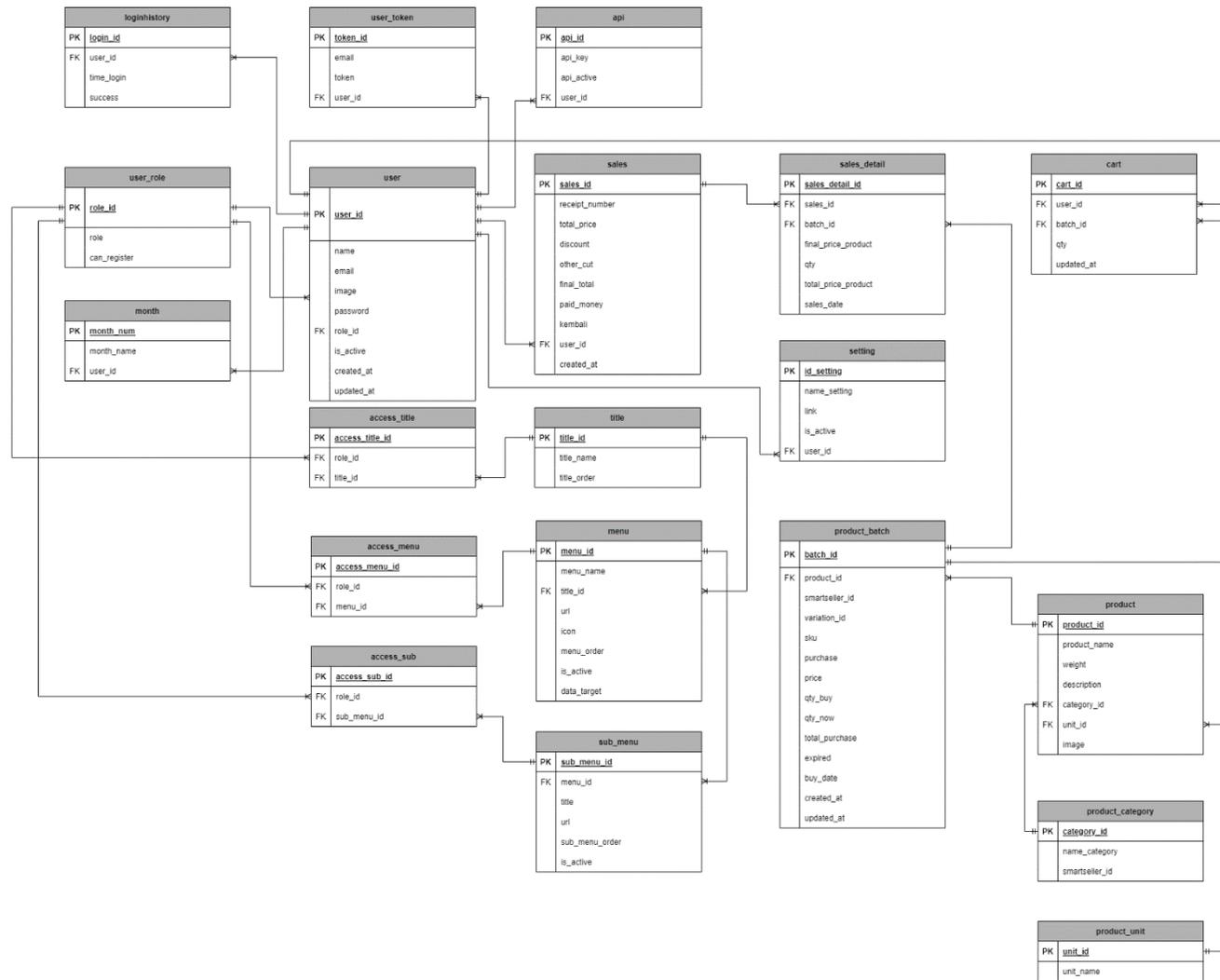
4.3.3 Perancangan Manajemen Data

Perancangan manajemen data untuk sistem *Web API Integrator* digambarkan dengan *table relationship diagram* dan *table description*.

4.3.3.1 Table Relationship Diagram

Gambar 4.22 berikut merupakan perancangan manajemen data dari sistem *Web API Integrator* yang digambarkan dengan *table relationship diagram*.





Gambar 4. 22 Table Relationship Diagram Sistem Web API Integrator

4.3.3.2 Table Description

Berikut ini merupakan deskripsi dari tabel-tabel yang digunakan pada *database* sistem *Web API Integrator*.

1. Tabel *user*

Tabel 4. 3 *Table Description* pada tabel *User*

<i>Field name</i>	<i>Type</i>	<i>Length</i>	<i>Key</i>
<i>user_id</i>	int	-	<i>Primary key</i>
<i>name</i>	varchar	256	
<i>email</i>	varchar	256	
<i>image</i>	varchar	256	
<i>password</i>	varchar	256	
<i>role_id</i>	int	-	<i>Foreign key</i>
<i>is_active</i>	int	-	
<i>created_at</i>	datetime	-	
<i>updated_at</i>	datetime	-	

2. Tabel *user_role*

Tabel 4. 4 *Table Description* pada tabel *User_Role*

<i>Field name</i>	<i>Type</i>	<i>Length</i>	<i>Key</i>
<i>role_id</i>	int	-	<i>Primary key</i>
<i>role</i>	varchar	128	
<i>can_register</i>	int		

3. Tabel *access_menu*

Tabel 4. 5 *Table Description* pada tabel *Access_Menu*

<i>Field name</i>	<i>Type</i>	<i>Length</i>	<i>Key</i>
<i>access_menu_id</i>	int	-	<i>Primary key</i>
<i>role_id</i>	int	-	<i>Foreign key</i>
<i>menu_id</i>	int	-	<i>Foreign key</i>

4. Tabel *access_sub*

Tabel 4. 6 *Table Description* pada tabel *Access_Sub*

<i>Field name</i>	<i>Type</i>	<i>Length</i>	<i>Key</i>
<i>access_sub_id</i>	int	-	<i>Primary key</i>
<i>role_id</i>	int	-	<i>Foreign key</i>
<i>sub_menu_id</i>	int	-	<i>Foreign key</i>

5. Tabel *menu*

Tabel 4. 7 *Table Description* pada tabel *Menu*

<i>Field name</i>	<i>Type</i>	<i>Length</i>	<i>Key</i>
<i>menu_id</i>	int	-	<i>Primary key</i>
<i>menu_name</i>	varchar	128	
<i>title_id</i>	int	-	<i>Foreign key</i>
<i>url</i>	varchar	128	
<i>icon</i>	varchar	128	

Tabel 4. 7 *Table Description* pada tabel Menu (Lanjutan)

<i>menu_order</i>	int	-	
<i>is_active</i>	int	-	
<i>data_target</i>	varchar	128	

6. Tabel *sub_menu*

Tabel 4. 8 *Table Description* pada tabel *Sub_Menu*

<i>Field name</i>	<i>Type</i>	<i>Length</i>	<i>Key</i>
<i>sub_menu_id</i>	int	-	<i>Primary key</i>
<i>menu_id</i>	int	-	<i>Foreign key</i>
<i>title</i>	varchar	128	
<i>url</i>	varchar	128	
<i>sub_menu_order</i>	int	-	
<i>is_active</i>	int	-	

7. Tabel *cart*

Tabel 4. 9 *Table Description* pada tabel *Cart*

<i>Field name</i>	<i>Type</i>	<i>Length</i>	<i>Key</i>
<i>cart_id</i>	int	-	<i>Primary key</i>
<i>user_id</i>	int	-	<i>Foreign key</i>
<i>batch_id</i>	int	-	<i>Foreign key</i>

Tabel 4. 9 *Table Description* pada tabel *Cart* (Lanjutan)

<i>qty</i>	int	-	
<i>created_at</i>	datetime	-	

8. Tabel *sales*

Tabel 4. 10 *Table Description* pada tabel *Sales*

<i>Field name</i>	<i>Type</i>	<i>Length</i>	<i>Key</i>
<i>sales_id</i>	int	-	<i>Primary key</i>
<i>receipt_number</i>	varchar	128	
<i>total_price</i>	int	-	
<i>discount</i>	int	-	
<i>other_cut</i>	int	-	
<i>final_total</i>	int	-	
<i>paid_money</i>	int	-	
<i>kembali</i>	int	-	
<i>user_id</i>	int	-	<i>Foreign Key</i>
<i>created_at</i>	datetime		

9. Tabel *sales_detail*

Tabel 4. 11 *Table Description* pada tabel *Sales_Detail*

<i>Field name</i>	<i>Type</i>	<i>Length</i>	<i>Key</i>
<i>sales_detail_id</i>	int	-	<i>Primary key</i>

Tabel 4. 11 *Table Description* pada tabel *Sales_Detail* (Lanjutan)

<i>sales_id</i>	int	-	<i>Foreign key</i>
<i>batch_id</i>	int	-	<i>Foreign key</i>
<i>final_price_product</i>	int	-	
<i>qty</i>	int	-	
<i>total_price_product</i>	int	-	
<i>sales_date</i>	date	-	

10. Tabel *product*

Tabel 4. 12 *Table Description* pada tabel *Product*

<i>Field name</i>	<i>Type</i>	<i>Length</i>	<i>Key</i>
<i>product_id</i>	int	-	<i>Primary key</i>
<i>product_name</i>	varchar	128	
<i>weight</i>	int	-	
<i>description</i>	text	-	
<i>category_id</i>	int	-	<i>Foreign key</i>
<i>unit_id</i>	int	-	<i>Foreign key</i>
<i>image</i>	varchar	256	

11. Tabel *product_category*

Tabel 4. 13 *Table Description* pada tabel *Product_Category*

<i>Field name</i>	<i>Type</i>	<i>Length</i>	<i>Key</i>
<i>category_id</i>	int	-	<i>Primary key</i>
<i>name_category</i>	int	-	
<i>smartseller_id</i>	int	-	

12. Tabel *product_batch*

Tabel 4. 14 *Table Description* pada tabel *Product_Batch*

<i>Field name</i>	<i>Type</i>	<i>Length</i>	<i>Key</i>
<i>batch_id</i>	int	-	<i>Primary key</i>
<i>product_id</i>	int	-	<i>Foreign key</i>
<i>smartseller_id</i>	int	-	
<i>variation_id</i>	int	-	
<i>sku</i>	varchar	128	
<i>purchase</i>	int	-	
<i>price</i>	int	-	
<i>qty_buy</i>	int	-	
<i>qty_now</i>	int	-	
<i>total_purchase</i>	int	-	
<i>expired</i>	date	-	

Tabel 4. 14 *Table Description* pada tabel *Product_Batch* (Lanjutan)

<i>buy_date</i>	date	-	
<i>updated_at</i>	datetime	-	

13. Tabel *product_unit*

Tabel 4. 15 *Table Description* pada tabel *Product_Unit*

<i>Field name</i>	<i>Type</i>	<i>Length</i>	<i>Key</i>
<i>unit_id</i>	int	-	<i>Primary key</i>
<i>unit_name</i>	varchar	128	

14. Tabel *month*

Tabel 4. 16 *Table Description* pada tabel *Month*

<i>Field name</i>	<i>Type</i>	<i>Length</i>	<i>Key</i>
<i>month_num</i>	int	-	<i>Primary key</i>
<i>month_name</i>	varchar	20	
<i>user_id</i>	int	-	<i>Foreign key</i>

15. Tabel *setting*

Tabel 4. 17 *Table Description* pada tabel *Setting*

<i>Field name</i>	<i>Type</i>	<i>Length</i>	<i>Key</i>
<i>id_setting</i>	int	-	<i>Primary key</i>
<i>name_setting</i>	varchar	128	

Tabel 4. 17 *Table Description* pada tabel *Setting* (Lanjutan)

<i>link</i>	varchar	256	
<i>is_active</i>	int	-	
<i>user_id</i>	int	-	<i>Foreign key</i>

16. Tabel *login_history*

Tabel 4. 18 *Table Description* pada tabel *Login_History*

<i>Field name</i>	<i>Type</i>	<i>Length</i>	<i>Key</i>
<i>login_id</i>	int	-	<i>Primary key</i>
<i>user_id</i>	int	-	<i>Foreign key</i>
<i>time_login</i>	datetime	-	
<i>success</i>	int	-	

17. Tabel *user_token*

Tabel 4. 19 *Table Description* pada tabel *User-Token*

<i>Field name</i>	<i>Type</i>	<i>Length</i>	<i>Key</i>
<i>token_id</i>	int	-	<i>Primary key</i>
<i>email</i>	varchar	256	
<i>token</i>	varchar	256	
<i>user_id</i>	int	-	<i>Foreign key</i>

18. Tabel *title*

Tabel 4. 20 *Table Description* pada tabel *Title*

<i>Field name</i>	<i>Type</i>	<i>Length</i>	<i>Key</i>
<i>title_id</i>	int	-	<i>Primary key</i>
<i>title_name</i>	varchar	128	
<i>title_order</i>	int	-	

19. Tabel *access_title*

Tabel 4. 21 *Table Description* pada tabel *Access_Title*

<i>Field name</i>	<i>Type</i>	<i>Length</i>	<i>Key</i>
<i>access_title_id</i>	int	-	<i>Primary key</i>
<i>role_id</i>	int	-	<i>Foreign key</i>
<i>title_id</i>	int	-	<i>Foreign key</i>

20. Tabel *api*

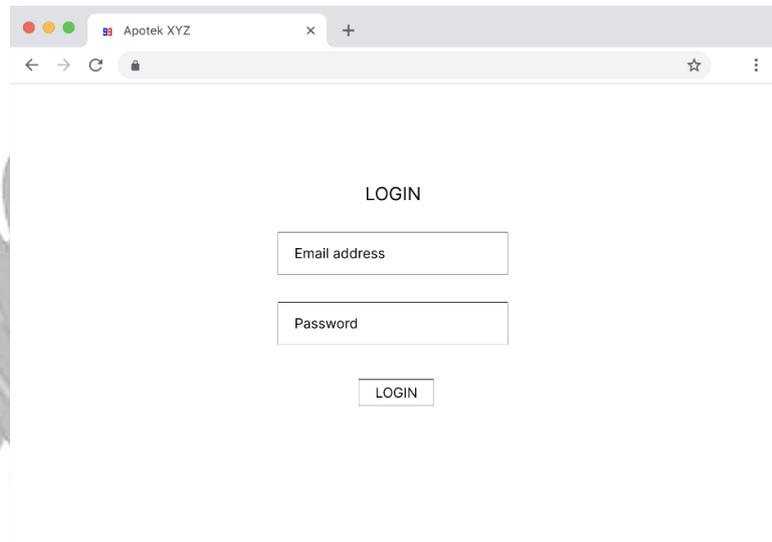
Tabel 4. 22 *Table Description* pada tabel *Api*

<i>Field name</i>	<i>Type</i>	<i>Length</i>	<i>Key</i>
<i>api_id</i>	int	-	<i>Primary key</i>
<i>api_key</i>	text	-	
<i>api_active</i>	int	-	
<i>user_id</i>	int	-	<i>Foreign key</i>

4.3.4 Mockup Lapisan Antarmuka

1) Login

Gambar 4.23 merupakan rancangan tampilan halaman *mockup* login di sistem *Web API Integrator* Apotek 99.



Apotek XYZ

LOGIN

Email address

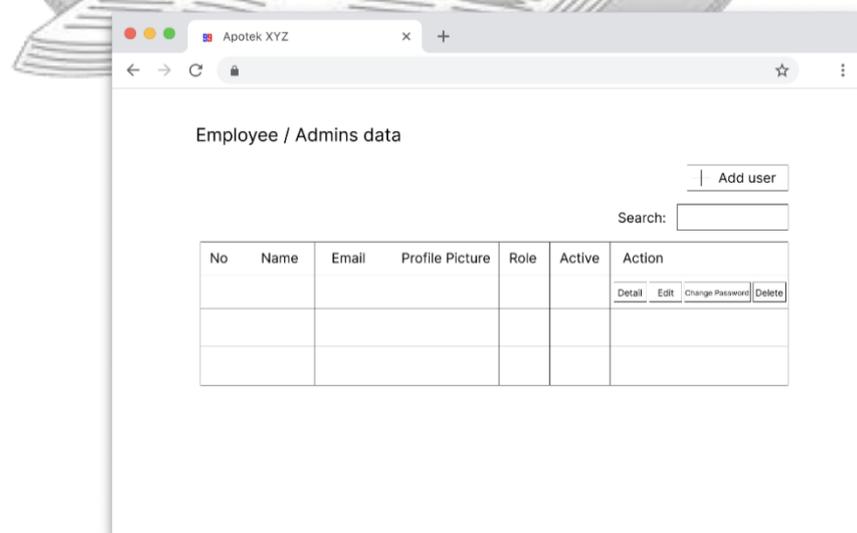
Password

LOGIN

Gambar 4. 23 Rancangan *Mockup* Login

2) Mengelola *user*

Gambar 4.24 merupakan rancangan tampilan halaman *mockup* pengelolaan *user* di sistem *Web API Integrator* Apotek 99.



Apotek XYZ

Employee / Admins data

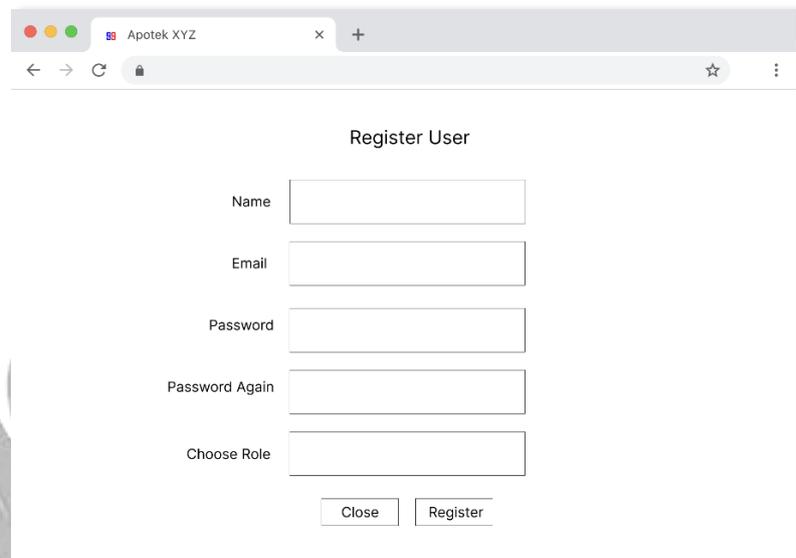
Add user

Search:

No	Name	Email	Profile Picture	Role	Active	Action
						Detail Edit Change Password Delete

Gambar 4. 24 Rancangan Lapisan Antarmuka Pengelolaan *User*

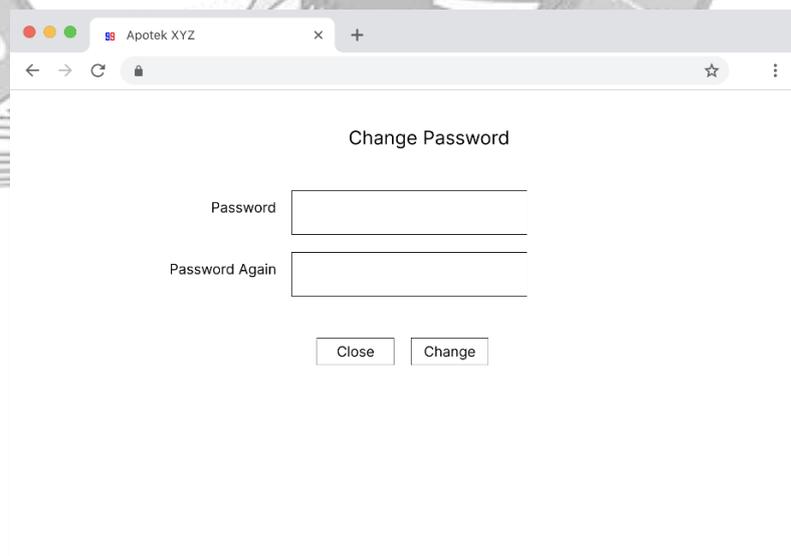
Gambar 4.25 merupakan rancangan tampilan halaman *mockup* registrasi pada menu *user* di sistem *Web API Integrator Apotek 99*.



The image shows a web browser window with the title 'Apotek XYZ'. The main content is a form titled 'Register User'. The form contains five input fields: 'Name', 'Email', 'Password', 'Password Again', and 'Choose Role'. Below the input fields are two buttons: 'Close' and 'Register'.

Gambar 4. 25 Rancangan Lapisan Antarmuka *Register User*

Gambar 4.26 merupakan rancangan tampilan halaman *mockup* ubah *password* pada menu *user* di sistem *Web API Integrator Apotek 99*.

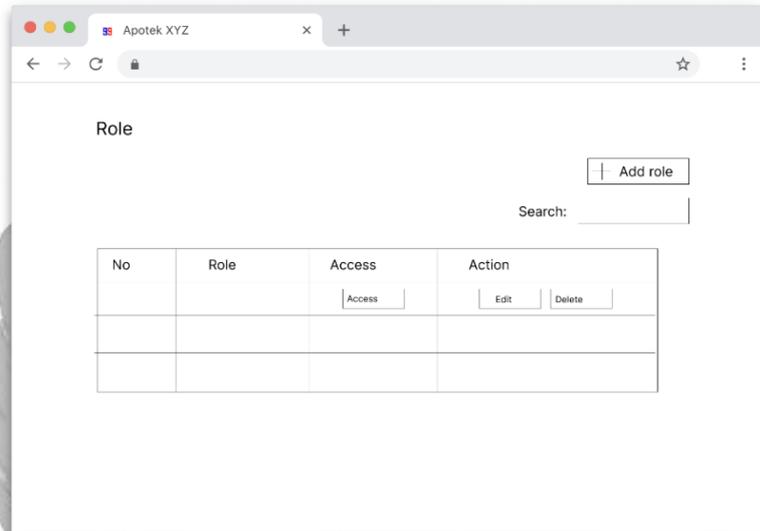


The image shows a web browser window with the title 'Apotek XYZ'. The main content is a form titled 'Change Password'. The form contains two input fields: 'Password' and 'Password Again'. Below the input fields are two buttons: 'Close' and 'Change'.

Gambar 4. 26 Rancangan Lapisan Antarmuka *Ubah Password*

3) Mengelola *role*

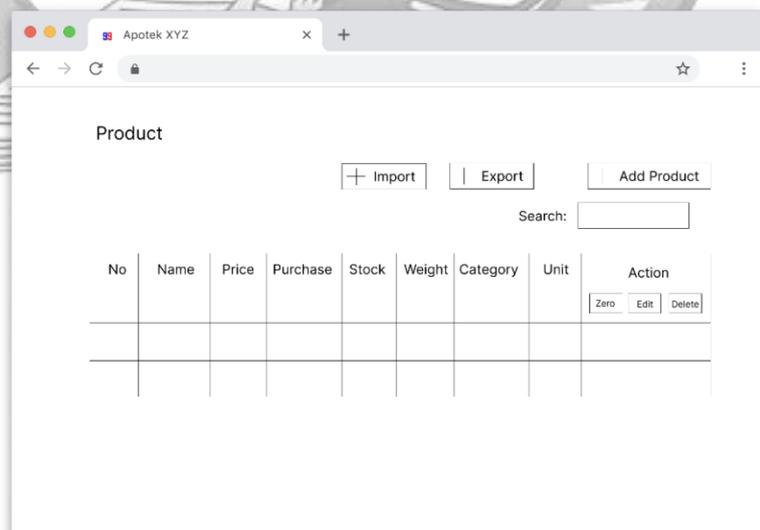
Gambar 4.27 merupakan rancangan tampilan halaman *mockup* pengelolaan *role* di sistem *Web API Integrator Apotek 99*.



Gambar 4. 27 Gambar Lapisan Antarmuka Mengelola *Role*

4) Mengelola data produk

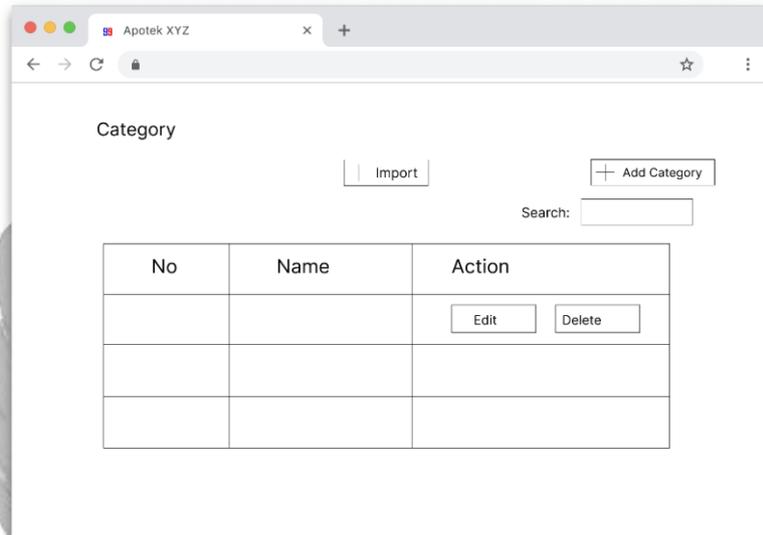
Gambar 4.28 merupakan rancangan tampilan halaman *mockup* pengelolaan data produk di sistem *Web API Integrator Apotek 99*.



Gambar 4. 28 Gambar Lapisan Antarmuka Mengelola Produk

5) Mengelola kategori produk

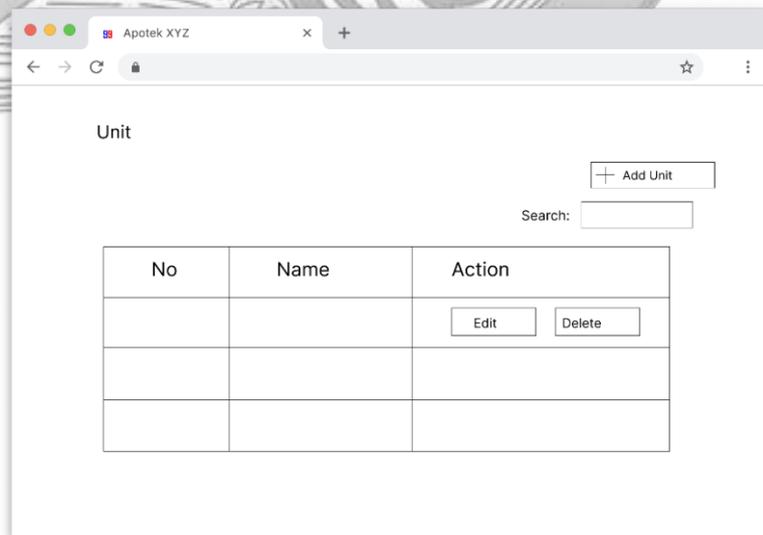
Gambar 4.29 merupakan rancangan tampilan halaman *mockup* pengelolaan data kategori produk di sistem *Web API Integrator* Apotek 99.



Gambar 4. 29 Gambar Lapisan Antarmuka Mengelola Kategori Produk

6) Mengelola unit produk

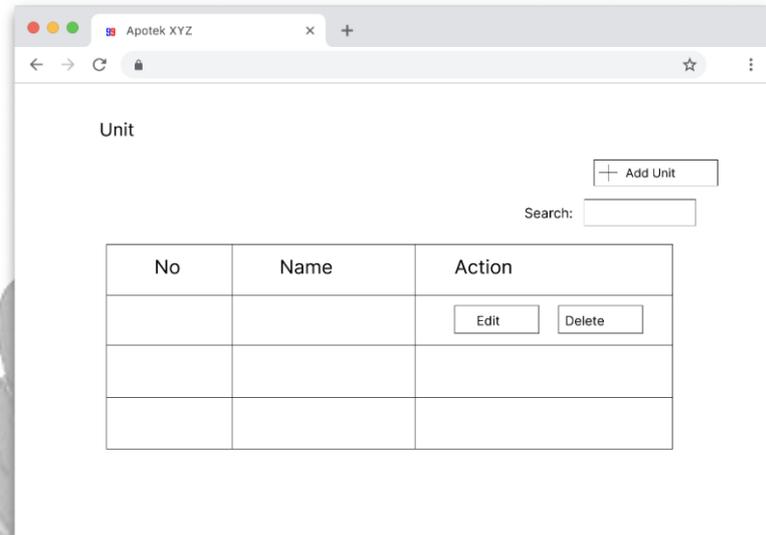
Gambar 4.30 merupakan rancangan tampilan halaman *mockup* pengelolaan data unit produk di sistem *Web API Integrator* Apotek 99.



Gambar 4. 30 Gambar Lapisan Antarmuka Mengelola Unit Produk

7) Mengelola data batch produk

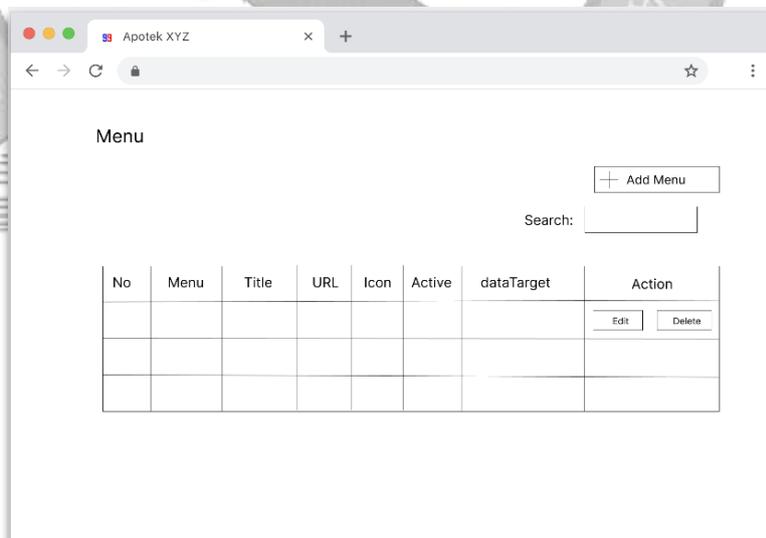
Gambar 4.31 merupakan rancangan tampilan halaman *mockup* pengelolaan data batch di sistem *Web API Integrator* Apotek 99.



Gambar 4. 31 Gambar Lapisan Antarmuka Mengelola *Batch* Produk

8) Mengelola *menu*

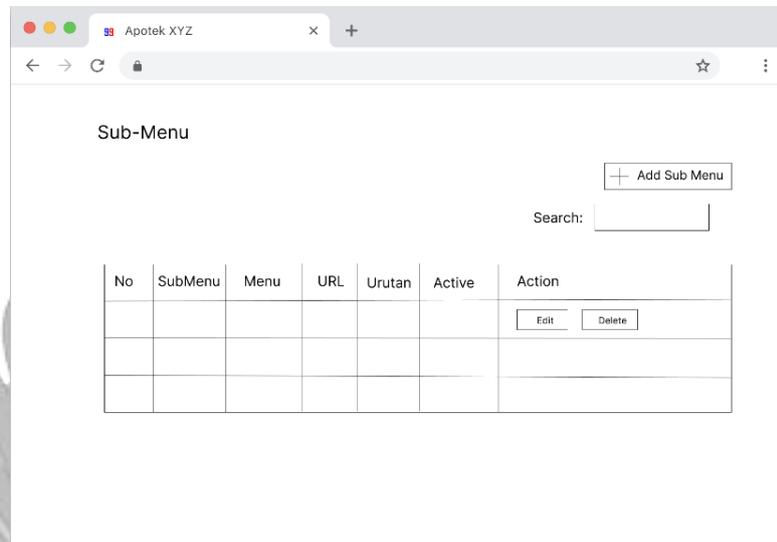
Gambar 4.32 merupakan rancangan tampilan halaman *mockup* pengelolaan data menu di sistem *Web API Integrator* Apotek 99.



Gambar 4. 32 Gambar Lapisan Antarmuka Mengelola Menu

9) Mengelola *submenu*

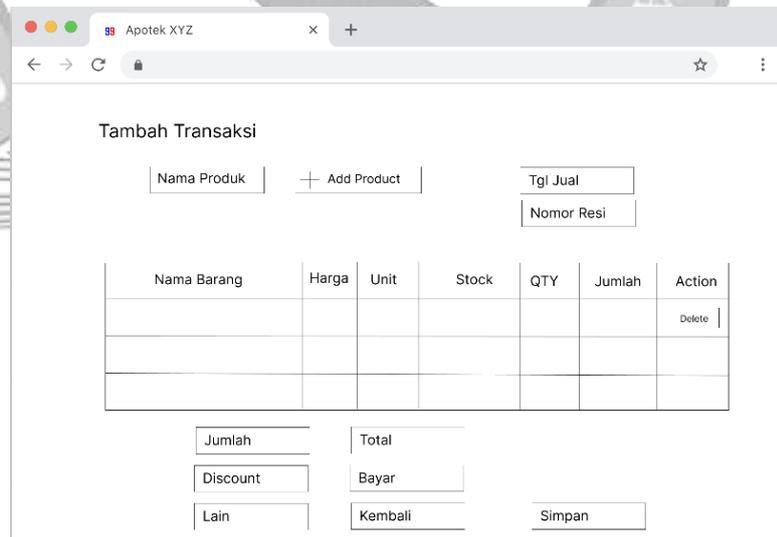
Gambar 4.33 merupakan rancangan tampilan halaman *mockup* pengelolaan data *submenu* di sistem *Web API Integrator Apotek 99*.



Gambar 4. 33 Gambar Lapisan Antarmuka Mengelola *Submenu*

10) Menambah transaksi toko

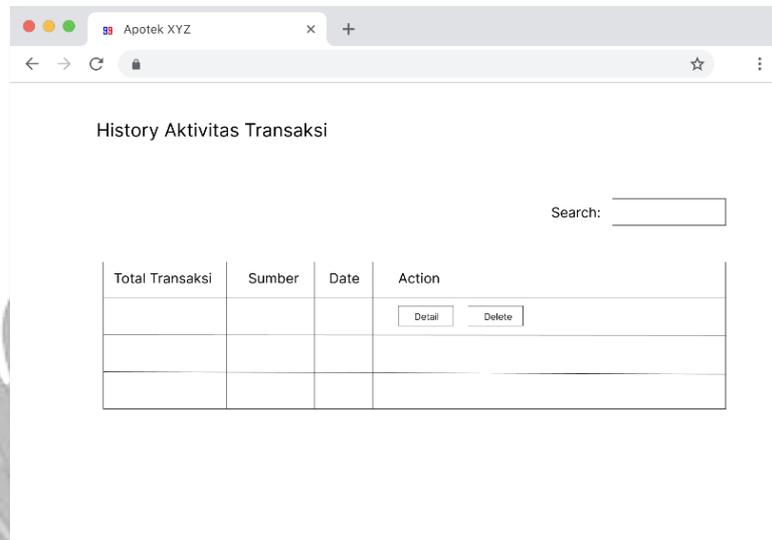
Gambar 4.34 merupakan rancangan tampilan halaman *mockup* transaksi toko di sistem *Web API Integrator Apotek 99*.



Gambar 4. 34 Gambar Lapisan Antarmuka Menambah Transaksi Toko

11) Melihat transaksi toko

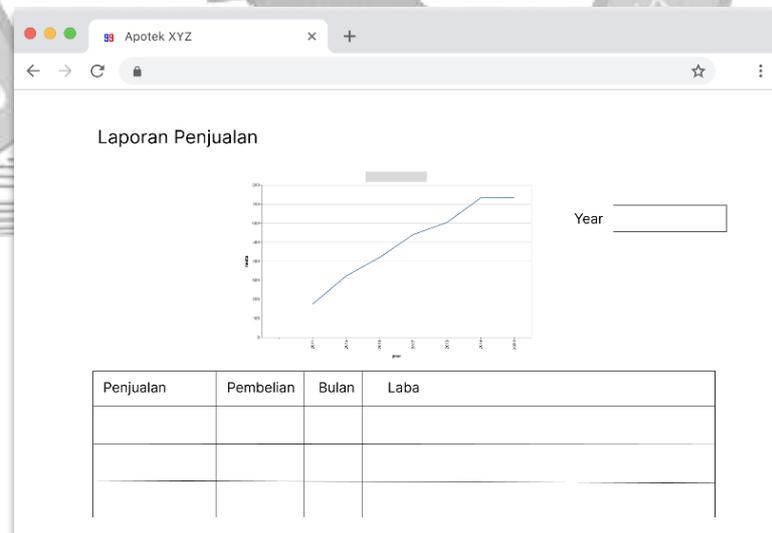
Gambar 4.35 merupakan rancangan tampilan halaman *mockup* aktivitas transaksi di sistem *Web API Integrator* Apotek 99.



Gambar 4. 35 Gambar Lapisan Antarmuka Aktivitas Transaksi

12) Laporan penjualan

Gambar 4.36 merupakan rancangan tampilan halaman *mockup* laporan penjualan di sistem *Web API Integrator* Apotek 99.



Gambar 4. 36 Gambar Lapisan Antarmuka Laporan Penjualan

4.4 Implementasi Sistem

Implementasi sistem usulan terdiri dari beberapa tahap, yaitu tahap pemrograman, tahap lapisan antarmuka, dan tahap pengujian.

4.4.1 Tahap Pemrograman

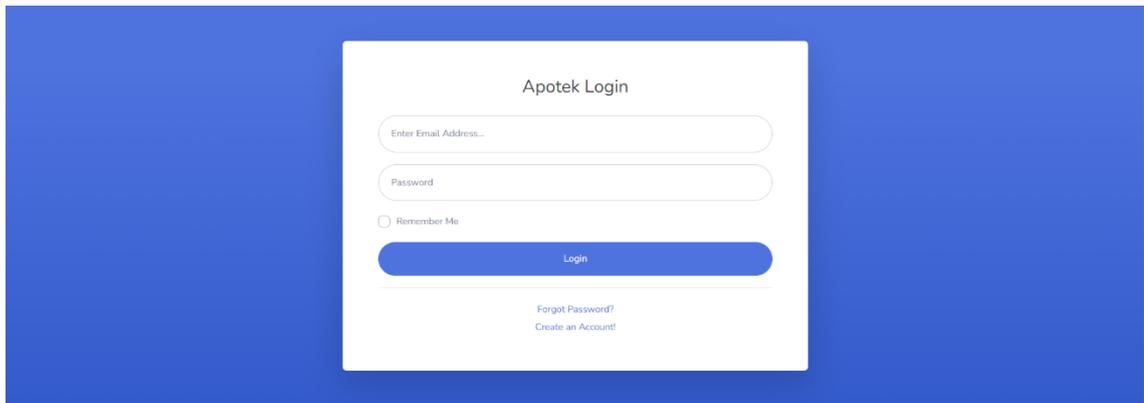
Sistem *Web API Integrator* yang diusulkan dikembangkan berbasis *web* dengan menggunakan bahasa pemrograman *PHP* dan *JavaScript*, menggunakan *framework CodeIgniter 3*. *IDE* yang digunakan dalam pembuatan program ini adalah *Visual Studio Code* dan menggunakan *PHPMysqlAdmin* sebagai *database*. Desain yang terkandung di dalam sistem ini seperti desain halaman *form*, dan tabel menggunakan *template* dari *SB Admin 2*, dan menggunakan *Bootstrap*. Adapun juga penggunaan *plugin JavaScript*, yaitu *SweetAlert2*, yang digunakan dalam sistem ini, dengan kegunaan untuk mempercantik *popup* atau *alert* yang terkandung dalam sistem ini, seperti *pop up* jika data telah berhasil atau gagal ditambahkan. Selain itu, ada *Select2*, yang digunakan untuk menampilkan *dropdown* yang bersumber dari *database*, dalam menentukan data yang akan diinput.

4.4.2 Perancangan Lapisan Antarmuka

Sistem *Web API Integrator* yang diusulkan untuk Apotek 99 memiliki dua aktor, yaitu admin sebagai pemimpin tertinggi, yang dapat mengakses semua fungsi pada sistem, oleh pemilik, dan kasir. Berikut merupakan *user interface* dari fungsi-fungsi yang terdapat dalam sistem usulan.

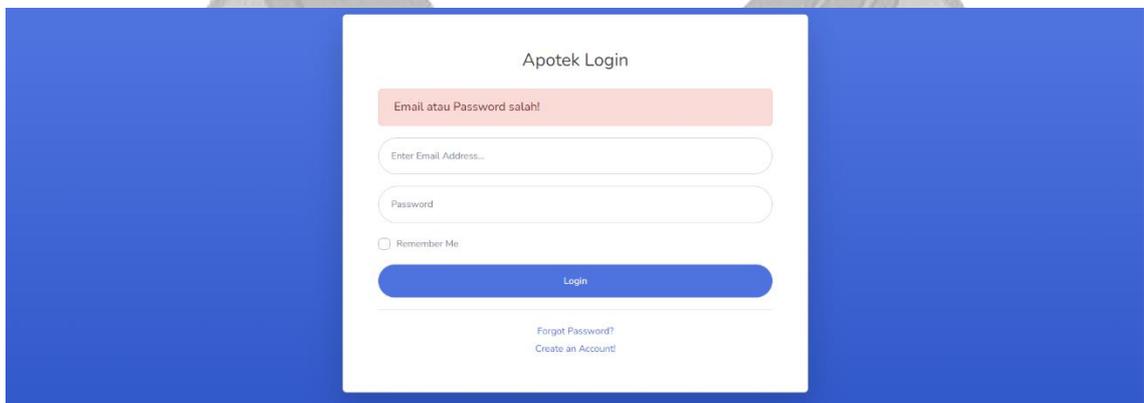
1) *Login*

Berikut adalah tampilan *login* yang dapat diakses oleh semua *user* baik *admin* atau kasir. Pada halaman ini, *user* diminta memasukkan email dan *password* yang telah terdaftar dan memiliki status aktif, agar dapat masuk ke dalam sistem. Terdapat pada Gambar 4.37.



Gambar 4. 37 Tampilan *Log In*

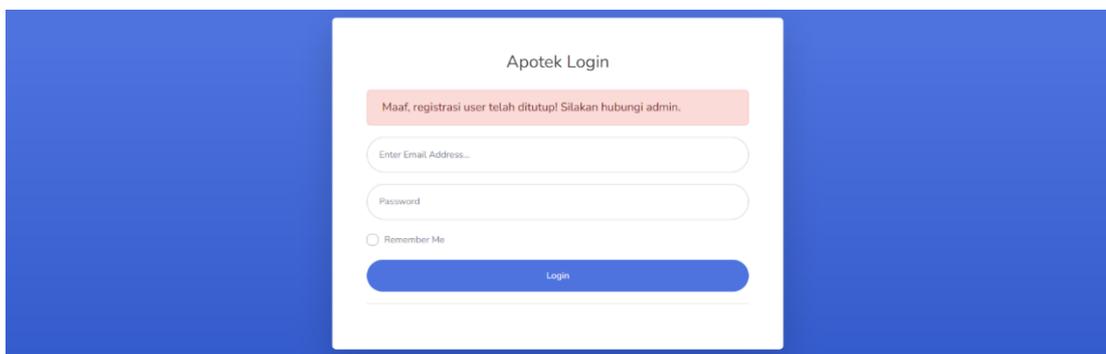
Jika *user* memasukkan email dan/atau *password* yang belum terdaftar, maka sistem akan memberi pesan error, yang terdapat pada Gambar 4.38.



Gambar 4. 38 Tampilan *Log In* jika *User* Salah Memasukkan Email dan/atau *Password*

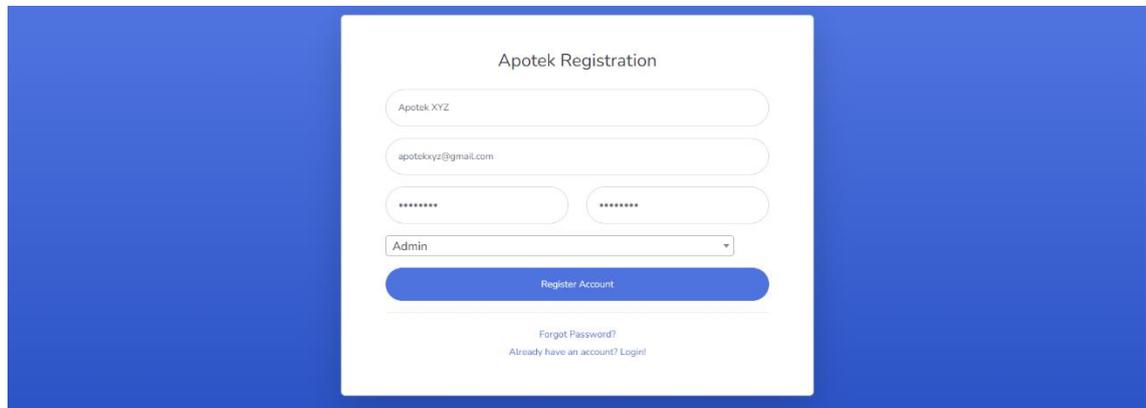
2) *Register*

Berikut adalah tampilan *register* yang dapat diakses oleh semua *user* baik *admin* atau kasir. Halaman ini diatur oleh *admin* pada bagian *setting*. Jika *admin* menonaktifkan fitur, maka sistem akan mengirim pesan *error* pada halaman *login*. Terdapat pada Gambar 4.39.



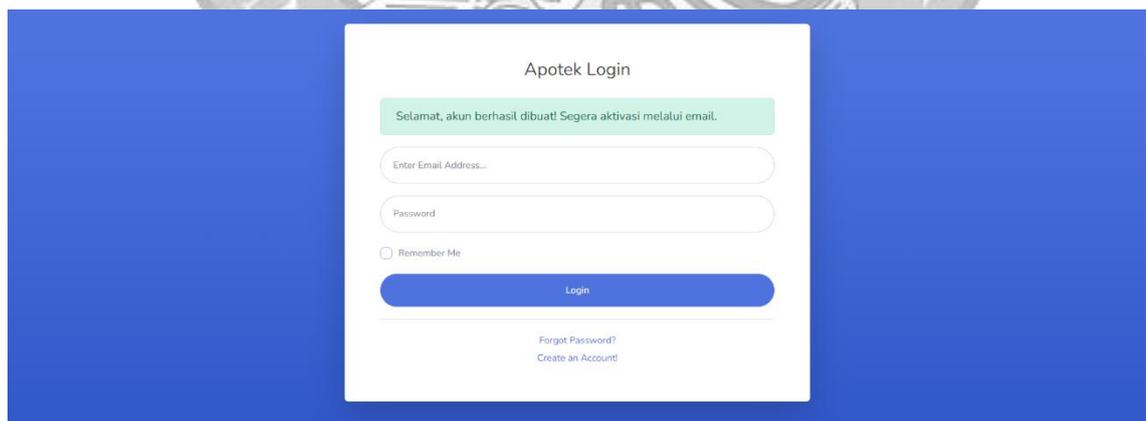
Gambar 4. 39 Tampilan Buka *Register* jika Halaman Dinonaktifkan Admin

Jika admin mengaktifkan fitur, maka sistem akan membuka halaman *register*. Pada halaman ini, *user* diminta memasukkan nama lengkap, email yang terdaftar dan memiliki status aktif di layanan domain, kombinasi *password* dan konfirmasi *password*, memilih *role*, dan menekan *Register*. Terdapat pada Gambar 4.40.

The image shows a registration form titled "Apotek Registration" on a white background with a blue border. The form contains the following elements: a text input field with "Apotek XYZ", an email input field with "apotekxyz@gmail.com", two password input fields with masked characters "*****", a dropdown menu with "Admin" selected, a blue "Register Account" button, a "Forgot Password?" link, and a "Already have an account? Login!" link.

Gambar 4. 40 Tampilan Register dengan Input Data Lengkap

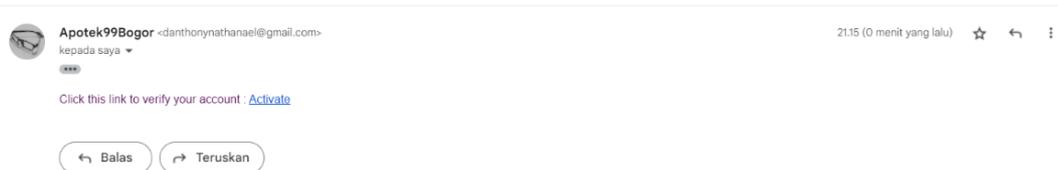
Sistem akan memeriksa validasi pada data yang diinput. Jika data tidak valid, termasuk email telah terdaftar, kombinasi *password* tidak sama, maka sistem akan menampilkan pesan *error*. Jika semua data dinyatakan valid, maka sistem akan membuat data token, dan mengirim link aktivasi akun, ke email yang diinput. Terdapat pada Gambar 4.41.

The image shows a login form titled "Apotek Login" on a white background with a blue border. At the top, there is a green success message: "Selamat, akun berhasil dibuat! Segera aktivasi melalui email." Below this are input fields for "Enter Email Address..." and "Password", a "Remember Me" checkbox, a blue "Login" button, and links for "Forgot Password?" and "Create an Account!".

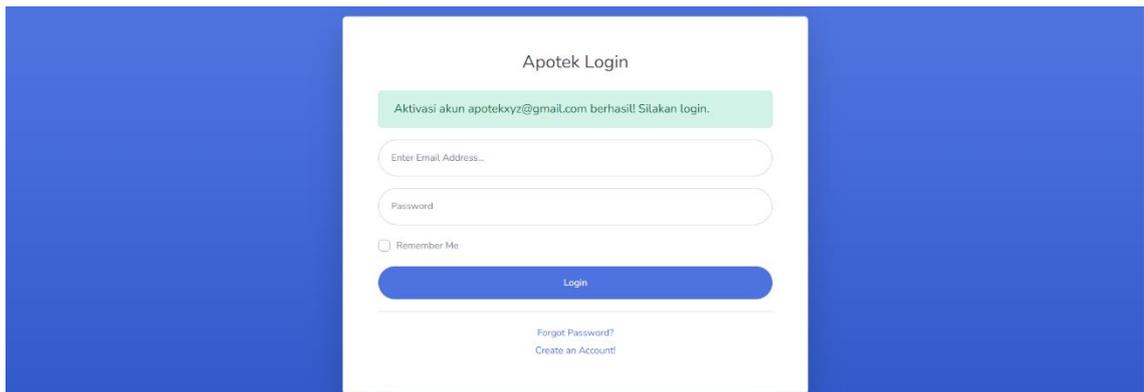
Gambar 4. 41 Tampilan Registrasi Berhasil

User membuka email, dan menekan link aktivasi, dan sistem akan memeriksa validasi pada link yang diinput. Jika link tidak valid, maka sistem akan mengirim pesan error. Jika link valid, maka sistem akan

mengaktifkan akun, dan menghasilkan pesan sukses. *User* sudah bisa melakukan *login* dan menggunakan sistem. Terdapat pada Gambar 4.42 dan Gambar 4.43.



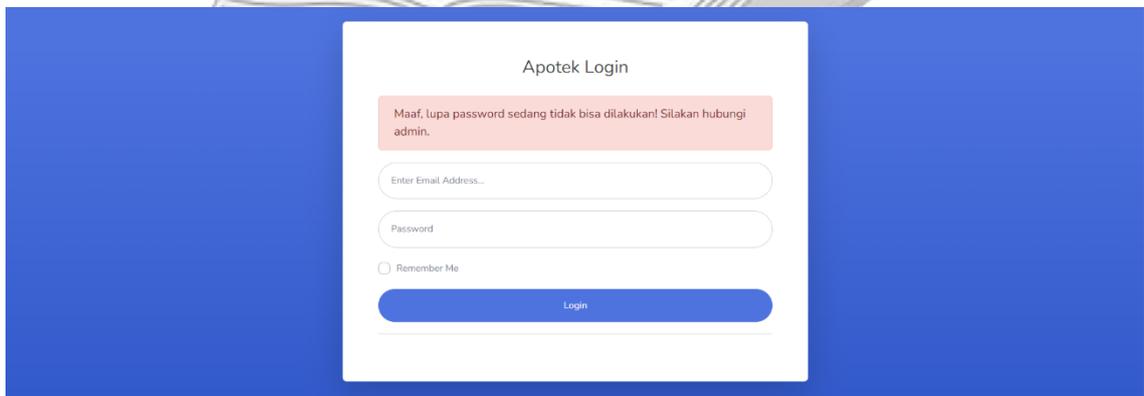
Gambar 4. 42 Email Masuk dengan Link Aktivasi



Gambar 4. 43 Pesan Akun Berhasil Diaktifkan

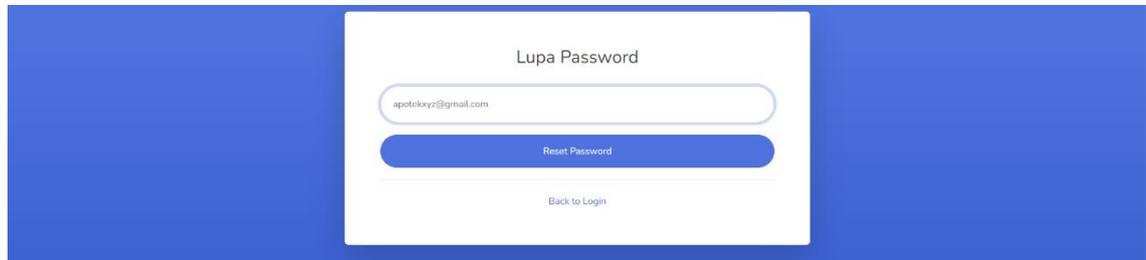
3) *Forgot Password*

Berikut adalah tampilan *forgot password* yang dapat diakses oleh semua *user* baik *admin* atau kasir. Halaman ini diatur oleh *admin* pada bagian *setting*. Jika *admin* menonaktifkan fitur, maka sistem akan mengirim pesan *error* pada halaman *login*. Terdapat pada Gambar 4.44.



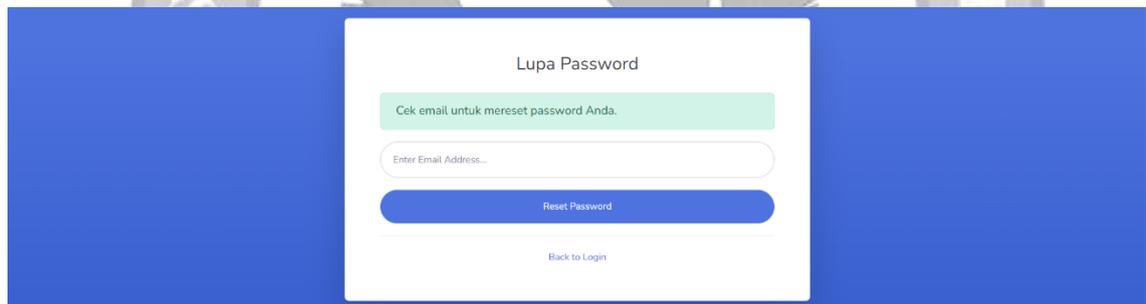
Gambar 4. 44 Tampilan Buka *Forgot Password* jika Dinonaktifkan Admin

Jika admin mengaktifkan fitur, maka sistem akan membuka halaman *forgot password*. Pada halaman ini, *user* diminta memasukkan email yang terdaftar pada sistem, lalu menekan tombol *Reset Password*. Terdapat pada Gambar 4.45.

The image shows a web form titled "Lupa Password" centered on a blue background. The form has a white background and contains a text input field with the email address "apotekeyz@gmail.com" entered. Below the input field is a blue button labeled "Reset Password". At the bottom of the form, there is a link labeled "Back to Login".

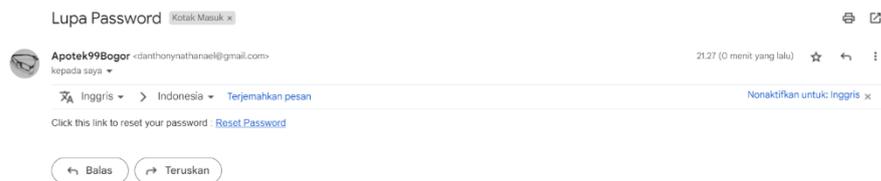
Gambar 4. 45 Tampilan *Forgot Password* dengan Data Email Aktif

Sistem akan memeriksa validasi pada data yang diinput. Jika data tidak valid, termasuk email tidak terdaftar atau belum teraktivasi, maka sistem akan menampilkan pesan *error*. Jika data dinyatakan valid, maka sistem akan membuat data token, dan mengirim link *reset password*, ke email yang diinput. Terdapat pada Gambar 4.46.

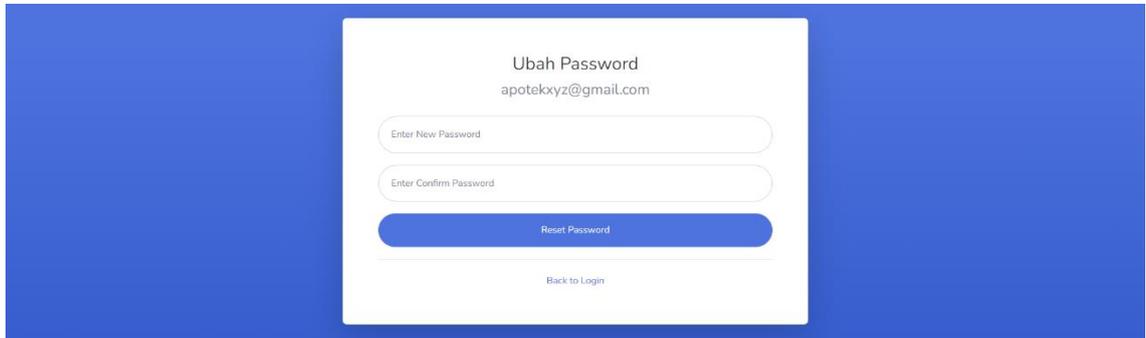
The image shows the same "Lupa Password" form as in the previous screenshot. However, a green message box at the top of the form displays the text "Cek email untuk mereset password Anda." Below this message, the input field now contains the placeholder text "Enter Email Address...". The "Reset Password" button and "Back to Login" link remain visible at the bottom.

Gambar 4. 46 Tampilan *Forgot Password* Berhasil

User menekan link reset, dan sistem akan memeriksa validasi pada link yang diinput. Jika link tidak valid, maka sistem akan mengirim pesan error. Jika link valid, maka sistem akan membuka tampilan *reset password*. *User* harus mengisi *form* dengan kombinasi *password* dan kombinasi *password*, lalu menekan tombol *Reset Password*. Terdapat pada Gambar 4.47 dan Gambar 4.48.

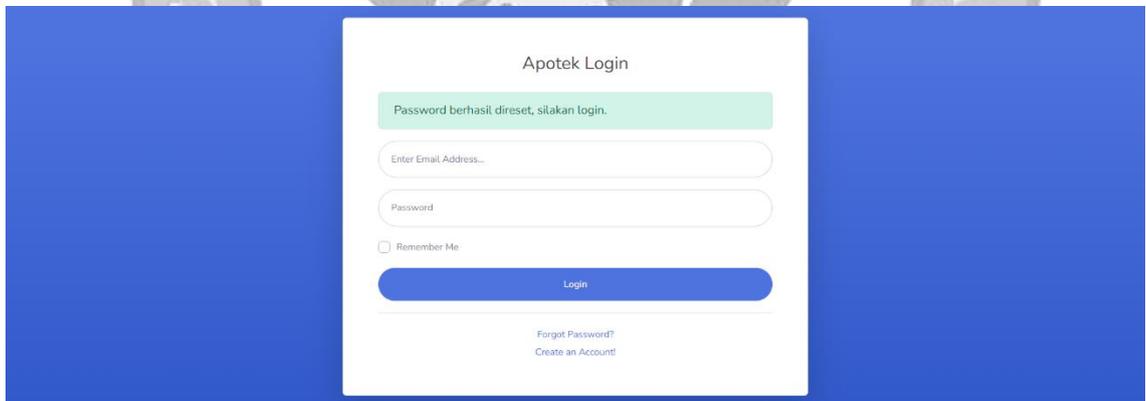


Gambar 4. 47 Email Masuk dengan Link *Reset Password*



Gambar 4. 48 Tampilan *Reset Password*

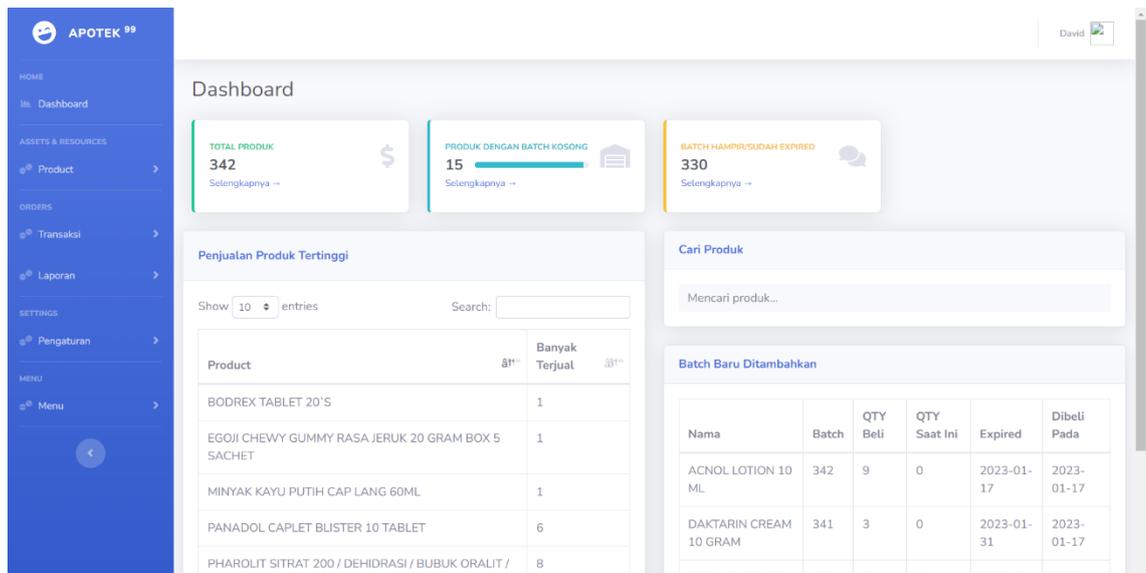
Sistem akan memeriksa validasi pada data yang diinput. Jika data tidak valid, maka sistem akan menampilkan pesan *error*. Jika data dinyatakan valid, maka sistem akan menyimpan data *password* terbaru, dan menampilkan pesan sukses. Terdapat pada Gambar 4.49.



Gambar 4. 49 Tampilan *Reset Password* Berhasil

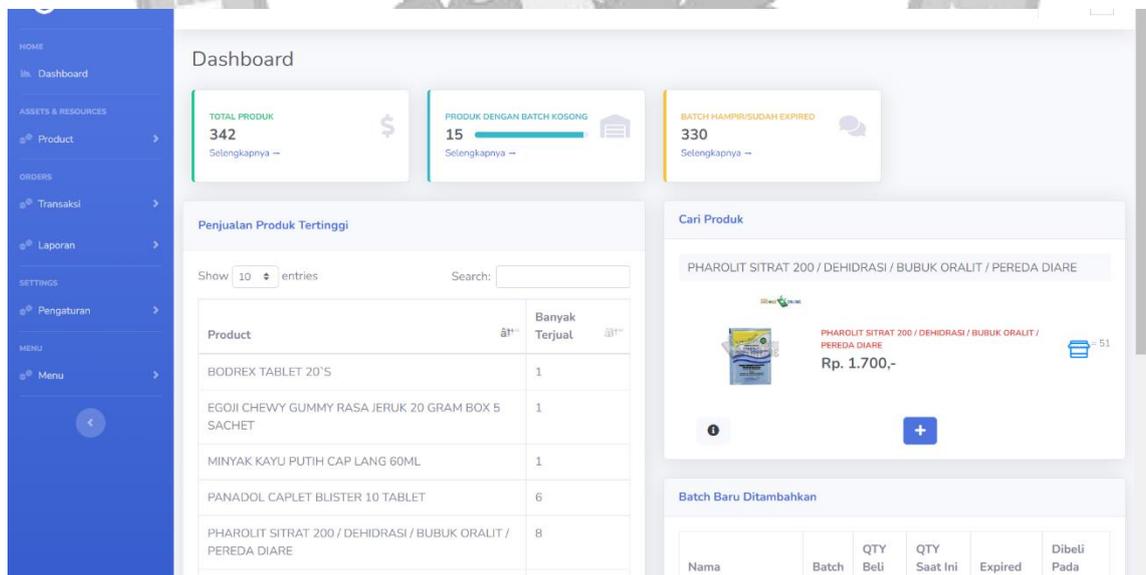
4) *Dashboard*

Berikut merupakan tampilan *dashboard* dari sistem *Web API Integrator* yang diusulkan pada menu ini. Admin dapat melihat berbagai macam informasi, seperti total produk, produk dengan *batch* kosong, *batch* produk hampir/sudah *expired*, cari produk, *batch* produk yang baru ditambahkan, dan penjualan produk tertinggi. Terdapat pada Gambar 4.50.



Gambar 4. 50 Tampilan Dashboard

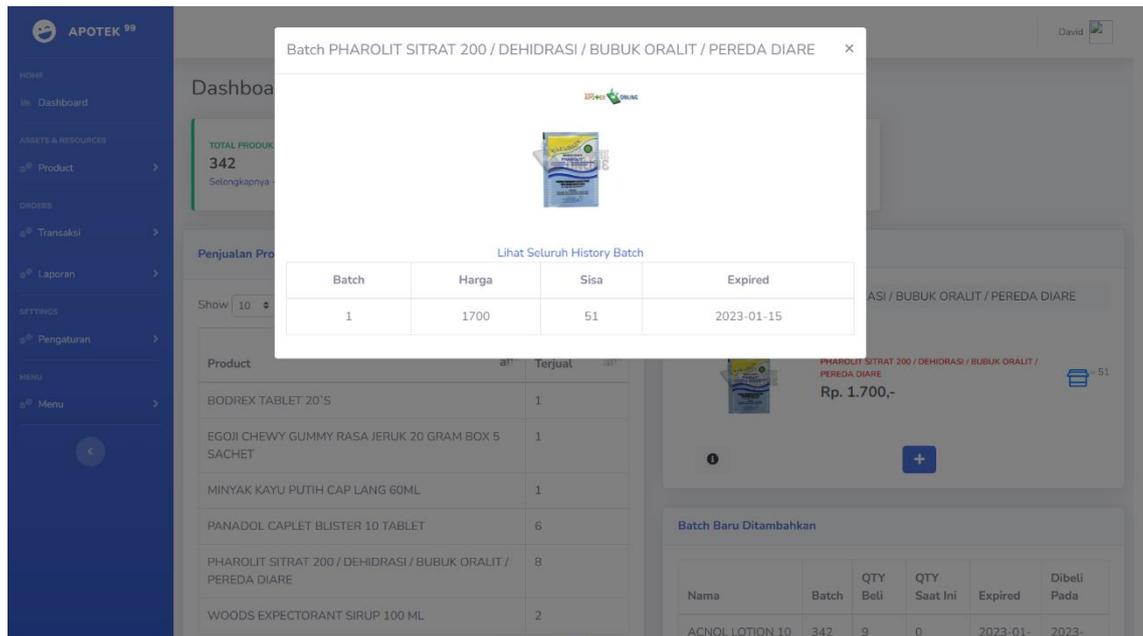
User dapat mencari produk, dengan mengisi kolom input. Apabila produk ditemukan, maka akan muncul menu, untuk memilih produk tersebut. Setelah menu tersebut dipilih, maka produk akan muncul, dengan informasi nama, harga, dan stok produk. Terdapat pada gambar 4.51.



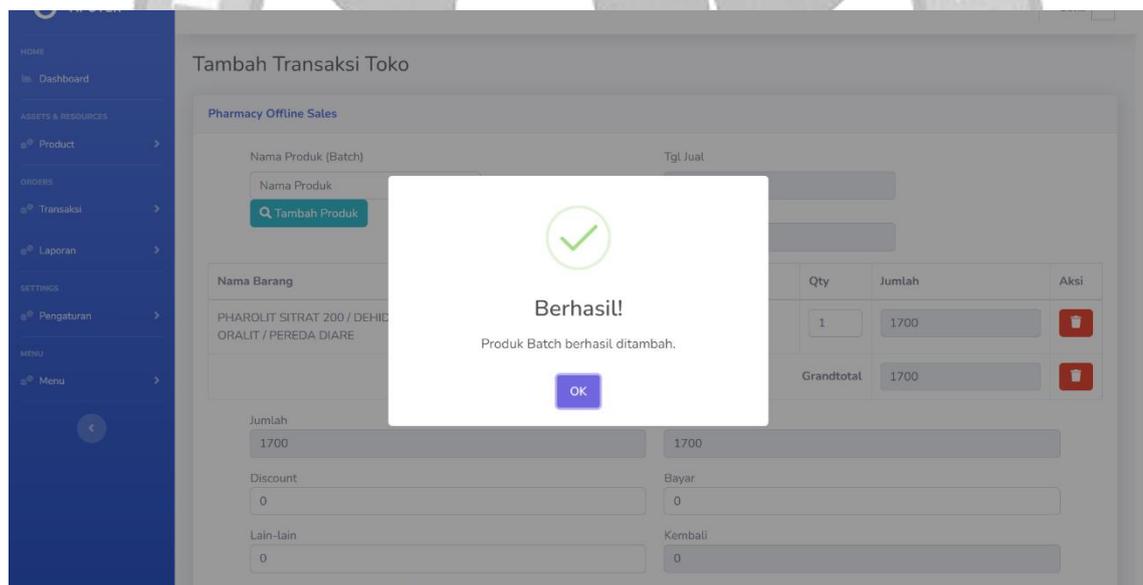
Gambar 4. 51 Tampilan Cari Produk dalam Dashboard

User dapat melihat informasi lebih lanjut tentang batch produk, dengan menekan tombol informasi, berwarna transparan. Bila perlu, user dapat menambahkan produk tersebut ke transaksi offline, dengan

menekan tombol tambah, berwarna biru. Terdapat pada gambar 4.52 dan 4.53.



Gambar 4. 52 Tampilan Informasi Produk

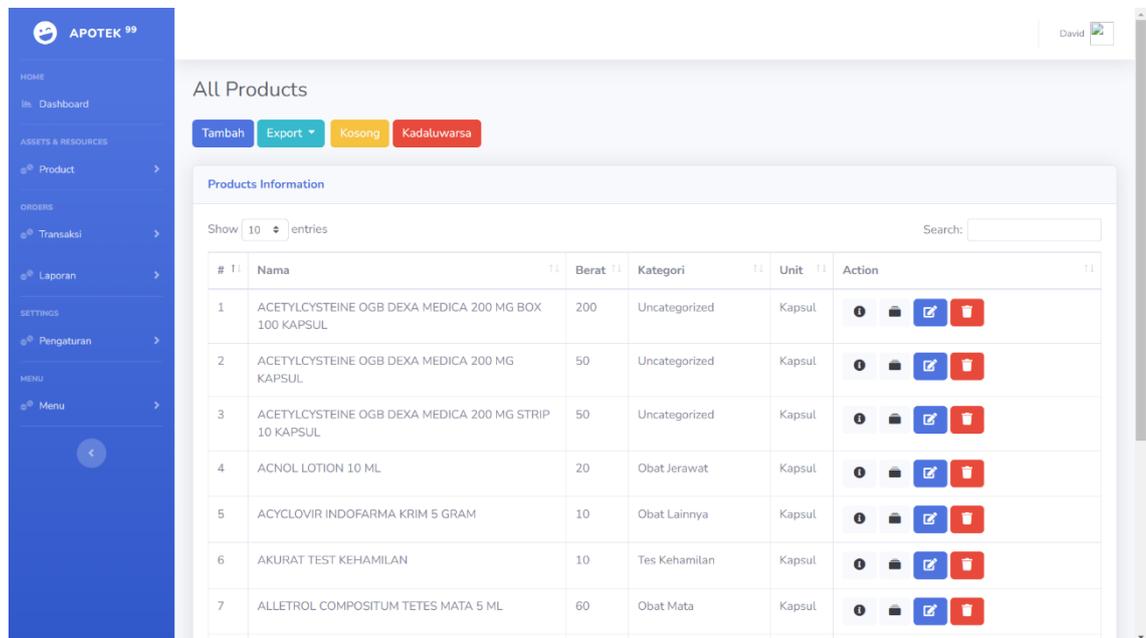


Gambar 4. 53 Tampilan Produk Ditambah ke Transaksi Toko dari Dashboard

5) Kelola Informasi Produk

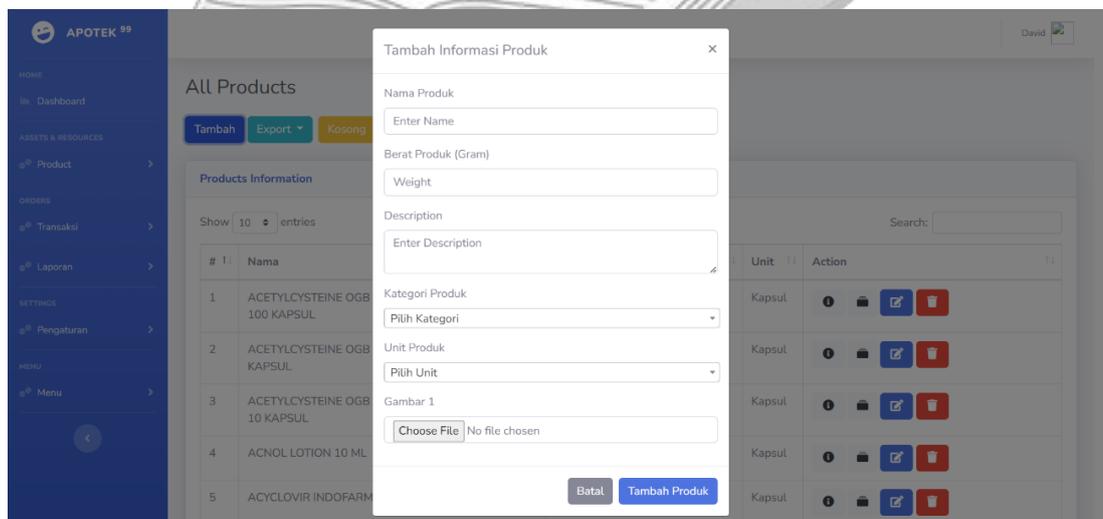
Berikut merupakan tampilan dari menu informasi produk, yang dapat dilihat pada Gambar 4.54. Sistem akan melakukan pembaruan data stok produk dari *SmartSeller* API ke *database*, yang membutuhkan waktu sekitar 40 detik, setelah itu, halaman akan

menampilkan informasi produk. Pada tampilan ini, admin dapat melihat produk, menambahkan produk sesuai ketersediaan, melakukan edit produk, melakukan edit stok, mengosongkan stok, dan menghapus produk. Kasir dapat melihat produk, dengan pembatasan data, seperti data nominal pembelian produk.



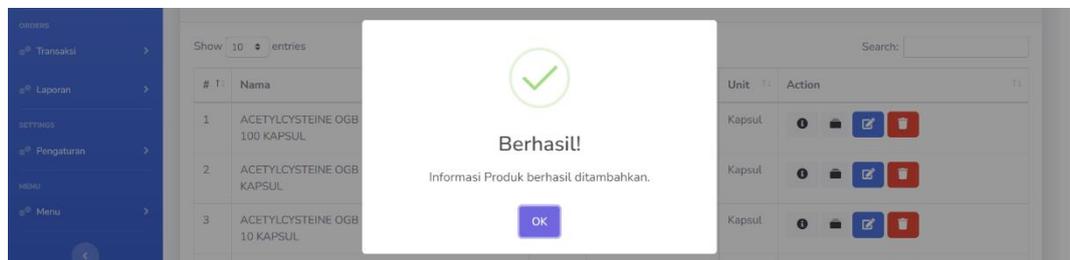
Gambar 4. 54 Tampilan Kelola Produk

Admin dapat menambahkan produk terbaru, dengan menekan tombol tambah, dan sistem akan menampilkan *pop-up* berisi form penambahan produk. *Admin* mengisi data secara lengkap, kemudian menekan tombol Tambah Produk. Dapat dilihat pada Gambar 4.55.



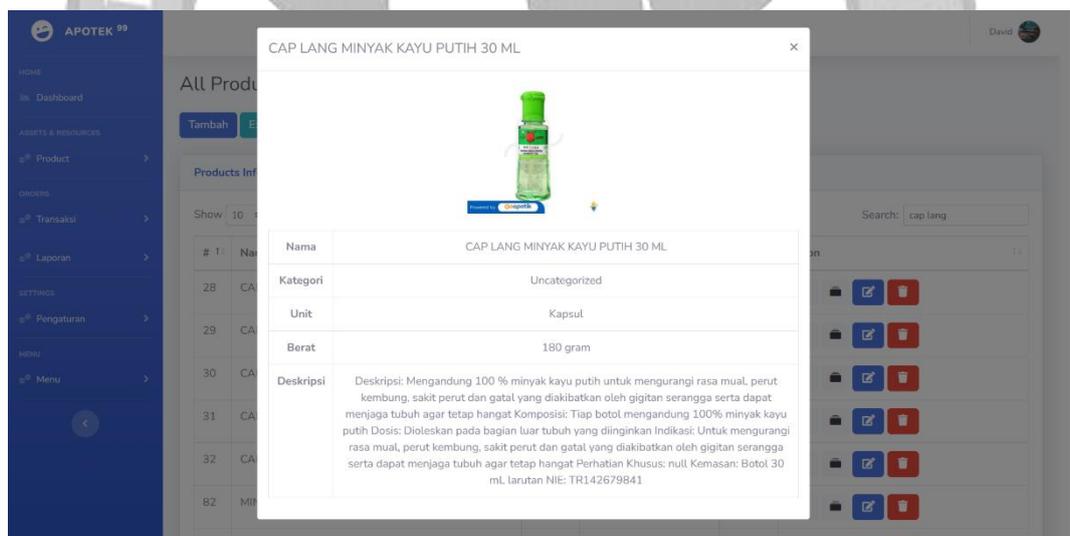
Gambar 4. 55 Tampilan Tambah Produk

Sistem akan memeriksa validasi pada data yang diinput. Jika data tidak valid, maka sistem akan menampilkan pesan *error*. Jika data dinyatakan valid, maka sistem akan menyimpan data terbaru, dan menampilkan pesan sukses. Terdapat pada Gambar 4.56.



Gambar 4. 56 Tampilan Pesan Sukses Penambahan Produk

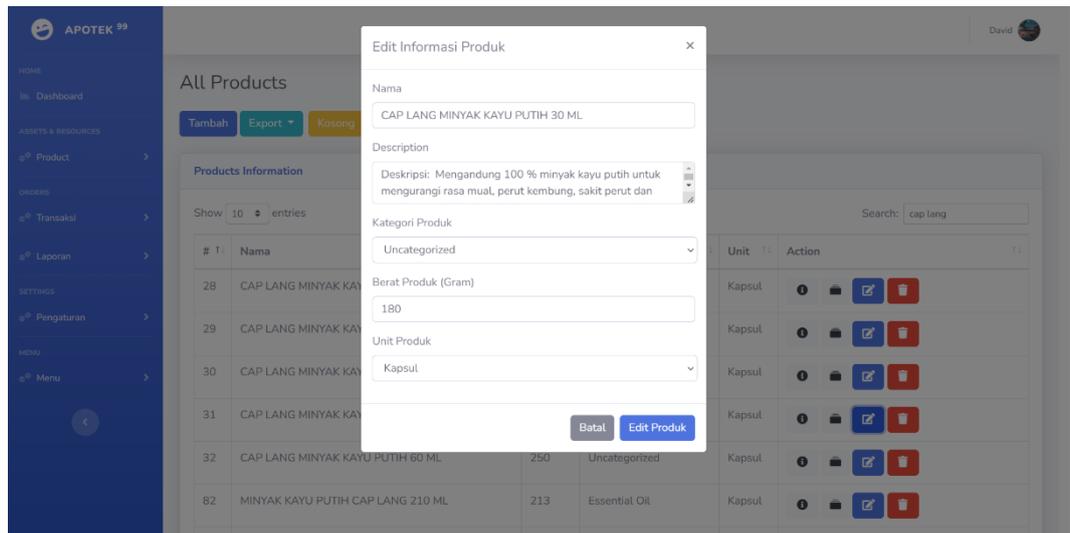
Kemudian, *user* dapat melihat informasi produk dengan menekan tombol informasi. Sistem akan menampilkan halaman informasi produk dalam bentuk jendela *pop-up*. Dapat dilihat pada Gambar 4.57.



Gambar 4. 57 Tampilan Informasi Produk

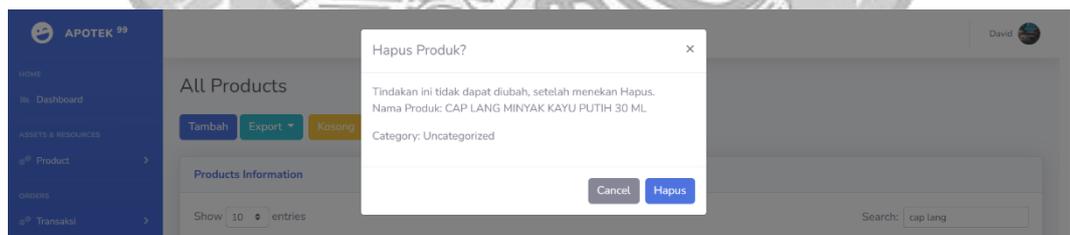
Admin dapat mengedit data informasi produk, dengan menekan tombol edit. Sistem akan menampilkan form edit dalam bentuk jendela *pop-up*. Admin dapat mengubah informasi produk sesuai kebutuhan, lalu menekan tombol edit produk. Sistem akan melakukan validasi terhadap data yang diedit. Jika valid, maka sistem akan

merubah dan menyimpan data tersebut. Dapat dilihat pada Gambar 4.58.



Gambar 4. 58 Tampilan Edit Informasi Produk

Admin dapat menghapus data informasi produk, dengan menekan tombol hapus. Sistem akan menampilkan form konfirmasi hapus dalam bentuk jendela *pop-up*. Jika Admin mengonfirmasi penghapusan produk, maka sistem akan menghapus produk. Dapat dilihat pada Gambar 4.59.

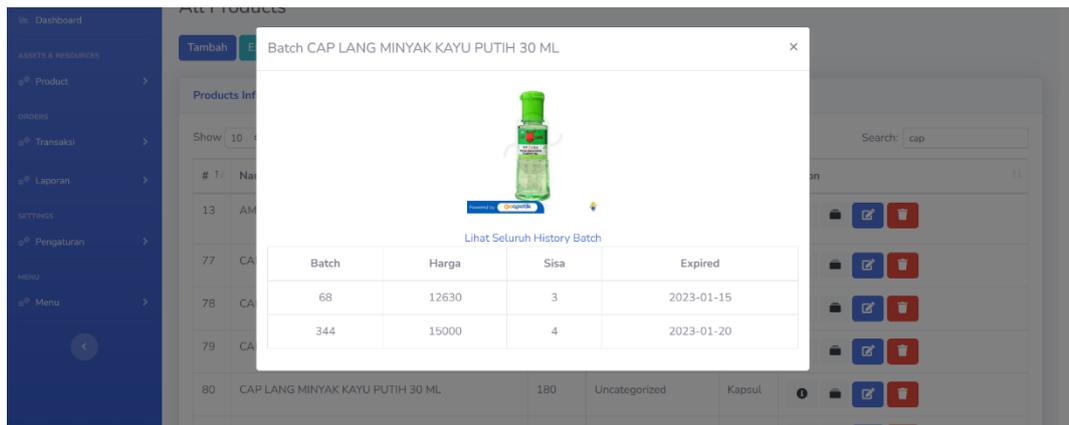


Gambar 4. 59 Tampilan Hapus Produk

6) Kelola Batch Produk

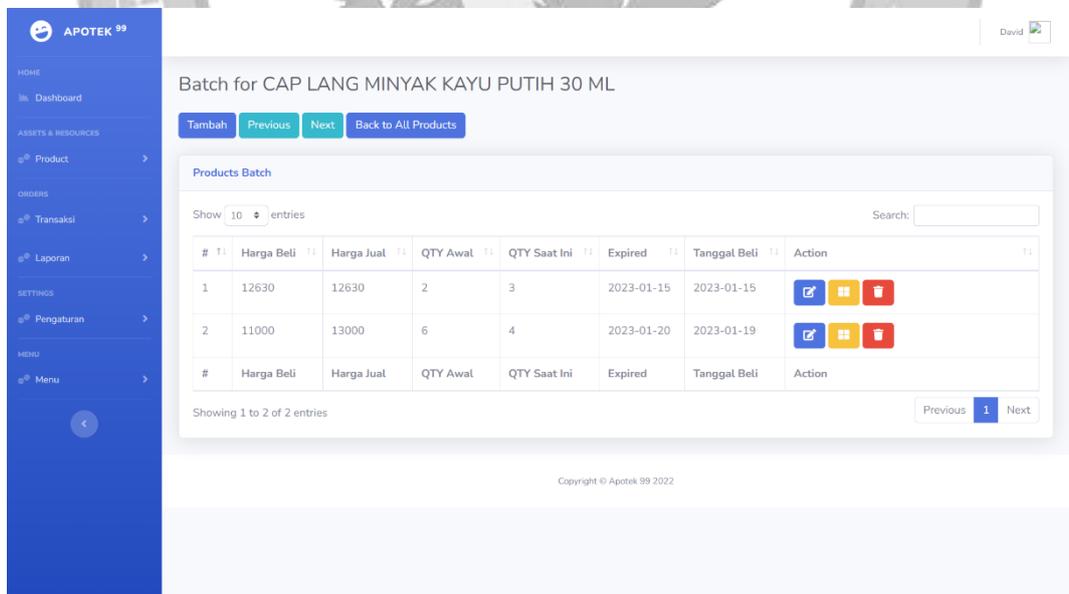
Setelah melalui informasi produk, *user* dapat melihat *batch* produk dengan menekan tombol *batch* berwarna abu-abu. Sistem akan menampilkan data *batch* produk dalam bentuk jendela *pop-up*. Jika

ingin melihat seluruh data *batch*, maka *user* dapat menekan tulisan "Lihat Seluruh History Batch". Dapat dilihat pada Gambar 4.60.



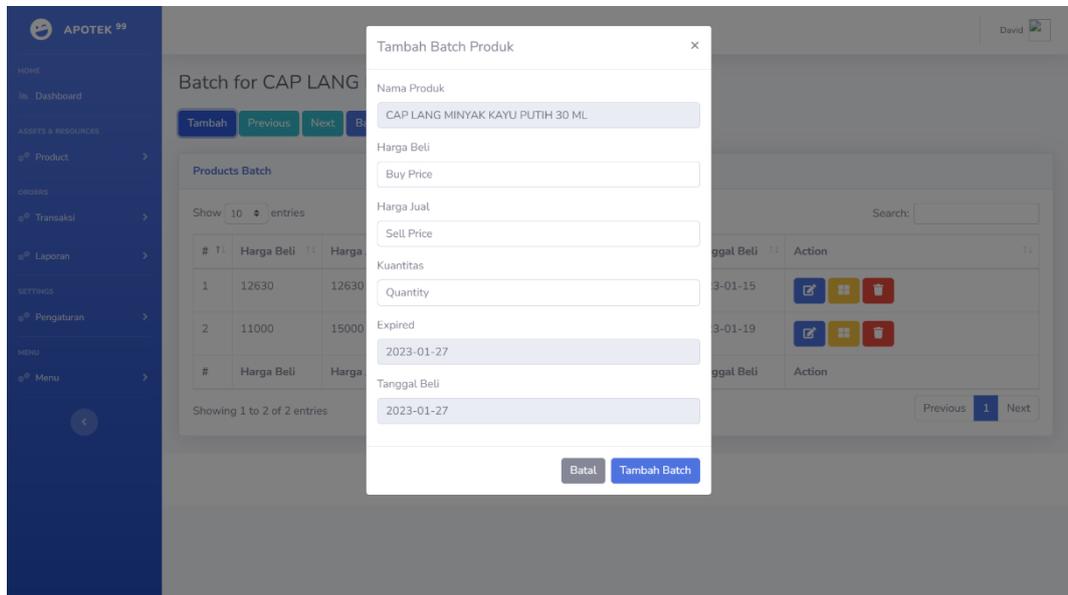
Gambar 4. 60 Tampilan *Batch* Produk dalam Jendela *Pop-Up*

Sistem akan melakukan pembaruan data stok produk dari *SmartSeller* API ke *database*, yang membutuhkan waktu sekitar 40 detik, setelah itu, halaman akan menampilkan *batch* produk yang dipilih oleh *user*. Pada halaman ini, admin dapat menambahkan batch, mengubah batch, mengubah stok *batch*, dan menghapus batch. Dapat dilihat pada Gambar 4.61.



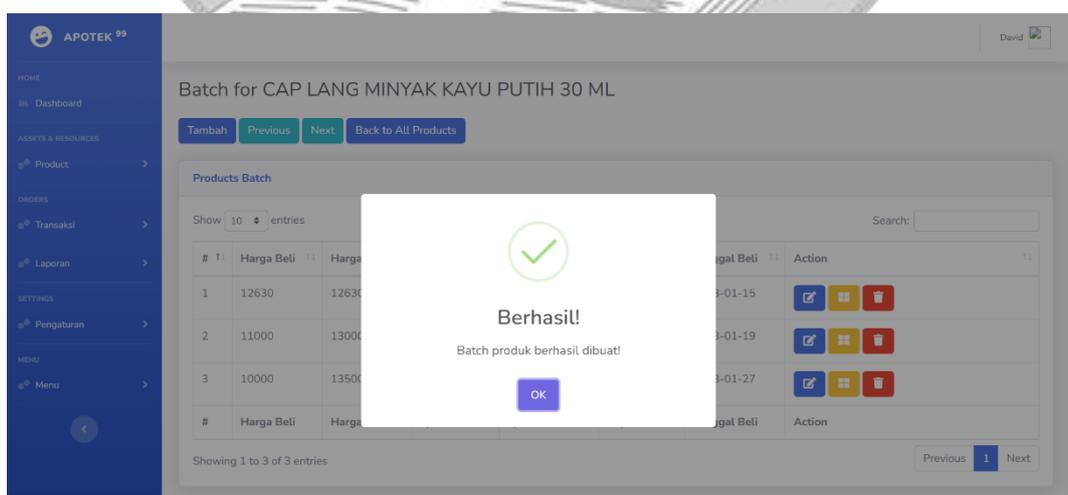
Gambar 4. 61 Tampilan Semua *Batch* dalam Satu Produk

Untuk menambahkan batch produk, admin dapat menekan tombol Tambah, dan akan menghasilkan form tambah dalam bentuk jendela *pop-up*. Dapat dilihat pada Gambar 4.62.



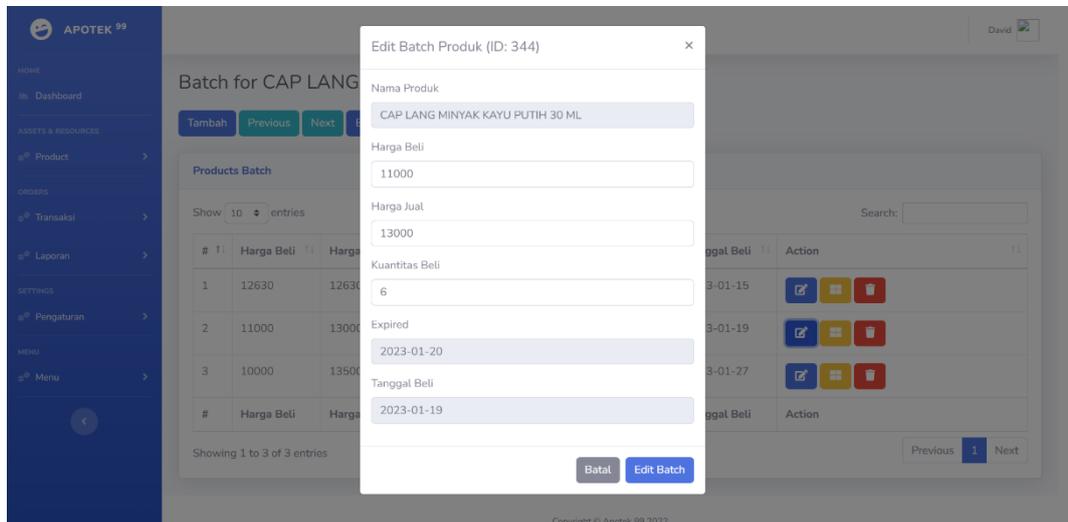
Gambar 4. 62 Tampilan Tambah *Batch*

Sistem akan melakukan validasi untuk data *batch* yang diinput. Jika data valid, maka sistem akan menambahkan data *batch* terbaru, menambahkan produk ke *Smartseller*, dan menghasilkan pesan sukses. Dapat dilihat Gambar 4.63.



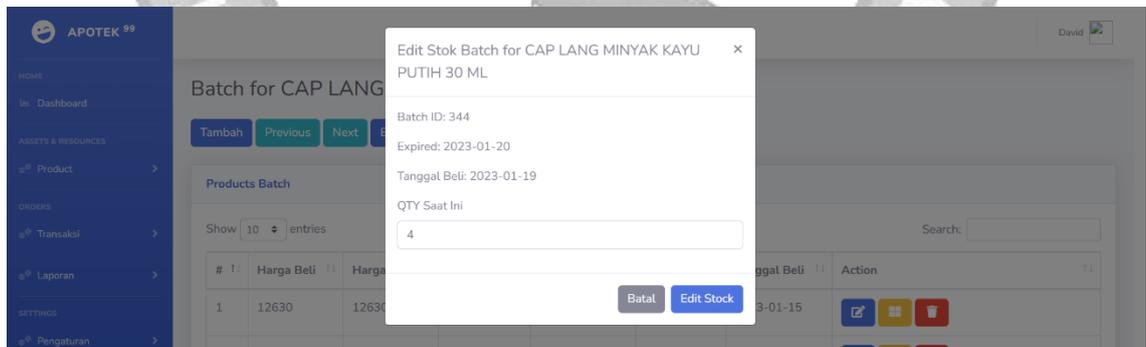
Gambar 4. 63 Pesan Sukses *Batch* Produk Berhasil Dibuat

Selain menambahkan *batch*, admin juga dapat melakukan pengeditan data *batch*. Jika data yang diubah valid, maka sistem akan mengubah data *batch*. Dapat dilihat pada Gambar 4.64.

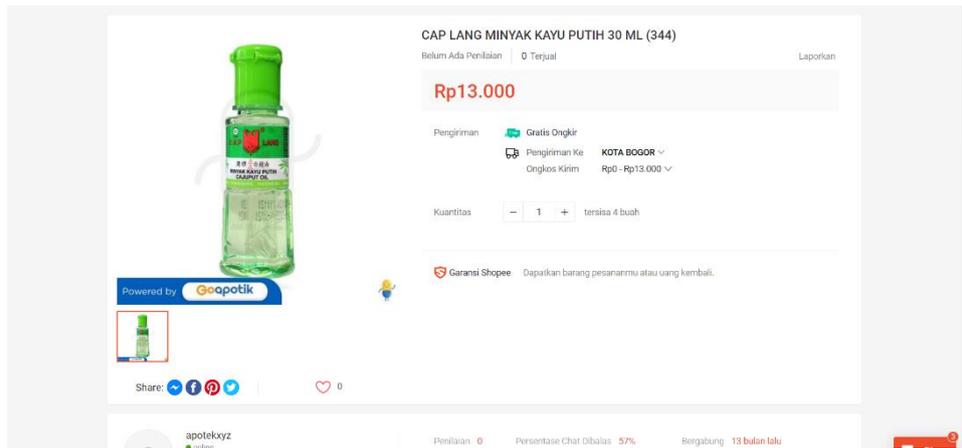


Gambar 4. 64 Pengeditan Data *Batch* Produk

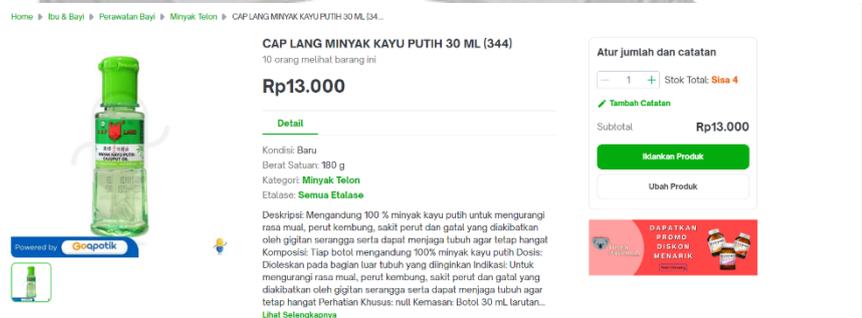
Admin dapat melakukan pengeditan stok *batch* produk dalam sistem, dan akan merubah data stok produk pada *marketplace*. Terdapat pada Gambar 4.65, Gambar 4.66 dan Gambar 4.67.



Gambar 4. 65 Pengeditan Data Stok *Batch* Produk



Gambar 4. 66 Tampilan Halaman Produk Pada Shopee



Gambar 4. 67 Tampilan Halaman Produk Pada Tokopedia

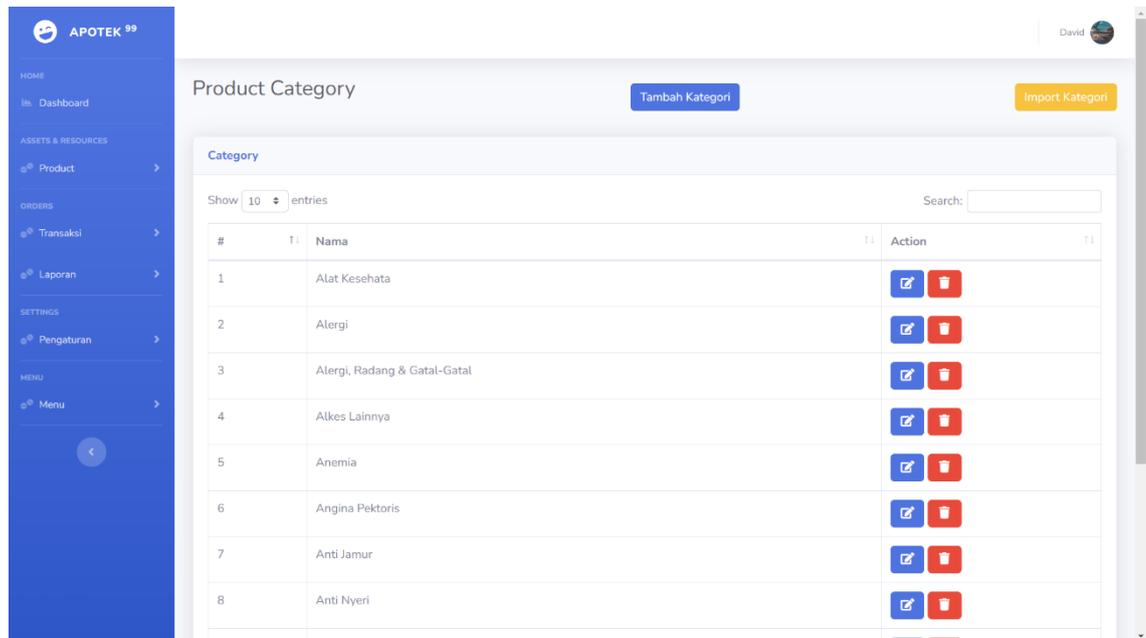
Admin juga dapat melakukan penghapusan *batch* pada produk, dengan menekan hapus. Sistem akan menampilkan pesan konfirmasi dalam bentuk jendela *pop-up*, yang terdapat pada Gambar 4.68. Bila admin mengonfirmasi penghapusan *batch*, maka sistem akan menghapus *batch* produk.



Gambar 4. 68 Konfirmasi Mengosongkan Stok Produk

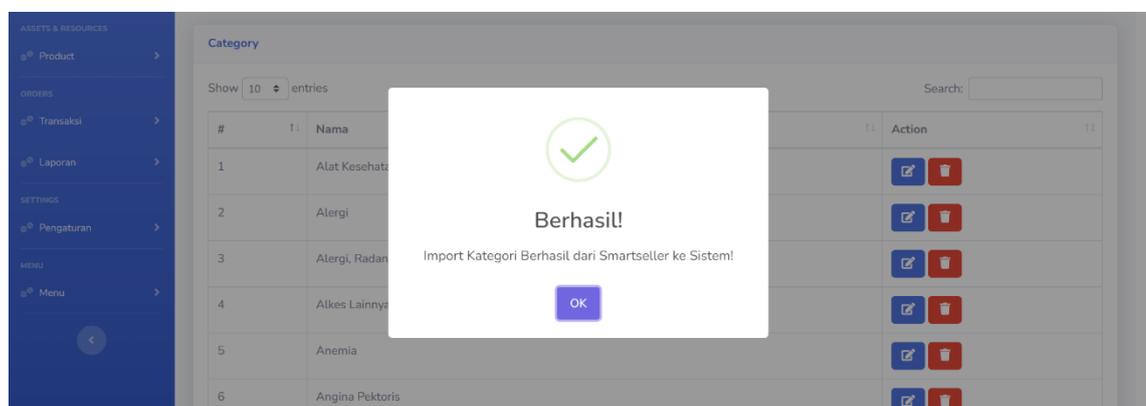
7) Kelola Kategori Produk

Berikut merupakan tampilan dari menu kategori produk, yang dapat dilihat pada Gambar 4.69. Pada tampilan ini, admin dapat melihat kategori, menambahkan kategori sesuai kebutuhan, melakukan edit kategori, melakukan *import* kategori dari SmartSeller, dan menghapus kategori.



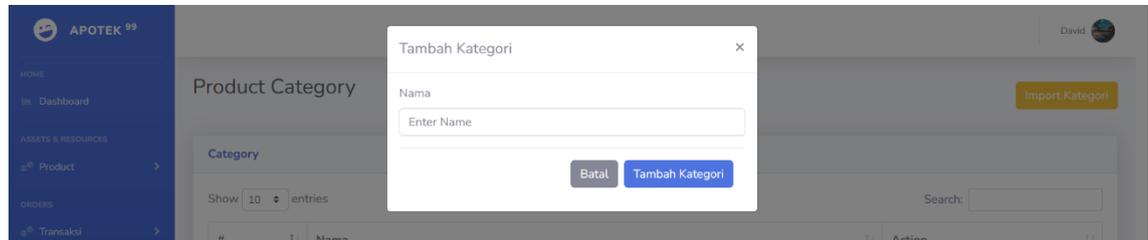
Gambar 4. 69 Tampilan Kelola Kategori Produk

Admin dapat melakukan *import* kategori dari SmartSeller, dengan menekan tombol *import*. Sistem akan mendeteksi seluruh data kategori *SmartSeller*, sistem akan menambahkan data kategori sebagai data baru, untuk data kategori *Smartseller* yang belum terdapat dalam *database*. Dapat dilihat pada Gambar 4.70.



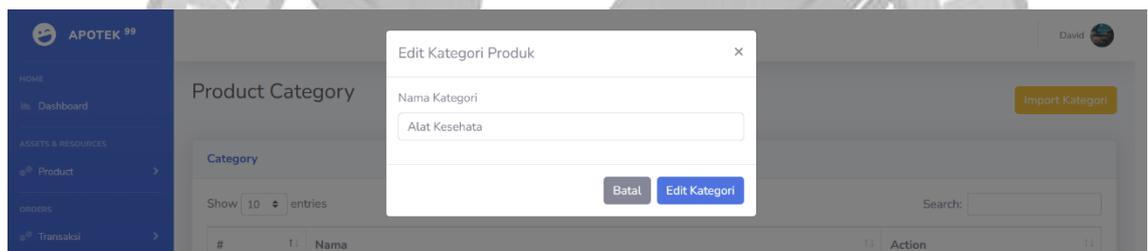
Gambar 4. 70 Tampilan Pesan Berhasil Impor Kategori dari *SmartSeller* API

Selanjutnya, admin dapat menambahkan kategori, yang ditampilkan dalam bentuk jendela *pop-up*. Admin cukup memasukkan nama kategori sesuai yang diinginkan ke dalam *form* tambah kategori. Dapat dilihat pada Gambar 4.71.



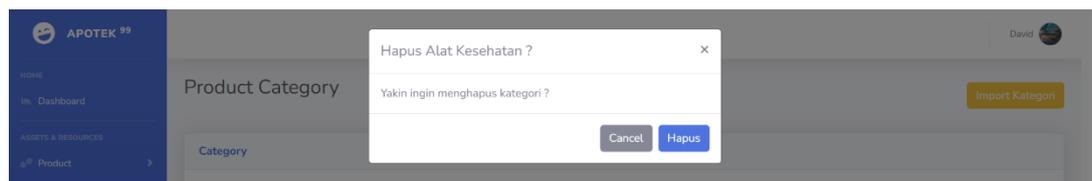
Gambar 4. 71 Tampilan Tambah Kategori

Dalam menu kategori, admin juga dapat melakukan perubahan data. Berikut Gambar 4.72, yang merupakan tampilan form edit kategori yang ditampilkan dalam jendela *pop-up*.



Gambar 4. 72 Tampilan Edit Kategori

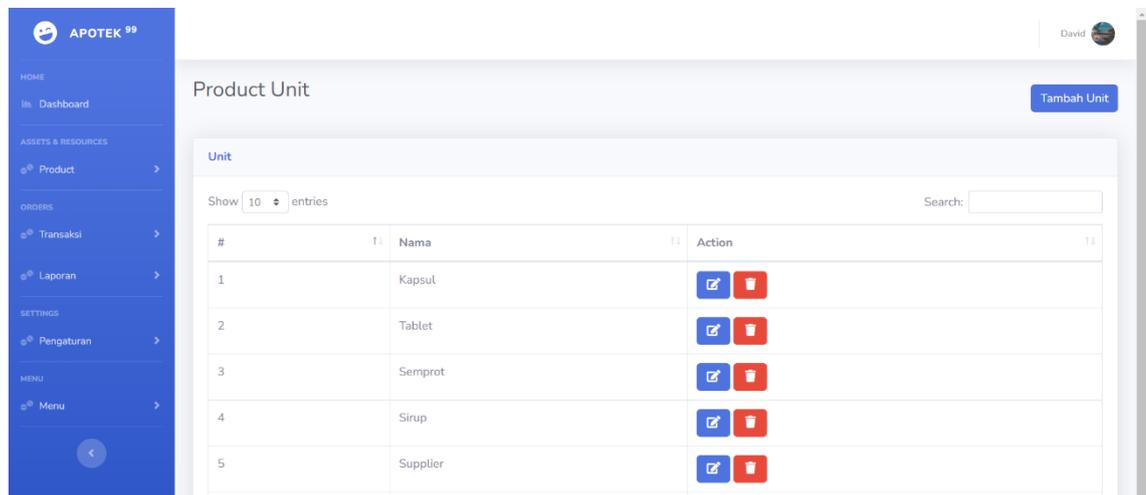
Selain dapat melihat data kategori, menambah data kategori, melakukan *import* data kategori, dan melakukan pengeditan data kategori, admin dapat menghapus kategori, jika diperlukan. Jika admin ingin menghapus data kategori, maka sistem akan menampilkan pesan konfirmasi dalam bentuk jendela *pop-up*, yang terdapat pada Gambar 4.73.



Gambar 4. 73 Tampilan Hapus Kategori

8) Kelola Unit Produk

Berikut merupakan tampilan dari menu pengelolaan unit produk, yang dapat dilihat pada Gambar 4.74. Pada tampilan ini, admin dapat melihat unit, menambahkan unit sesuai kebutuhan, melakukan edit unit, dan *delete unit*.



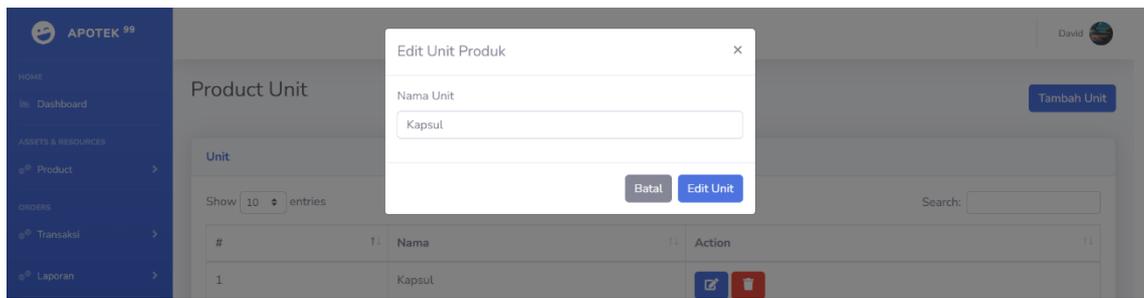
Gambar 4. 74 Tampilan Kelola Unit Produk

Berikut merupakan tampilan dari *add new unit* yang ditampilkan dalam bentuk jendela *pop-up*. Jika admin ingin menambahkan data unit produk, maka admin cukup memasukkan nama unit sesuai dengan yang diinginkan ke dalam *form add new unit*. Dapat dilihat pada Gambar 4.75.



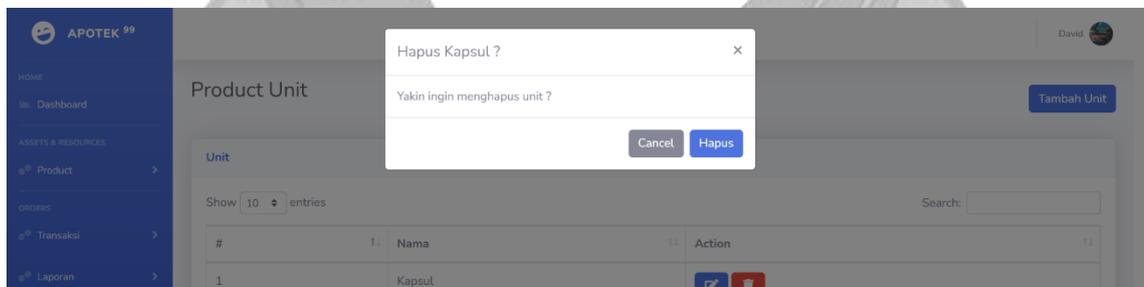
Gambar 4. 75 Tampilan Tambah Unit Produk

Dalam menu unit produk, admin juga dapat melakukan perubahan data. Berikut Gambar 4.76 merupakan tampilan *form edit* data unit yang ditampilkan dalam bentuk jendela *pop-up*.



Gambar 4. 76 Tampilan Edit Unit Produk

Selain dapat melihat data unit, melakukan penambahan unit dan melakukan edit unit, pada menu ini, admin juga dapat melakukan penghapusan unit jika diperlukan. Jika admin ingin menghapus unit, maka sistem akan menampilkan pesan konfirmasi penghapusan seperti yang dapat dilihat pada Gambar 4.77.



Gambar 4. 77 Tampilan Hapus Unit Produk

9) Lihat Batch Produk Kadaluwarsa

Berikut merupakan tampilan dari menu *batch* produk kadaluwarsa. Pada halaman ini, sistem akan menampilkan tabel *batch* produk *expired*. *Batch* produk *expired* adalah stok produk yang telah melewati tanggal *expired*. Dapat dilihat pada Gambar 4.78.

All Batch Expired

Batch Hampir Kadaluwarsa

Produk Kadaluwarsa

Show 10 entries Search:

#	Nama	Batch	QTY Beli	QTY Saat Ini	Expired	Last Update	Action
1	PHAROLIT SITRAT 200 / DEHIDRASI / BUBUK ORALIT / PEREDA DIARE	1	52	52	2023-01-15	2023-01-17 12:44:30	[Yellow Trash Icon]
2	MINYAK KAYU PUTIH CAP LANG 60ML	2	4	3	2023-01-15	2023-01-17 13:06:50	[Yellow Trash Icon]
3	BODREX TABLET 20'S	3	10	8	2023-01-15	2023-01-17 13:05:46	[Yellow Trash Icon]
4	PANADOL CAPLET BLISTER 10 TABLET	4	10	7	2023-01-15	2023-01-17 12:30:42	[Yellow Trash Icon]
5	BIOGESIC 1 STRIP 4 TAB	5	21	21	2023-01-15	2023-01-15 08:42:43	[Yellow Trash Icon]
6	BINTANG TOEDJOE PUYER NO.16 SACHET	7	24	27	2023-01-15	2023-01-15 08:42:43	[Yellow Trash Icon]

Gambar 4. 78 Tampilan Produk *Expired* / Hampir *Expired*

Admin mempunyai opsi untuk mengosongkan stok *batch* produk, dengan menekan tombol berwarna kuning. Sistem akan menampilkan pesan konfirmasi, apakah stok *batch* produk ingin dikosongkan. Jika admin mengonfirmasi, maka sistem akan mengosongkan stok *batch* produk. Dapat dilihat pada Gambar 4.79.



Gambar 4. 79 Halaman Konfirmasi Pengosongan Stok *Batch* Produk

10) Lihat Produk Kosong

Berikut merupakan tampilan dari menu produk kosong. Pada tampilan ini, sistem akan menampilkan tabel produk yang belum memiliki *batch* atau telah memiliki *batch* dengan stok kosong. Dapat dilihat pada Gambar 4.80.

APOTEK 99

HOME

- Dashboard

ASSETS & RESOURCES

- Product

ORDERS

- Transaksi
- Laporan

SETTINGS

- Pengaturan

MENU

- Menu

All Products Empty

Produk Hampir Kosong

Produk Kosong

Show 10 entries

Search:

#	Ti	Nama	Berat	Kategori	Unit	Action
1		asdasd	230	Uncategorized	Kapsul	
2		Bisolvon	200	Uncategorized	Kapsul	
3		CAP LANG MINYAK KAYU PUTIH 210 ML	210	Uncategorized	Kapsul	
4		COUNTERPAIN 1 TUBE ISI 30 GR	44	Salep Otot	Kapsul	
5		DAKTARIN CREAM 10 GRAM	19	Kulit	Kapsul	
6		DAKTARIN CREAM 5 GRAM	13	Kulit	Kapsul	
7		gen v	100	Uncategorized	Kapsul	
8		ilop	300	Uncategorized	Kapsul	

Gambar 4. 80 Tampilan Produk dengan *Batch* Kosong

Admin dapat menambahkan *batch* baru pada produk, dengan menekan tombol tambah berwarna biru. Sistem akan menampilkan form dalam bentuk *pop-up*. Admin mengisi data *batch* produk, sesuai nama produk yang ditampilkan, lalu menambahkan *batch*. Jika data yang diinput valid, maka sistem akan menambahkan *batch* terbaru ke *database*. Dapat dilihat pada Gambar 4. 81.

APOTEK 99

HOME

- Dashboard

ASSETS & RESOURCES

- Product

ORDERS

- Transaksi
- Laporan

SETTINGS

- Pengaturan

MENU

- Menu

All Products Empty

Produk Hampir Kosong

Produk Kosong

Show 10 entries

Search:

Tambah Batch Produk

Nama Produk

Bisolvon

Harga Beli

Buy Price

Harga Jual

Sell Price

Kuantitas

Quantity

Expired

2023-01-27

Tanggal Beli

2023-01-27

Batal Tambah Batch

#	Ti	Nama	Berat	Kategori	Unit	Action
1		asdasd	230	Uncategorized	Kapsul	
2		Bisolvon	200	Uncategorized	Kapsul	
3		CAP LANG MINYAK KAYU PUTIH 210 ML	210	Uncategorized	Kapsul	
4		COUNTERPAIN 1 TUBE ISI 30 GR	44	Salep Otot	Kapsul	
5		DAKTARIN CREAM 10 GRAM	19	Kulit	Kapsul	
6		DAKTARIN CREAM 5 GRAM	13	Kulit	Kapsul	
7		gen v	100	Uncategorized	Kapsul	
8		ilop	300	Uncategorized	Kapsul	

Gambar 4. 81 Tampilan Form Penambahan *Batch* Produk

11) Tambah Transaksi Toko

Berikut merupakan tampilan dari menu tambah transaksi toko. Saat halaman dibuka, membutuhkan waktu sekitar 40 detik, untuk memperbarui data stok produk dari *Smartseller API* ke *database*, kemudian menampilkan halaman. Halaman ini dapat diakses oleh admin dan kasir, untuk menambahkan transaksi di apotek. Fitur ini dapat menambahkan keranjang, mengubah keranjang, dan menghapus keranjang. Dapat dilihat pada Gambar 4.82.

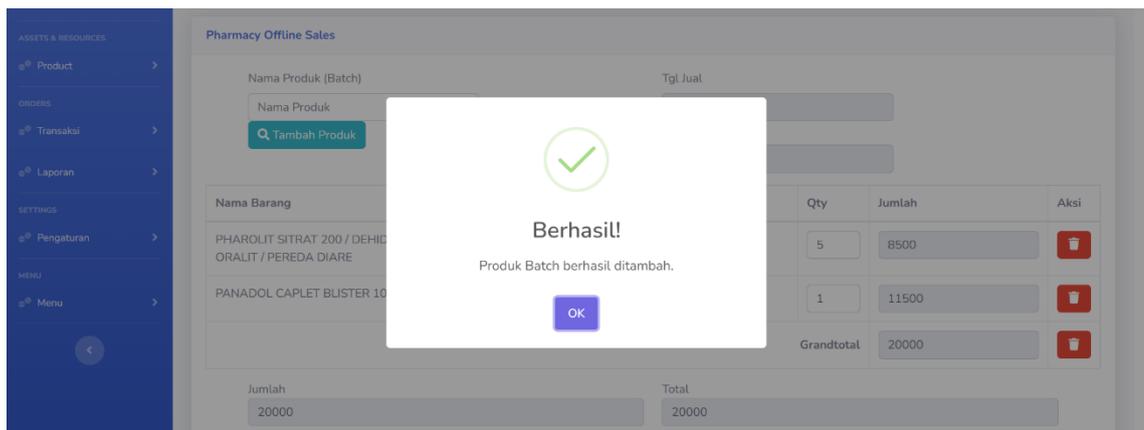
The screenshot displays the 'Tambah Transaksi Toko' (Add Store Transaction) page. On the left is a blue sidebar menu with options like 'HOME', 'Dashboard', 'ASSETS & RESOURCES', 'Product', 'ORDERS', 'Transaksi', 'Laporan', 'SETTINGS', 'Pengaturan', and 'MENU'. The main content area is titled 'Pharmacy Offline Sales' and includes a search bar for 'Nama Produk (Batch)', a date picker for 'Tgl Jual' (27/1/2023), and a 'No. Resi' field (OFF_0000020). Below this is a table with columns: Nama Barang, Batch, Harga, Unit, Stock, Qty, Jumlah, and Aksi. One item is listed: PHAROLIT SITRAT 200 / DEHIDRASI / BUBUK ORALIT / PEREDA DIARE, with a quantity of 5 and a total value of 8500. A 'Grandtotal' row shows a total of 8500. At the bottom, there are input fields for 'Jumlah' (8500), 'Total' (8500), 'Discount' (0), 'Bayar' (0), 'Lain-lain' (0), and 'Kembali' (0). A note at the bottom states: 'Discount, Lain-lain, Bayar, dan Kembali akan direset ke nol, jika halaman direfresh atau mengubah produk'.

Gambar 4. 82 Tampilan Tambah Transaksi Toko

Untuk menambahkan produk ke keranjang, *user* cukup mengetik nama produk sesuai kebutuhan. Sistem akan menampilkan *dropdown* produk dari *keyword* yang diketik. *User* menekan tombol Tambah Produk. Sistem akan melakukan validasi data produk di keranjang. Jika datanya valid, maka sistem akan menambahkan data produk ke keranjang. Dapat dilihat pada Gambar 4.83 dan Gambar 4.84.

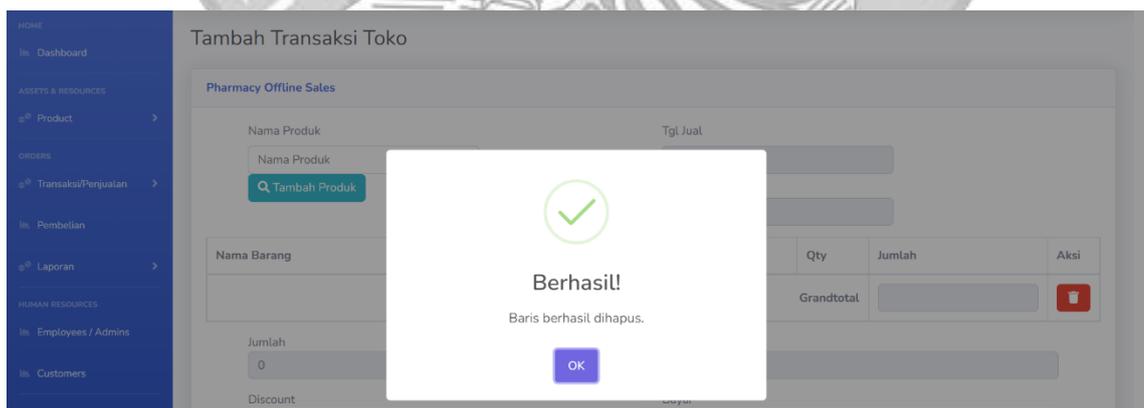


Gambar 4. 83 Tampilan *Dropdown* Pencarian Produk Dalam Transaksi



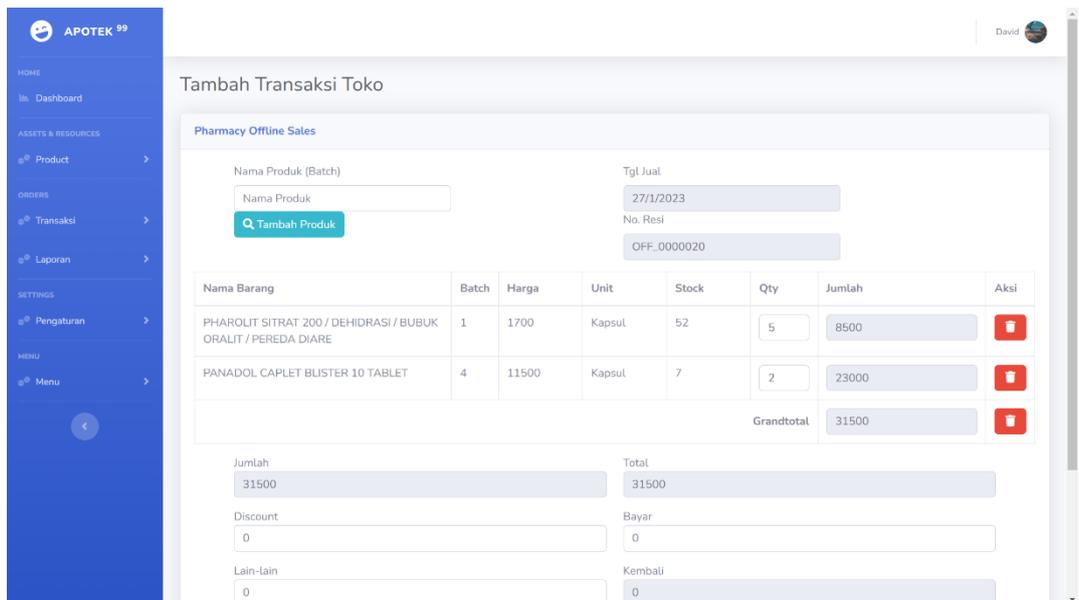
Gambar 4. 84 Tampilan Produk Berhasil Ditambah ke Keranjang Transaksi

Admin dapat menghapus baris produk, atau bila perlu, menghapus semua baris produk dalam satu tombol. Sistem akan menghapus data keranjang. Dapat dilihat pada Gambar 4.85.



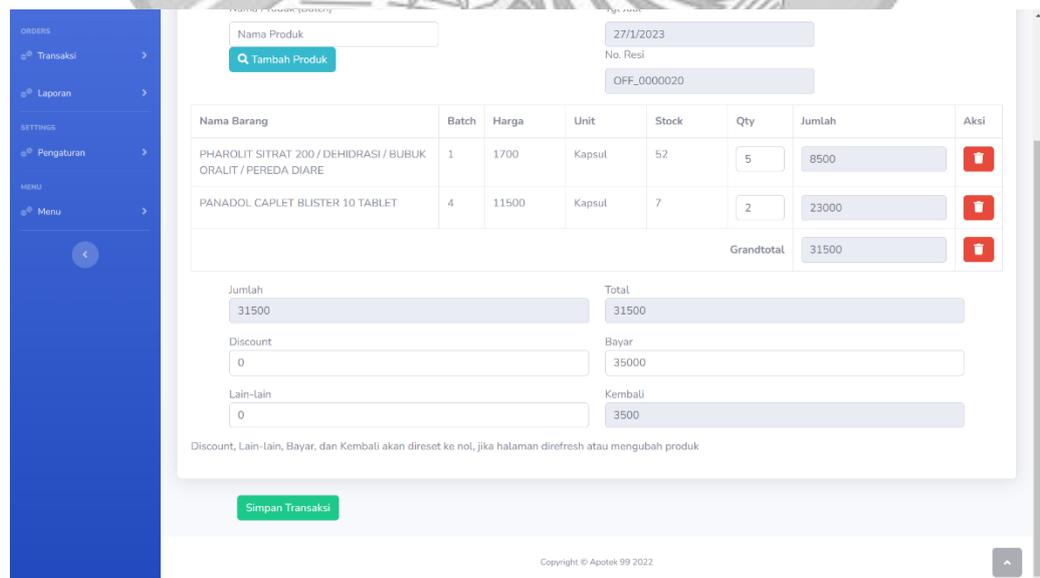
Gambar 4. 85 Tampilan Keranjang Transaksi Berhasil Dihapus

User dapat mengubah kuantitas stok produk yang ingin diubah. Sistem akan mengubah data kuantitas pada baris keranjang, dan menghitung ulang total belanja. Terdapat pada Gambar 4.86.



Gambar 4. 86 Tampilan Ubah Keranjang Transaksi Toko

Jika *user* ingin menyimpan data transaksi, maka *user* dapat melanjutkan dengan mengisi kolom input pada diskon, lain-lain, dan bayar. Kolom input pada diskon dan lain-lain bersifat opsional untuk diisi, dan kolom input pada bayar wajib diisi. Setelah *user* menginput data, maka sistem akan menampilkan hasil hitungan dari ketiga kolom input tersebut. Setelah selesai, *user* dapat menekan tombol simpan transaksi. Dapat dilihat pada Gambar 4.87.

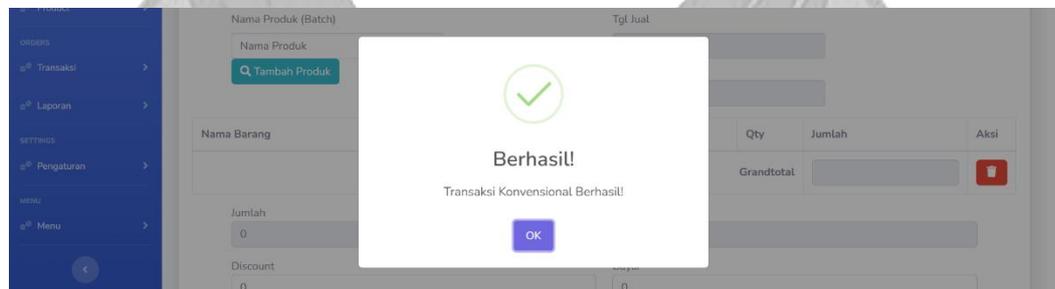


Gambar 4. 87 Tampilan Isi Kolom Input pada Transaksi Toko

Sistem akan melakukan validasi terhadap data yang diinput. Jika tidak valid, maka sistem akan menampilkan pesan *error*. Jika valid, maka sistem akan menyimpan data keranjang ke transaksi. Dapat dilihat pada Gambar 4.88 dan Gambar 4.89.



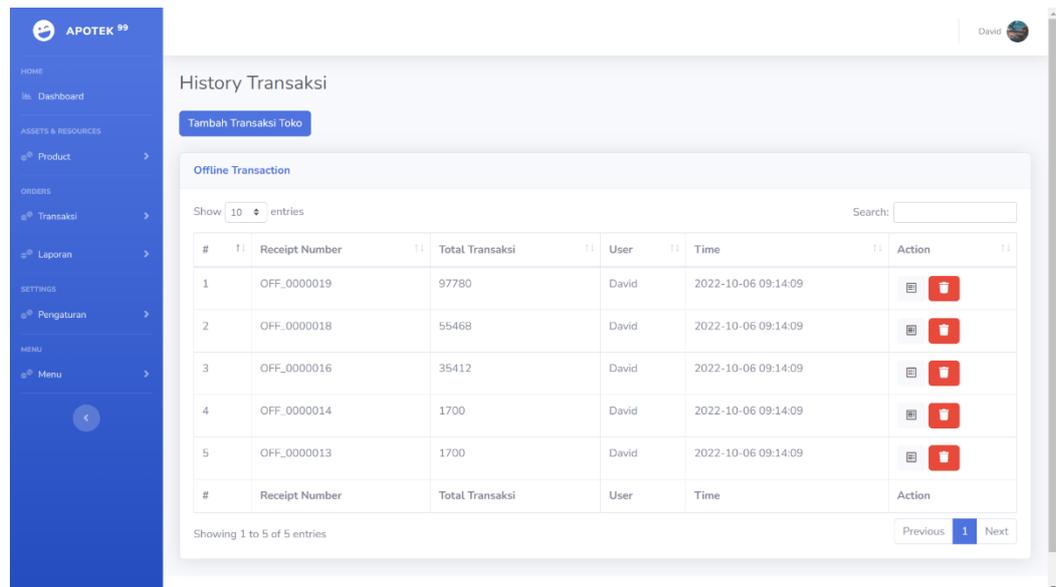
Gambar 4. 88 Tampilan Pesan *Error* pada Penyimpanan Transaksi



Gambar 4. 89 Tampilan Pesan Sukses pada Penyimpanan Transaksi

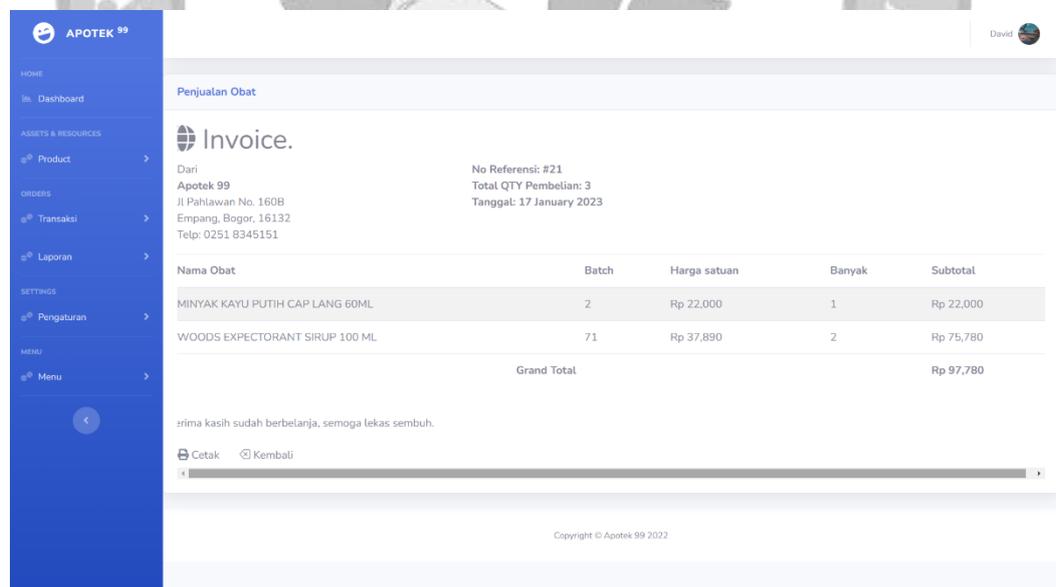
12) Lihat Transaksi

Berikut merupakan tampilan dari menu produk lihat transaksi. Pada tampilan ini, sistem akan menampilkan data dari transaksi yang telah dibuat pada transaksi toko. Dapat dilihat pada Gambar 4.90.



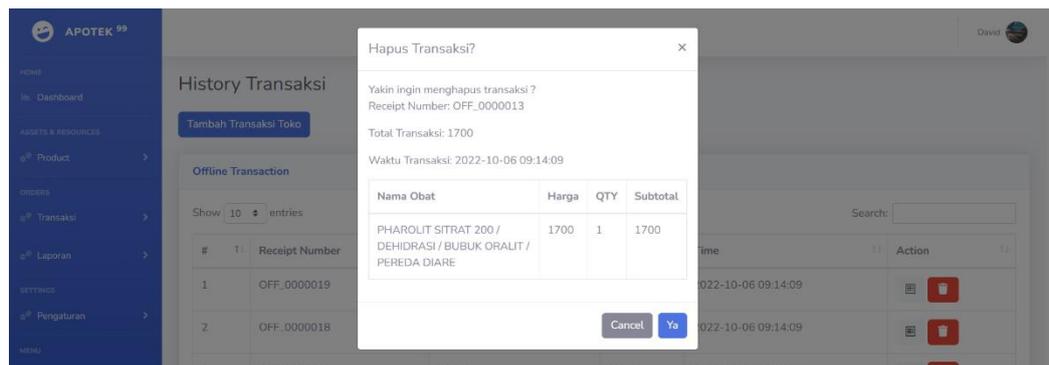
Gambar 4. 90 Tampilan *History* Transaksi

User dapat melihat detail transaksi, dengan menekan tombol dengan *icon* kertas. Sistem akan menampilkan detail transaksi dalam bentuk *invoice*. Dapat dilihat pada Gambar 4.91.



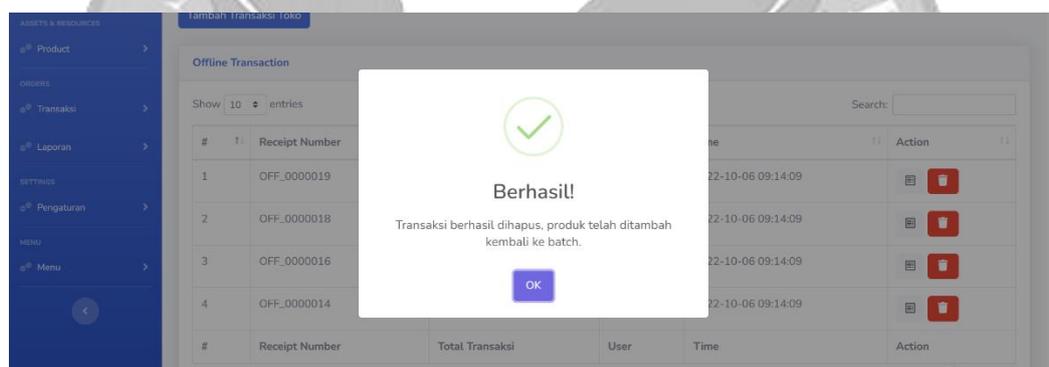
Gambar 4. 91 Tampilan *Invoice* Transaksi

User dapat menghapus transaksi, dengan menekan tombol hapus. Sistem akan menampilkan pesan konfirmasi dalam bentuk *pop-up*. Dapat dilihat pada Gambar 4.92.

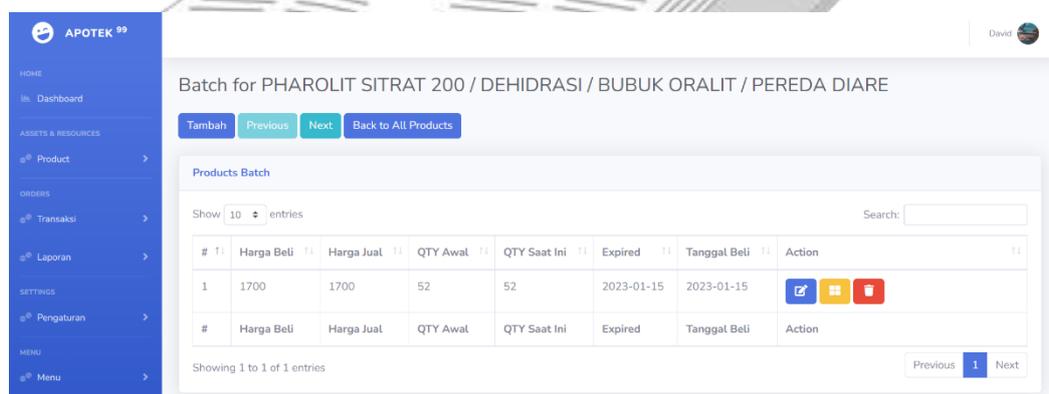


Gambar 4. 92 Tampilan Pesan Konfirmasi Hapus Transaksi

Jika user mengonfirmasi penghapusan transaksi, maka sistem akan menghapus transaksi, dan menambahkan kuantitas dari transaksi ke *batch* produk. Dapat dilihat pada Gambar 4.93 dan Gambar 4.94.



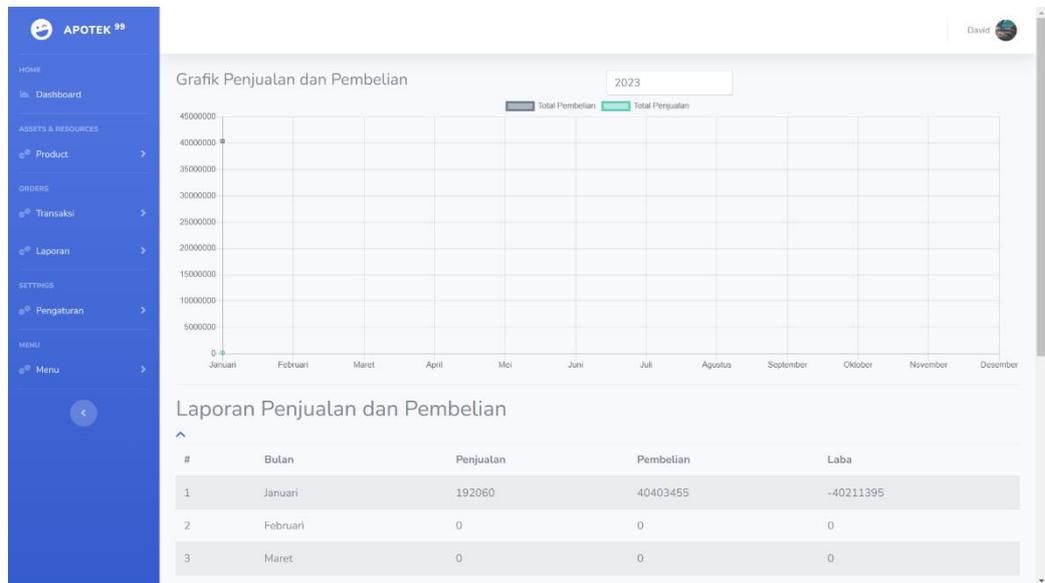
Gambar 4. 93 Tampilan Pesan Berhasil Hapus Transaksi



Gambar 4. 94 Halaman Stok Produk Bertambah setelah Hapus Transaksi

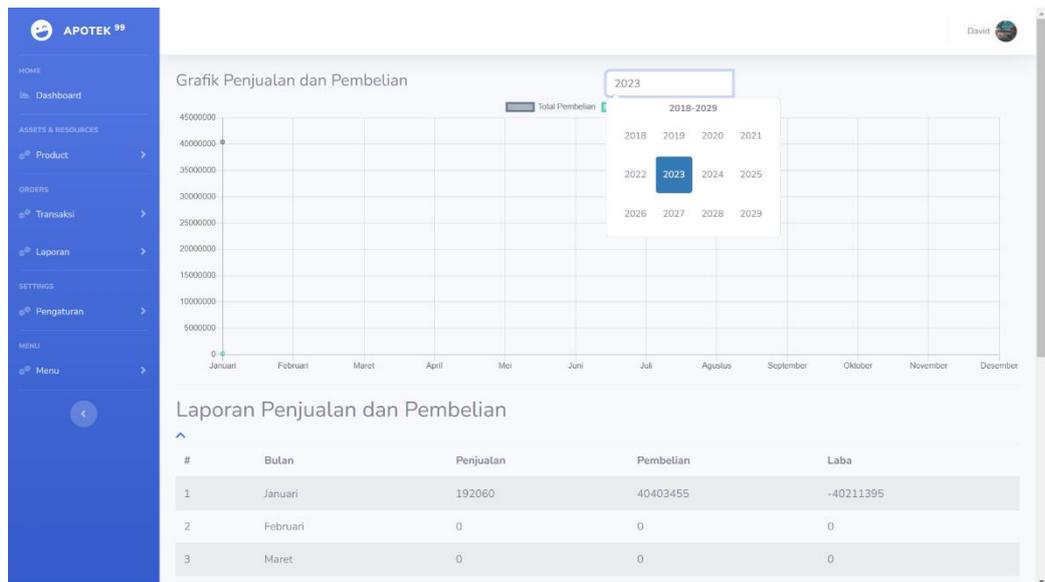
13) Lihat Laporan

Berikut merupakan tampilan dari menu laporan, yang terdiri dari laporan penjualan dan pembelian. Sistem akan menampilkan laporan laba jual beli sesuai tahun. Jika tahun sekarang adalah 2023, maka sistem akan menampilkan laporan tahun 2023. Dapat dilihat pada Gambar 4.95.



Gambar 4. 95 Grafik dan Laporan Penjualan dan Pembelian

Admin dapat menentukan tahun laporan, dan sistem akan menampilkan laporan pada tahun yang ditentukan. Laporan ditampilkan dalam bentuk grafik dan tabel. Dapat dilihat pada Gambar 4.96 dan Gambar 4.97.



Gambar 4. 96 Pemilihan Tahun Laporan Penjualan dan Pembelian

Laporan Penjualan dan Pembelian

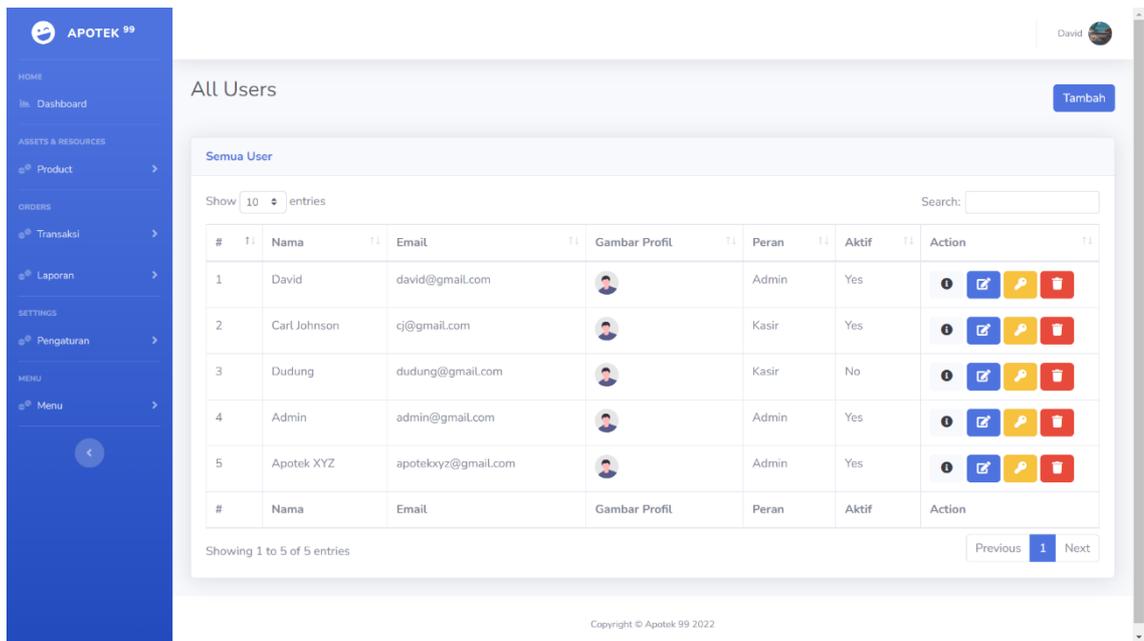
#	Bulan	Penjualan	Pembelian	Laba
1	Januari	192060	40403455	-40211395
2	Februari	0	0	0
3	Maret	0	0	0
4	April	0	0	0
5	Mei	0	0	0
6	Juni	0	0	0
7	Juli	0	0	0
8	Agustus	0	0	0
9	September	0	0	0
10	Oktober	0	0	0
11	November	0	0	0
12	Desember	0	0	0
#	Total	192060	40403455	-40211395

Copyright © Apotek 99 2022

Gambar 4. 97 Laporan Penjualan dan Pembelian

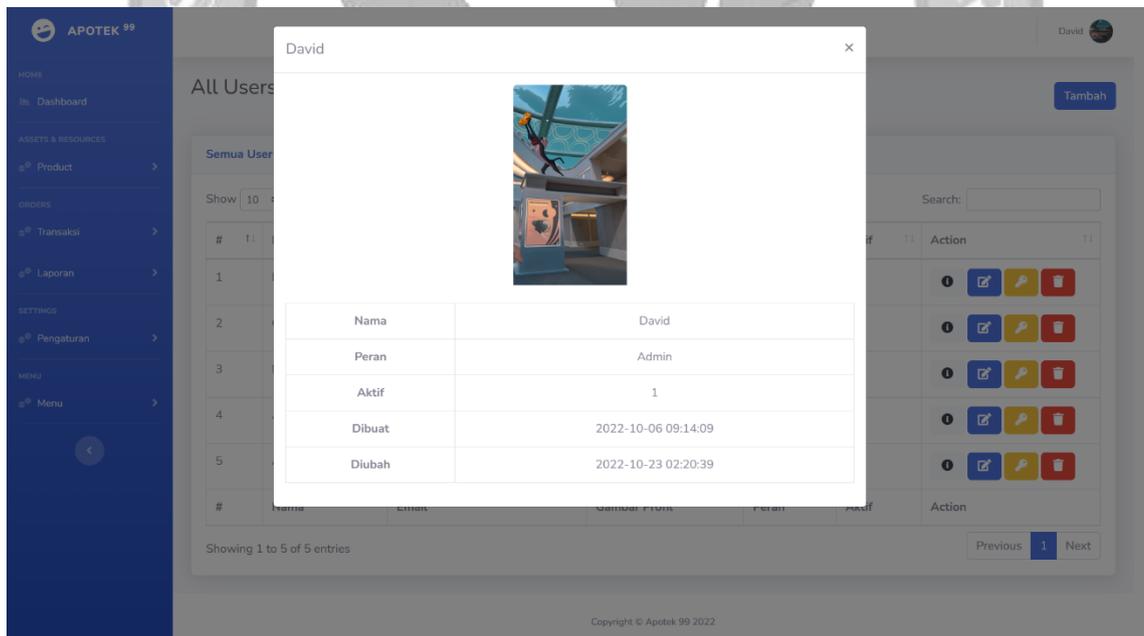
14) Kelola User

Berikut merupakan tampilan dari menu *user*, yang terdiri dari admin dan kasir. Pada menu ini, admin dapat melihat seluruh data *user*, yang terdapat pada Gambar 4.98. Pada tampilan ini, admin dapat melihat user, menambahkan *user* sesuai kebutuhan, melakukan edit user, melakukan *change password*, dan *delete user*.



Gambar 4. 98 Tampilan User

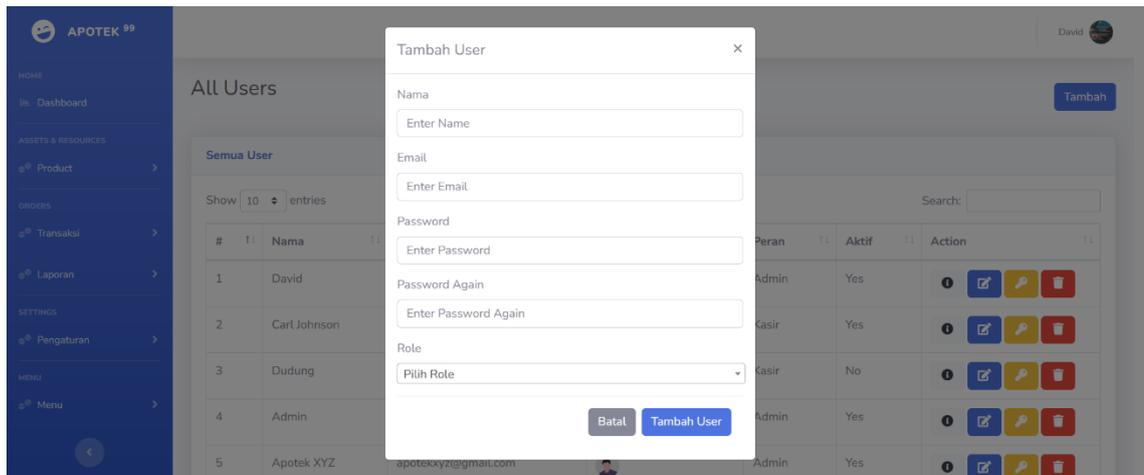
Berikut merupakan tampilan dari melihat informasi user yang ditampilkan dalam bentuk jendela *pop-up*. Dapat dilihat pada Gambar 4.99.



Gambar 4. 99 Tampilan Rincian User

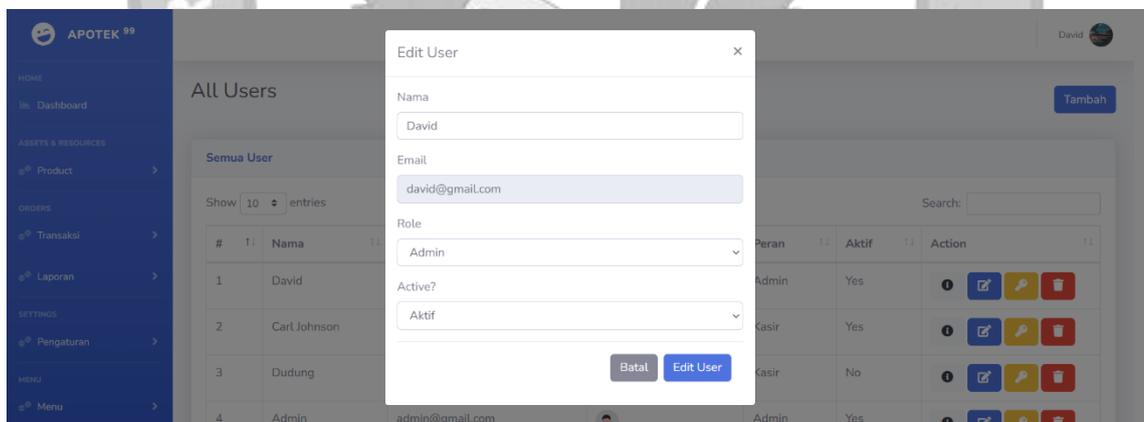
Berikut merupakan tampilan dari *add new user* yang ditampilkan dalam bentuk jendela *pop-up*. Jika admin ingin menambahkan data user, maka admin harus memasukkan data yang lengkap, dan

menentukan role, sesuai dengan yang diinginkan ke dalam *form add new user*. Dapat dilihat pada Gambar 4.100.



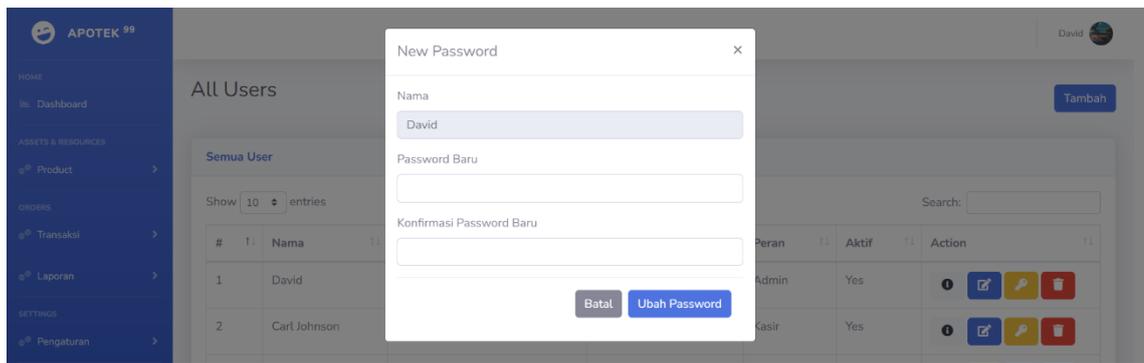
Gambar 4. 100 Tampilan Tambah User

Berikut merupakan tampilan dari *edit data user* yang ditampilkan dalam bentuk jendela *pop-up*. Jika admin ingin mengubah data user, maka admin dapat mengubah sesuai keinginan, sesuai dengan yang diinginkan ke dalam *form edit user*. Dapat dilihat pada Gambar 4.101.



Gambar 4. 101 Tampilan Edit User

Berikut merupakan tampilan dari *change password user* yang ditampilkan dalam bentuk jendela *pop-up*. Jika admin ingin mengubah *password* user, maka admin dapat mengubah *password* sesuai keinginan, sesuai dengan yang diinginkan ke dalam *form new password*. Dapat dilihat pada Gambar 4.102.



Gambar 4. 102 Tampilan Input Password Baru untuk *User*

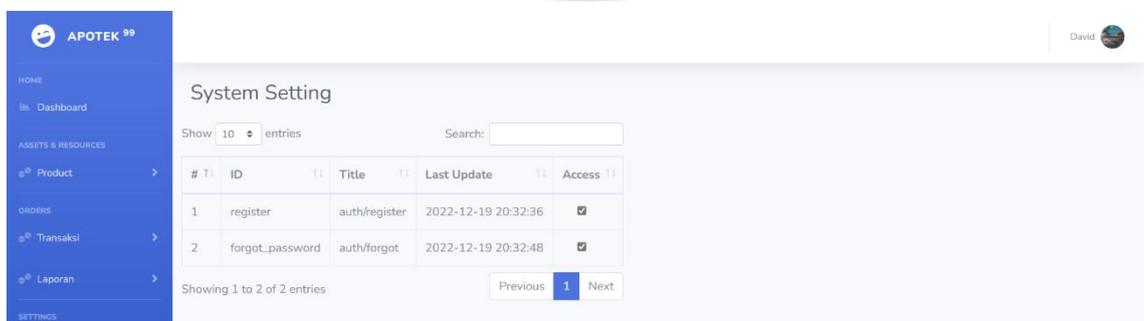
Selain dapat melihat data user, melakukan penambahan user, melakukan edit user, dan melakukan edit *password*, pada menu ini, admin juga dapat melakukan penghapusan user jika diperlukan. Jika admin ingin menghapus user, maka sistem akan menampilkan pesan konfirmasi penghapusan seperti yang dapat dilihat pada Gambar 4.103.



Gambar 4. 103 Tampilan Hapus Akun *User*

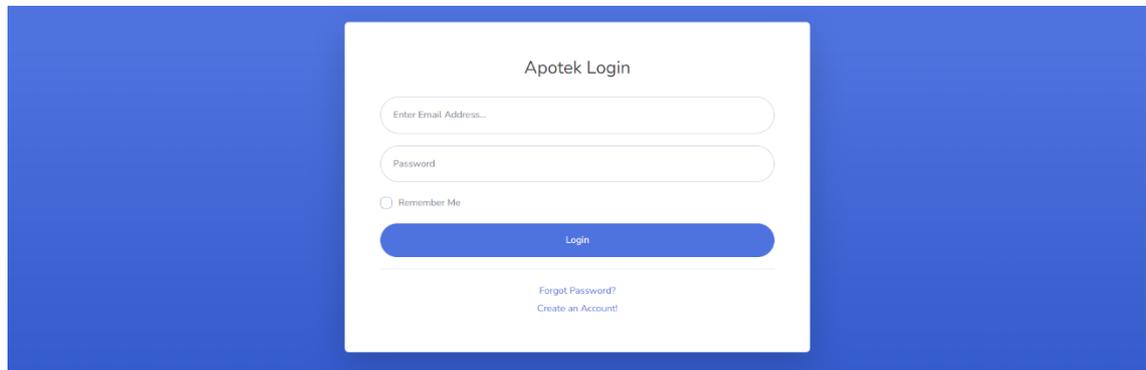
15) Kelola *Setting*

Berikut merupakan tampilan dari menu *setting*, untuk pengelolaan fitur halaman *login*. Pada menu, admin dapat melihat dua fitur pada halaman *login*, yaitu *register* dan *forgot password*, yang terdapat pada Gambar 4.104.



Gambar 4. 104 Tampilan *System Setting*

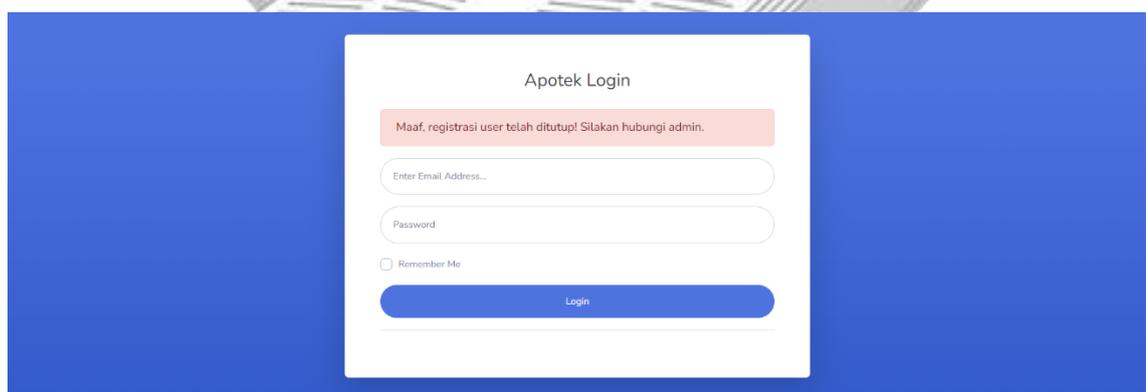
Jika admin ingin mengaktifkan fitur pada halaman *login*, maka admin harus memberi centang pada baris fitur yang akan diaktifkan. Sistem akan menampilkan pesan sukses, dan mengaktifkan akses fitur pada halaman *login*. Halaman *login* akan menampilkan nama fitur dibawah tombol *login*. Terdapat pada Gambar 4.105.



The image shows a login form titled "Apotek Login" on a white background with a blue border. It contains the following elements: a title "Apotek Login", an input field labeled "Enter Email Address...", another input field labeled "Password", a checkbox labeled "Remember Me", a blue "Login" button, and two links below the button: "Forgot Password?" and "Create an Account!".

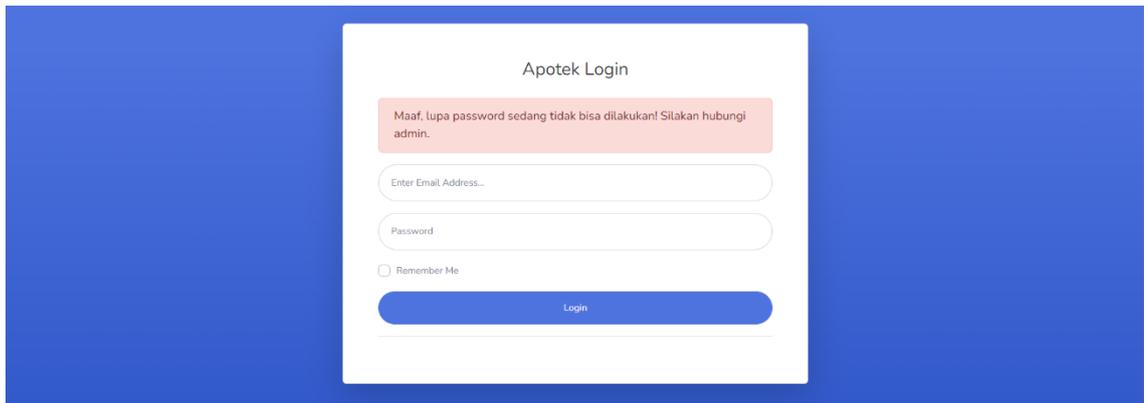
Gambar 4. 105 Tampilan *Login* Saat Fitur Aktif

Jika admin ingin mematikan fitur pada halaman *login*, maka admin harus melepas centang pada baris fitur yang akan dinonaktifkan. Sistem akan menampilkan pesan sukses, dan menonaktifkan akses fitur pada halaman *login*. Halaman *login* tidak akan nama fitur dibawah tombol *login*. Jika *user* membuka fitur dengan *url*, maka sistem akan memberi pesan *error*, dan melakukan *redirect* ke halaman *Login*. Terdapat pada Gambar 4.106 dan Gambar 4.107.



The image shows the same "Apotek Login" form as in the previous screenshot, but with a red error message at the top: "Maaf, registrasi user telah ditutup! Silakan hubungi admin." The form fields and "Login" button are still present below the message.

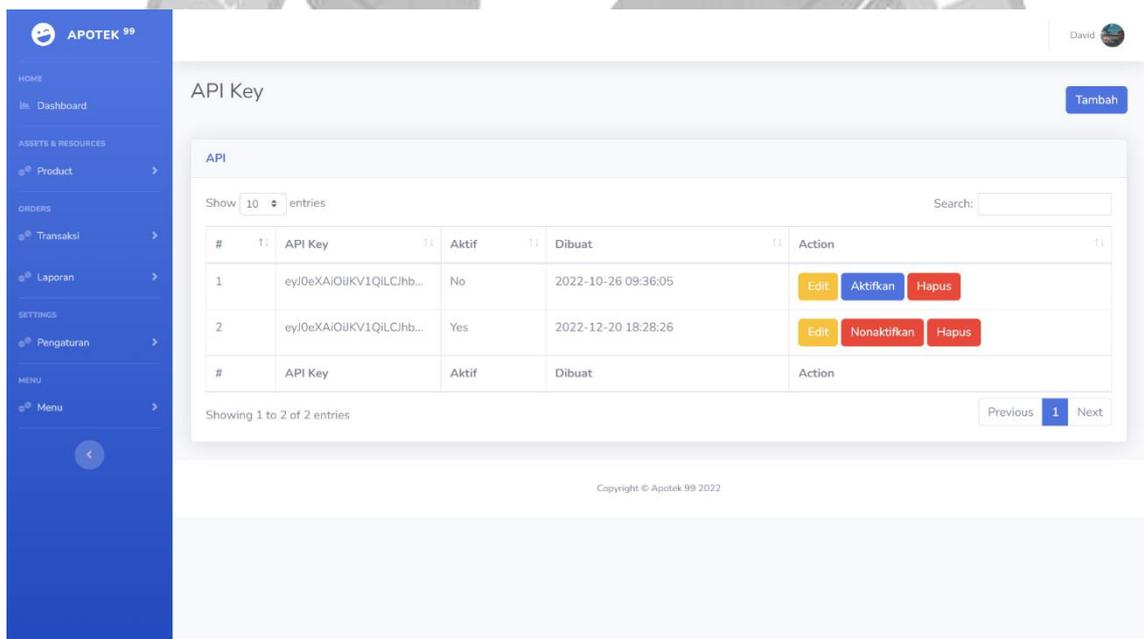
Gambar 4. 106 Tampilan Fitur Registrasi saat Dinonaktifkan Admin



Gambar 4. 107 Tampilan Fitur Lupa *Password* saat Dinonaktifkan Admin

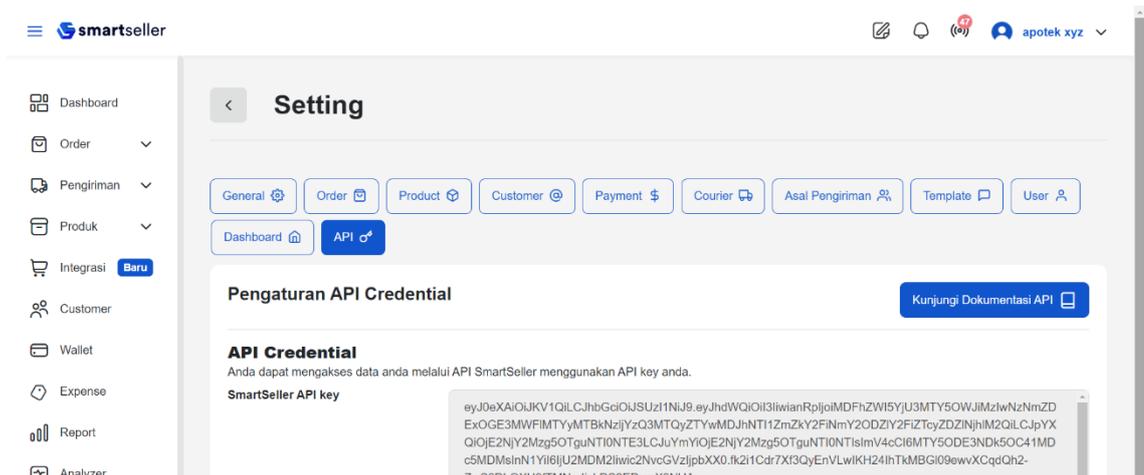
16) Kelola API

Berikut merupakan tampilan dari menu *API Key*, sebagai penghubung sistem ke SmartSeller. Yang dapat mengakses halaman ini adalah admin. Pada menu ini, admin dapat melihat master data *API Key*, yang terdapat pada Gambar 4.108.



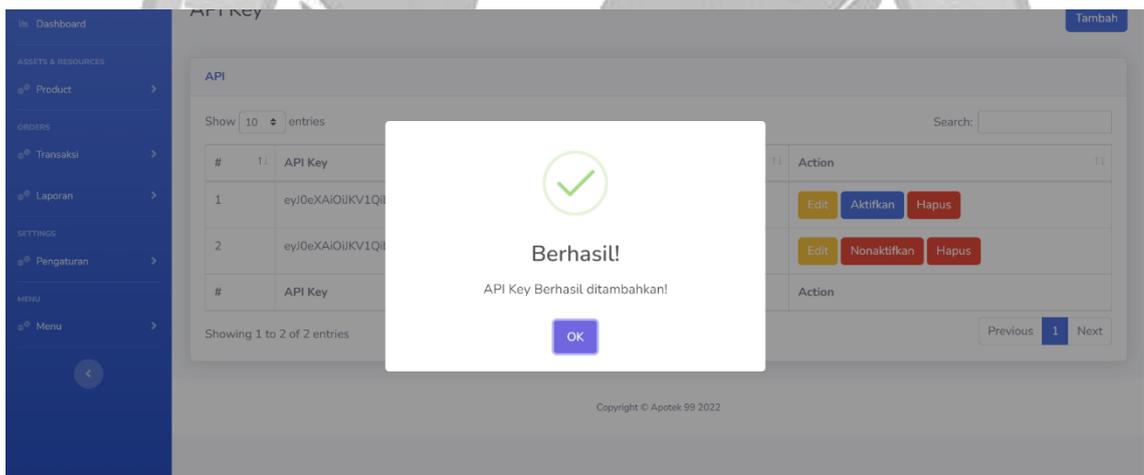
Gambar 4. 108 Tampilan Kelola *API Key*

Admin juga dapat melakukan penambahan *API Key* dengan mengisi form yang disediakan oleh sistem. Sebagai referensi, admin harus membuka dan *login* ke SmartSeller, untuk mendapatkan *API Key*. Terdapat pada Gambar 4.109.



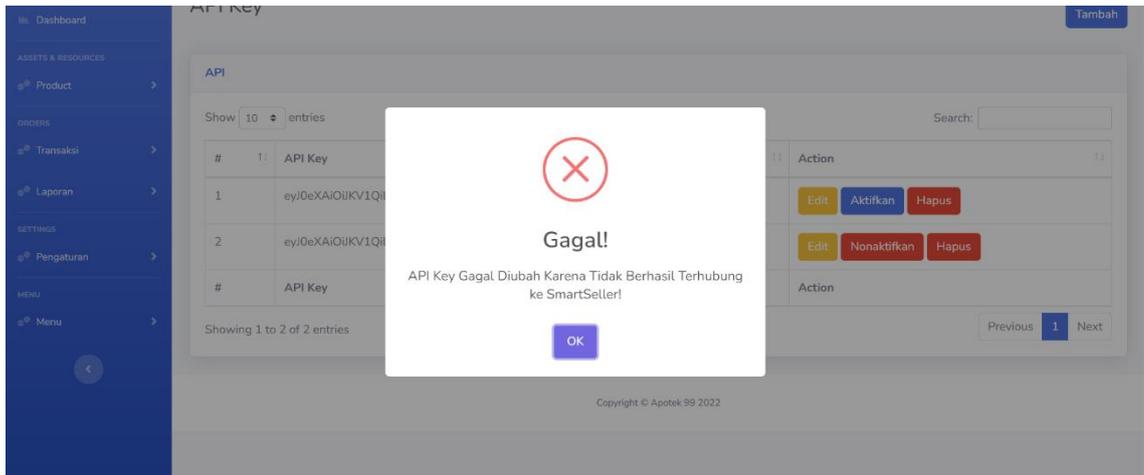
Gambar 4. 109 Referensi SmartSeller untuk Mendapatkan *API Key*

Jika data yang dimasukkan menghasilkan pesan sukses dari API, maka sistem akan menampilkan jendela *pop-up* yang menyatakan bahwa data tersebut telah berhasil ditambahkan ke dalam *database*. Terdapat pada gambar 4.110.



Gambar 4. 110 Tampilan *API Key* Berhasil Ditambahkan

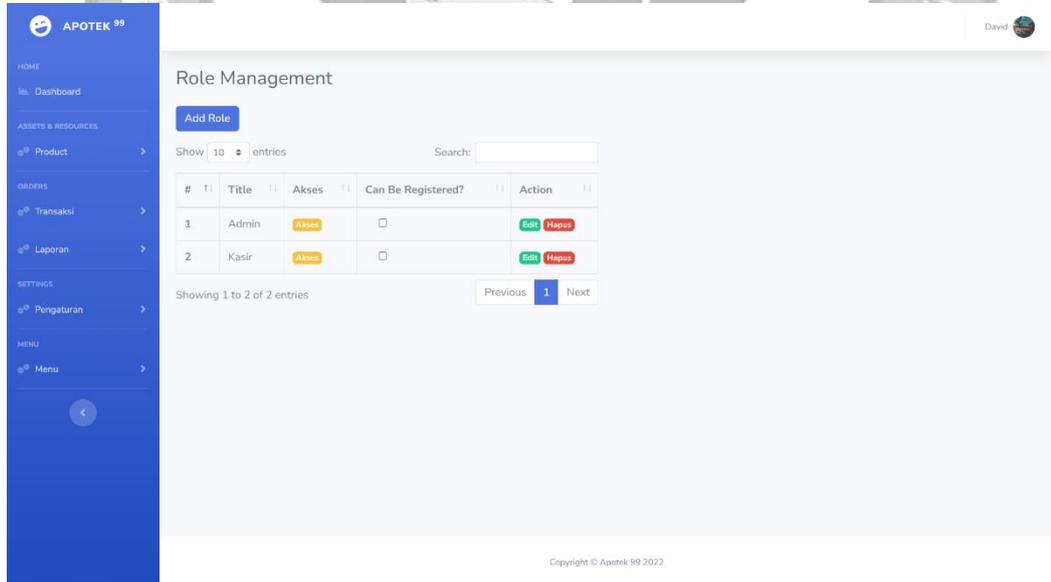
Admin juga dapat melakukan pengeditan *API Key* dengan mengisi form yang disediakan oleh sistem. Jika data yang dimasukkan menghasilkan pesan gagal dari API, maka akan ada pesan error yang dikirimkan melalui form tersebut. Hal ini juga berlaku untuk menambahkan dan mengaktifkan *API Key*. Terdapat pada Gambar 4.111.



Gambar 4. 111 Tampilan Pesan *API Key* Gagal Diedit

17) Kelola *Role* dan Akses

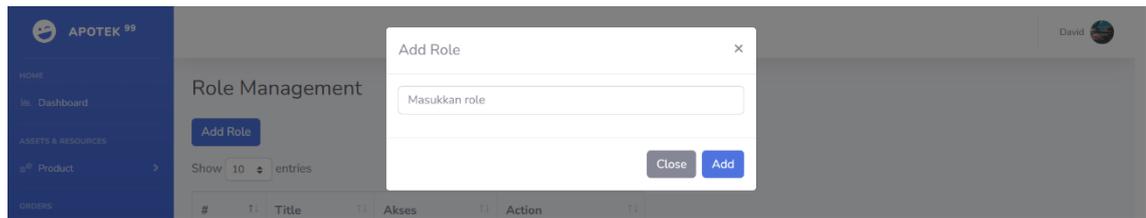
Berikut merupakan tampilan dari pengelolaan *role* pada Gambar 4.112. Menu ini hanya dapat diakses oleh admin. Pada menu ini, admin dapat melihat data seluruh *role* yang terdapat dalam sistem. Admin dapat melakukan penambahan *role*, melakukan *edit role*, atau *delete role*. Selain itu, admin dapat mengelola akses menu atau submenu terhadap *role*.



Gambar 4. 112 Tampilan *Role Management*

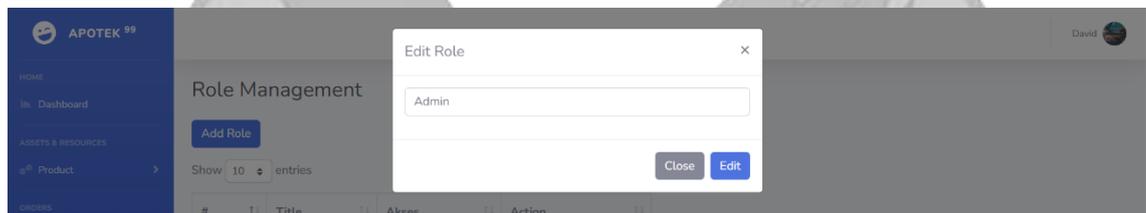
Admin dapat menambahkan *role*, dengan menekan tombol Tambah. Sistem akan menampilkan jendela *pop-up* untuk mengisi

form penambahan *role*. Admin mengisi nama *role*, dan menekan simpan. Dapat dilihat pada Gambar 4.113.



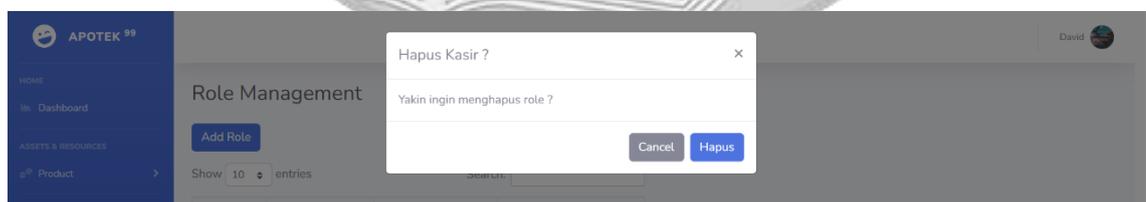
Gambar 4. 113 Tampilan Penambahan Data *Role*

Admin dapat melakukan pengeditan data *role*, dengan menekan tombol edit pada baris *role* yang akan diubah. Sistem akan menampilkan jendela *pop-up form* pengeditan *role*. Admin mengganti data *role*, dan menekan tombol edit. Dapat dilihat pada Gambar 4.114.



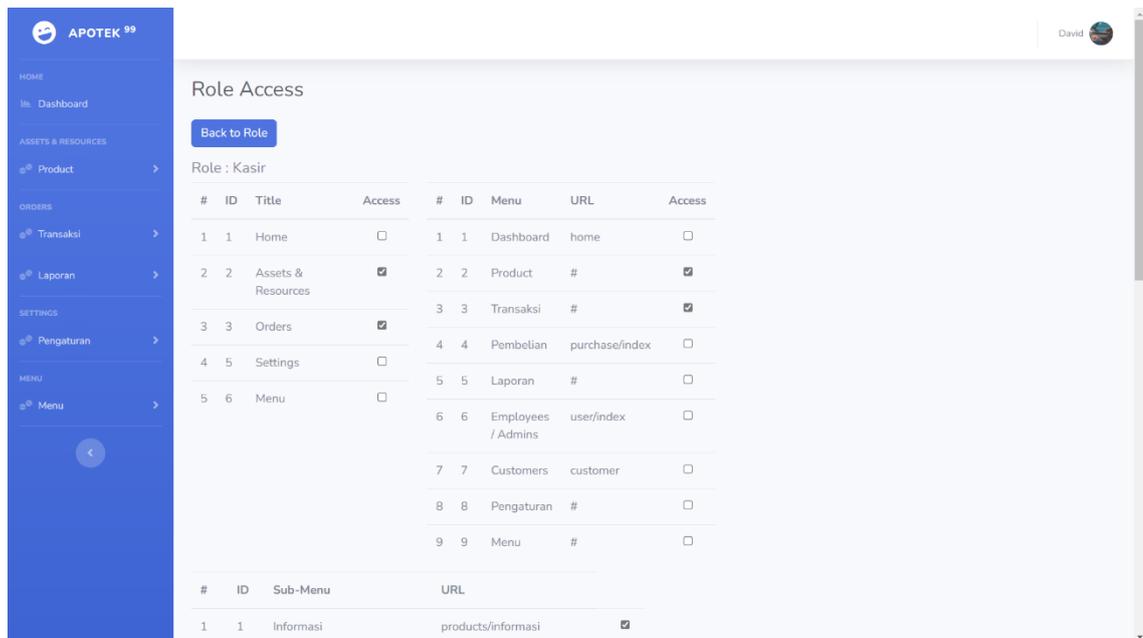
Gambar 4. 114 Tampilan Pengeditan Data *Role*

Admin dapat melakukan penghapusan data *role*, dengan menekan tombol hapus pada baris *role* yang akan diubah. Sistem akan menampilkan jendela *pop-up* konfirmasi penghapusan *role*. Jika dilanjutkan, maka sistem akan menghapus *role*. Dapat dilihat pada Gambar 4.115.



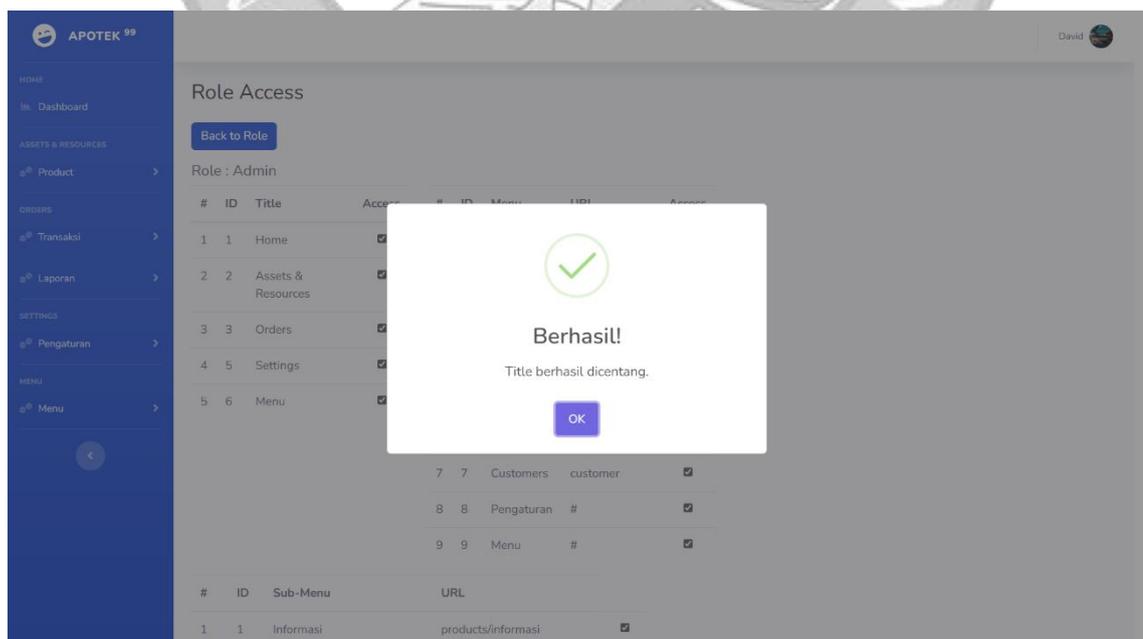
Gambar 4. 115 Tampilan Hapus *Role*

Setelah menambahkan, mengedit, dan menghapus *role*, admin dapat mengelola seluruh akses dari *role* tersebut. Dapat dilihat pada Gambar 4.116, dimana admin sedang mengelola akses *role* terhadap kasir.



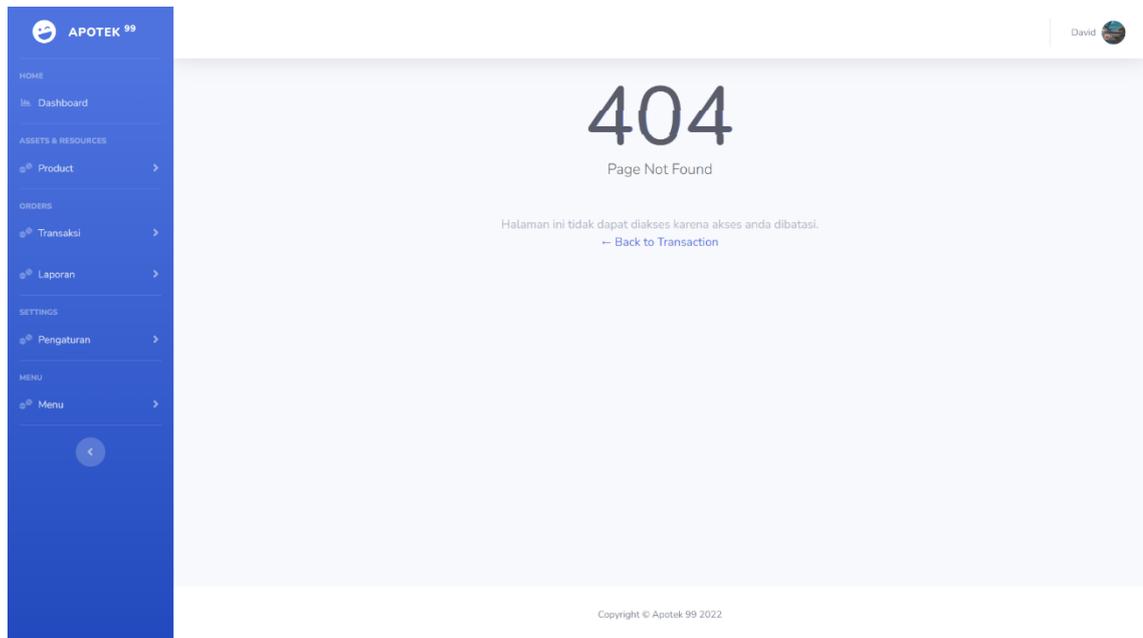
Gambar 4. 116 Tampilan Role Access

Pada menu *role access*, terdapat 3 tabel yang ditampilkan, yakni *title*, *menu*, dan *submenu*. Untuk memberikan akses kepada *role*, admin dapat mencentang baris tabel yang akan ditambah akses ke *role* yang tercantum pada halaman. Sistem akan menampilkan pesan sukses, dan *role* tersebut dapat melihat menu tersebut. Pemberian akses *title* akan menambah menu di sebelah kiri, untuk ditampilkan. Dapat dilihat pada Gambar 4.117.



Gambar 4. 117 Tampilan Pesan Sukses Penambahan Title pada Role

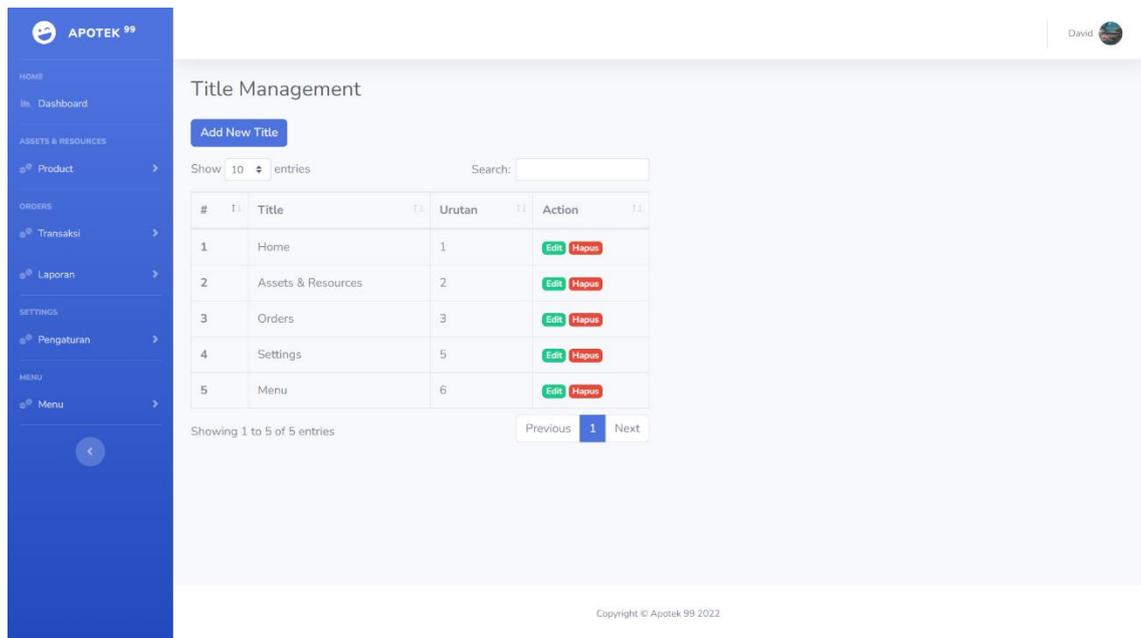
Perubahan centang ini akan bergantung terhadap akses menu dan *submenu* setiap *role*. Setiap *user* membuka halaman sistem, maka sistem akan memeriksa akses *role* pada menu atau submenu yang dibuka. Jika akses *role* terhadap menu atau submenu tidak ditemukan dalam *database*, maka sistem akan melakukan *redirect* ke halaman *blocked*. Dapat dilihat pada Gambar 4.118.



Gambar 4. 118 Tampilan *Blocked* jika *User* Tidak Mendapat Akses

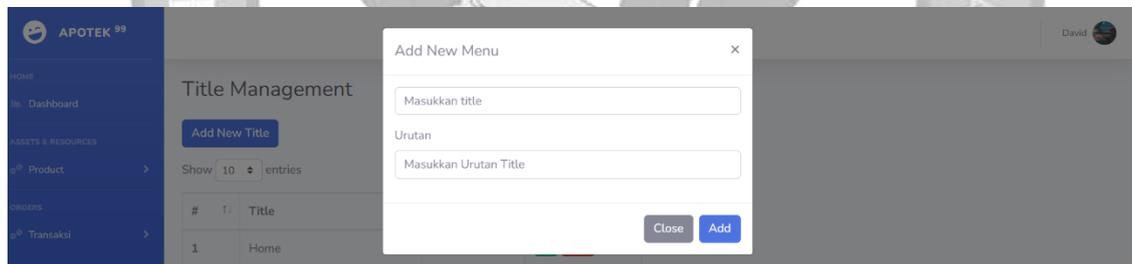
18) Kelola *Title*

Berikut merupakan tampilan dari pengelolaan *title* Gambar 4.119. Menu ini hanya dapat diakses oleh admin. Pada menu ini, admin dapat melihat data dari berbagai macam *title* yang terdapat dalam sistem, dan juga beberapa data seperti nama *title*, dan urutan *title*. Admin juga dapat melakukan penambahan *title* jika diperlukan, dan juga melakukan *edit* data *title*, atau *delete*. *Title* akan terlihat dengan tulisan kecil pada navigasi menu, di paling atas.



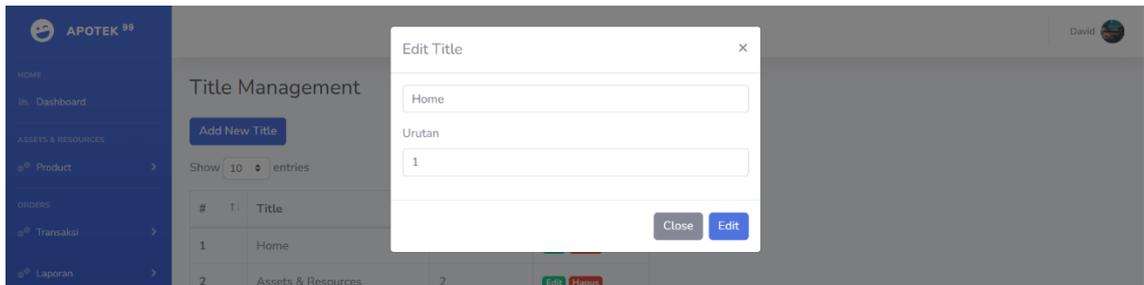
Gambar 4. 119 Tampilan *Title Management*

Jika admin ingin menambahkan data *title*, maka admin dapat menekan tombol *Add New Title*. Sistem akan menampilkan jendela *pop-up* untuk menambahkan *title*. Admin mengisi data *title*, dan menekan tombol *Add*. Dapat dilihat pada Gambar 4.120.



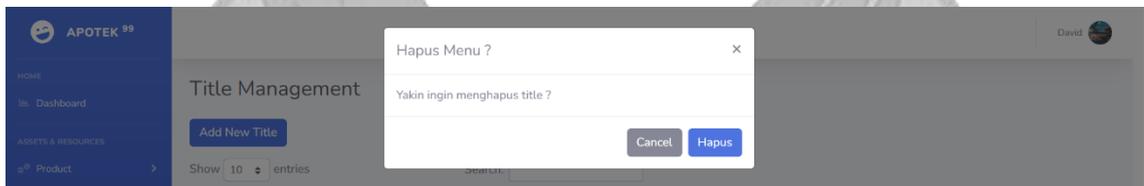
Gambar 4. 120 Tampilan Tambah *Title* Baru

Selain menambahkan, admin dapat mengubah data *title*, yang terdapat pada Gambar 4.121. Admin tombol *Edit*. Sistem akan menampilkan jendela *pop-up* untuk mengubah *title*. Admin mengisi data *title*, dan menekan tombol *Edit*.



Gambar 4. 121 Tampilan Edit *Title*

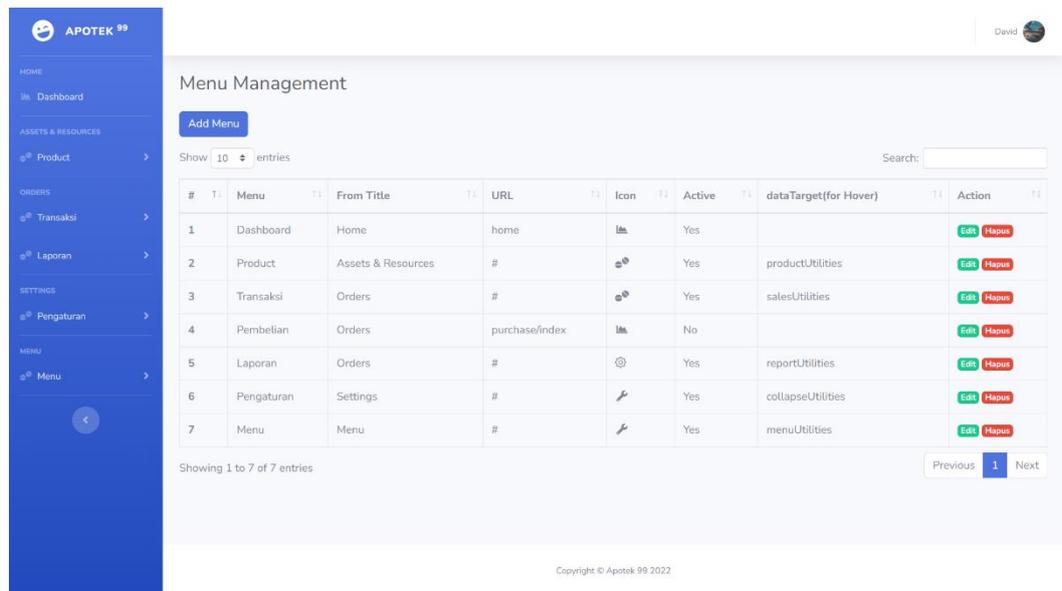
Setelah itu, apabila diperlukan, admin dapat menghapus data *title*, dengan menekan tombol Hapus. Sistem akan membuka halaman *pop-up* untuk mengonfirmasi penghapusan data *title*. Jika dilanjutkan, maka sistem akan menghapus *title*, termasuk pada *title* di navigasi menu. Dapat dilihat pada Gambar 4.122.



Gambar 4. 122 Tampilan Hapus *Title*

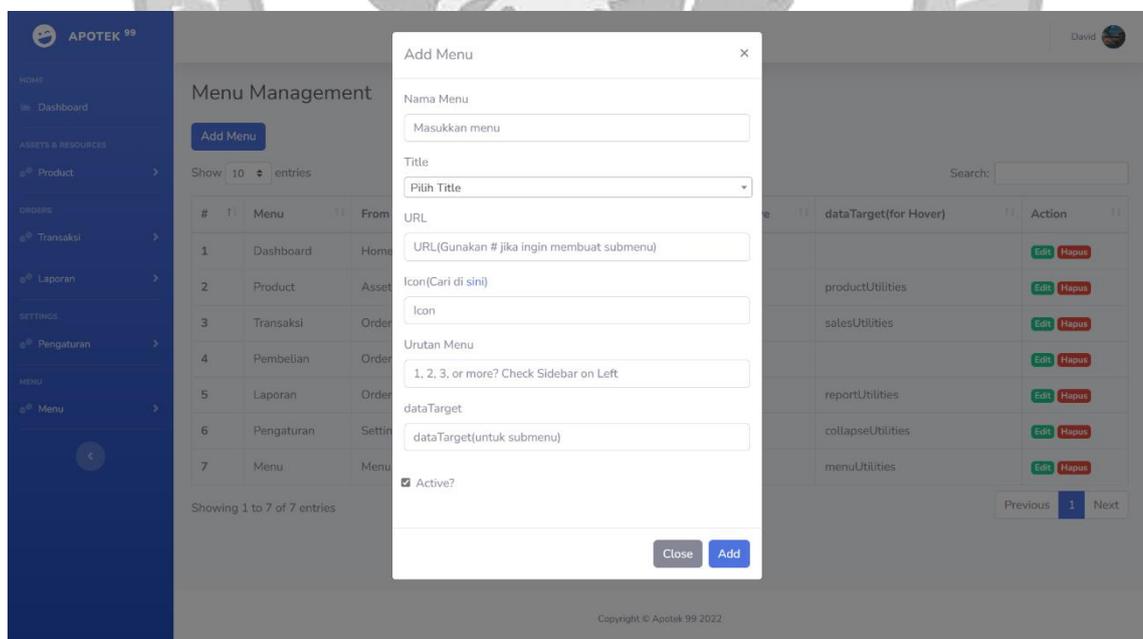
19) Kelola Menu

Berikut merupakan tampilan dari pengelolaan menu Gambar 4.123. Menu ini hanya bisa diakses oleh admin. Pada menu ini, dapat melihat data dari berbagai macam menu yang terdapat dalam sistem, dan juga beberapa data seperti nama *title*, *url*, icon, urutan *submenu*, *dataTarget* (untuk menu *hover*), dan status aktif. Admin juga dapat melakukan penambahan *menu* jika diperlukan, dan juga melakukan *edit* data *menu*, atau *delete*.



Gambar 4. 123 Tampilan Menu Management

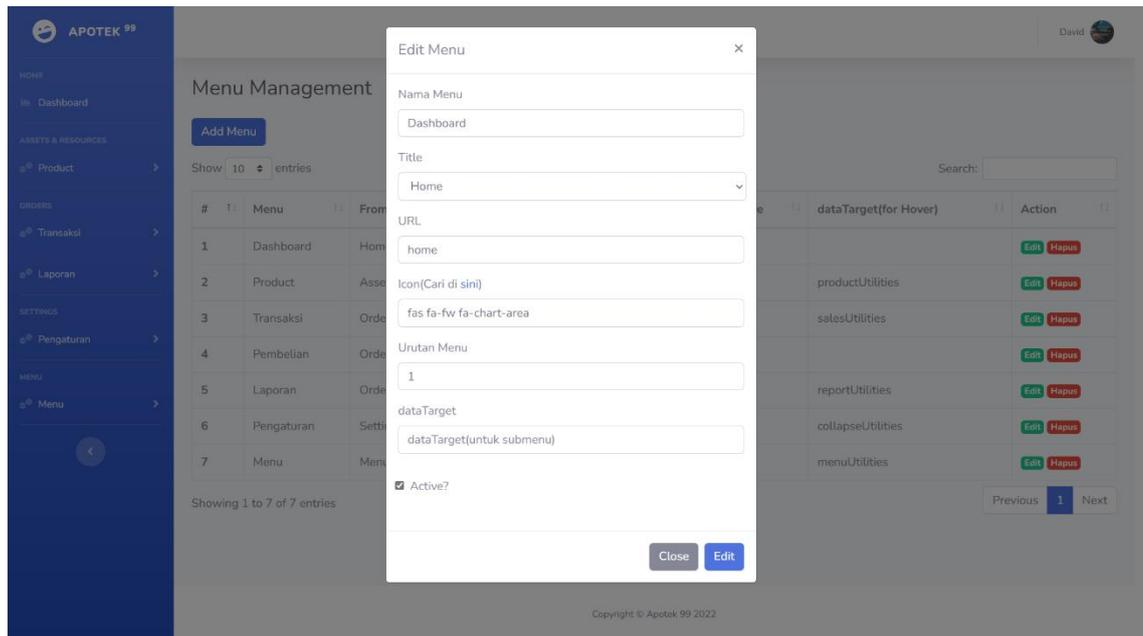
Jika admin ingin menambahkan data menu, maka admin dapat menekan tombol *Add New Menu*. Sistem akan menampilkan jendela *pop-up* untuk menambahkan *title*. Admin mengisi data menu, dan menekan tombol *Add*. Dapat dilihat pada Gambar 4.124.



Gambar 4. 124 Tampilan Penambahan Menu

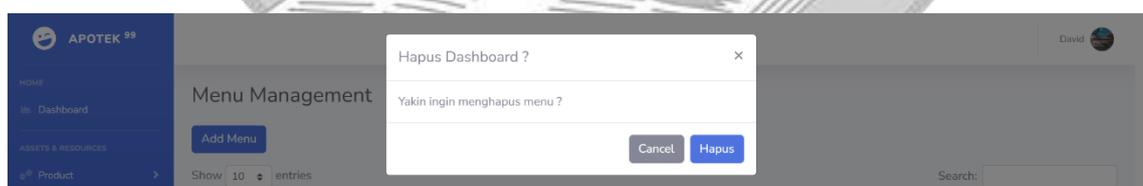
Selain menambahkan, admin dapat mengubah data menu, yang terdapat pada Gambar 4.125. Admin dapat menekan tombol

Edit. Sistem akan menampilkan jendela *pop-up* untuk mengubah menu. Admin mengisi data menu, dan menekan tombol *Edit*.



Gambar 4. 125 Tampilan Edit Menu

Setelah itu, apabila diperlukan, admin dapat menghapus data menu, dengan menekan tombol Hapus. Sistem akan membuka halaman *pop-up* untuk mengonfirmasi penghapusan data menu. Jika dilanjutkan, maka sistem akan menghapus menu, sehingga menu dihilangkan pada navigasi menu. Tampilan terdapat pada Gambar 4.126.



Gambar 4. 126 Tampilan Hapus Menu

20) Kelola Submenu

Berikut merupakan tampilan dari pengelolaan *submenu* Gambar 4.127. Menu ini hanya dapat diakses oleh admin. Pada menu ini, admin dapat melihat data dari berbagai macam *submenu* yang terdapat dalam sistem, dan juga beberapa data seperti nama menu, *url*, urutan *submenu*, dan status aktif. Admin juga dapat melakukan penambahan

submenu jika diperlukan, dan juga melakukan *edit* data *submenu*, atau *delete*.

#	Sub-Menu	From Menu	URL	Urutan	Active	Action
1	Informasi	Product	products/informasi	1	Yes	Edit Hapus
2	Kategori Produk	Product	products/category	3	Yes	Edit Hapus
3	Unit Produk	Product	products/unit	4	Yes	Edit Hapus
4	Gambar Produk	Product	products/image	4	No	Edit Hapus
5	Batch Kadaluwarsa	Product	products/expiredstock	5	Yes	Edit Hapus
6	Produk Kosong	Product	products/emptystock	6	Yes	Edit Hapus
7	Cek Order Marketplace	Transaksi	transaction/marketplace	1	No	Edit Hapus
8	Tambah Penjualan Toko	Transaksi	transaction/create	1	Yes	Edit Hapus
9	Laba Jual Beli	Laporan	report/laba	1	Yes	Edit Hapus
10	Penjualan Marketplace	Laporan	report/marketplace	2	No	Edit Hapus

Gambar 4. 127 Tampilan *Submenu Management*

Jika admin ingin menambahkan data *submenu*, maka admin dapat menekan tombol *Add New Submenu*. Sistem akan menampilkan jendela *pop-up* untuk menambahkan *submenu*. Admin mengisi data *submenu*, dan menekan tombol *Add*. Dapat dilihat pada Gambar 4.128.

Add New Sub Menu

Masukkan sub-menu

Select Menu

Masukkan URL

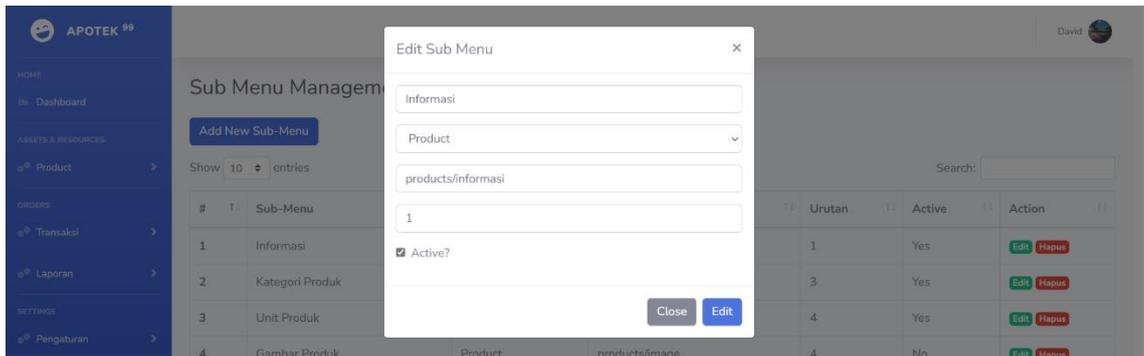
Masukkan Nomor Urut

Active?

Close Add

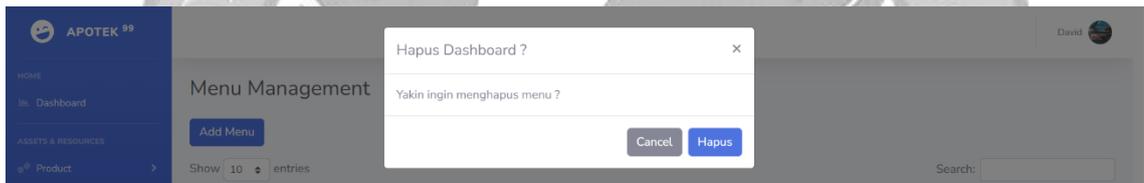
Gambar 4. 128 Tampilan Tambah *Submenu*

Selain menambahkan, *user* dapat mengubah data *submenu*, yang terdapat pada Gambar 4.129. Admin dapat menekan tombol *Edit*. Sistem akan menampilkan jendela *pop-up* untuk mengubah *submenu*. *User* mengisi data *submenu*, dan menekan tombol *Edit*.



Gambar 4. 129 Tampilan Edit *Submenu*

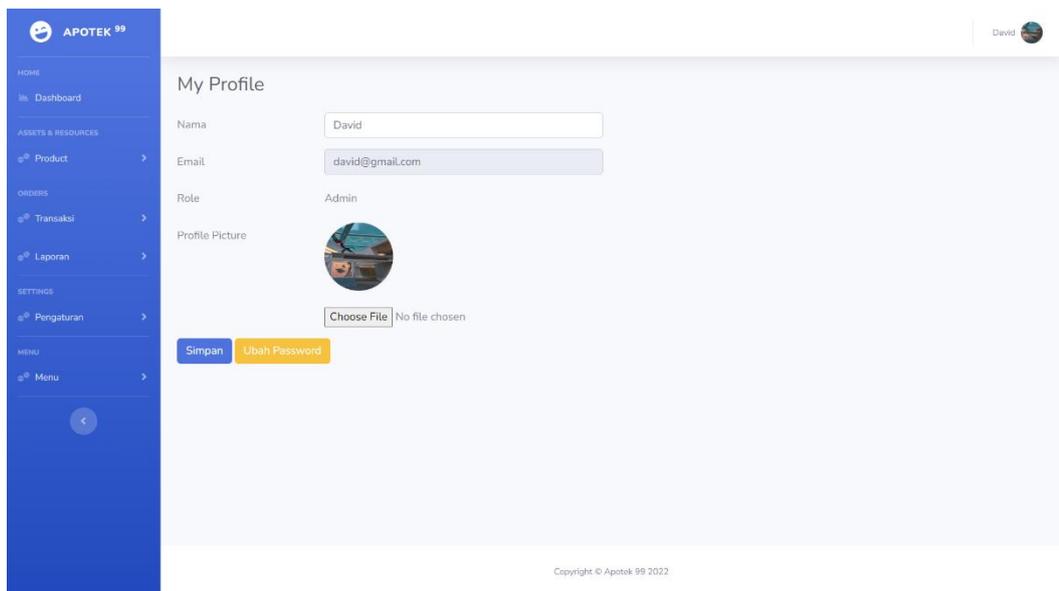
Setelah itu, apabila diperlukan, admin dapat menghapus data *submenu*, dengan mengklik tombol Hapus. Sistem akan membuka halaman *pop-up* untuk mengonfirmasi penghapusan data *submenu*. Jika dilanjutkan, maka sistem akan menghapus *submenu*, sehingga submenu dihilangkan pada navigasi menu. Terdapat pada Gambar 4.130.



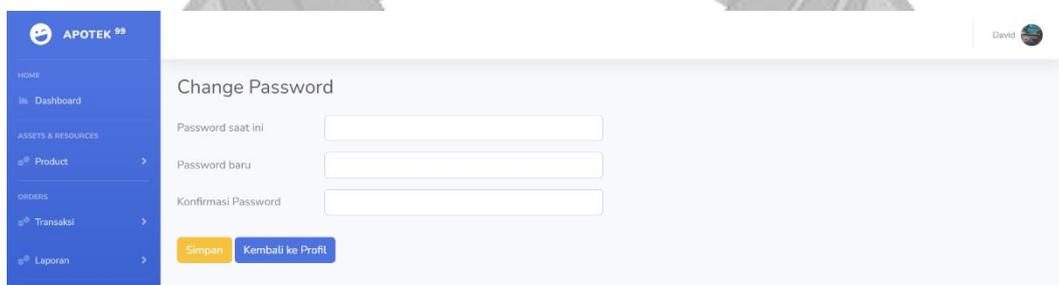
Gambar 4. 130 Tampilan Hapus Menu

21) Kelola Profil

Berikut merupakan tampilan dari pengelolaan profil. Fitur ini dapat diakses oleh seluruh *user*. Dalam menu ini, *user* dapat melihat data dari akun pribadinya, seperti nama, email, dan *role*. *User* dapat mengubah nama, *password*, dan *profile picture*, yang terdapat pada Gambar 4.131 dan Gambar 4.132.



Gambar 4. 131 Halaman *My Profile*



Gambar 4. 132 Halaman *Change Password*

22) Activity Log

Berikut merupakan tampilan dari *activity log* Gambar 4.133. Menu ini hanya dapat diakses oleh admin. *User* dapat melihat *history login* yang dilakukan, setiap *user* yang melakukan *login* ke dalam sistem. Pada menu ini, admin dapat melihat *email* yang digunakan saat masuk ke sistem, waktu *login*, dan status *login*.

#	Email	Date	Success
1	david@gmail.com	2022-10-06 09:57:10	Berhasil
2	david@gmail.com	2022-10-06 10:01:22	Berhasil
3	david@gmail.com	2022-10-06 10:08:24	Berhasil
4	david@gmail.com	2022-10-06 10:10:04	Berhasil
5	david@gmail.com	2022-10-06 10:24:49	Berhasil
6	cj@gmail.com	2022-10-06 11:02:07	Berhasil
7	david@gmail.com	2022-10-06 11:05:47	Berhasil
8	cj@gmail.com	2022-10-06 11:17:13	Berhasil
9	david@gmail.com	2022-10-06 11:19:17	Berhasil
10	david@gmail.com	2022-10-06 12:37:45	Berhasil

Gambar 4. 133 Tampilan Activity Log

4.5 Pengujian Sistem Usulan

Dalam tahap pengujian sistem usulan ini, metode pengujian sistem yang digunakan adalah *black box testing*. Melalui metode ini, sistem akan diuji dari tampilan luar (*interface*) seperti tampilan, fungsi, alur sistem, dan proses bisnis. Dalam tahap pengujian ini, *test case* yang diujikan disusun ke dalam tabel untuk mempermudah pengujian. Tahap pengujian dilampirkan pada Lampiran D. Berikut merupakan salah satu tabel pada Lampiran D, yaitu tabel 4.4 untuk fitur mengelola informasi produk.

Tabel 4. 23 Test Case Fitur Mengelola Informasi Produk oleh Admin sebagai Aktor

<i>Step Description</i>	<i>Test Steps</i>	<i>Expected Results</i>	<i>Result (Pass/Fail)</i>
Mengakses informasi produk	Mengakses submenu "Informasi" pada menu "Product"	Informasi nama produk, berat produk, kategori produk, dan unit produk muncul.	Pass
		Dapat melihat rincian informasi produk pada baris tabel masing-masing	Pass
		Dapat mengakses "Batch Produk" dengan daftar <i>batch</i> produk dengan stok tersedia	Pass

Tabel 4. 23 *Test Case* Fitur Mengelola Informasi Produk oleh Admin sebagai Aktor (Lanjutan)

Mengelola informasi produk	Menambahkan informasi produk baru dengan data nama produk yang sudah ada	Muncul pesan error: - "Mohon untuk tidak menginput nama produk yang telah tersedia"	<i>Pass</i>
----------------------------	--	--	-------------



BAB V

KESIMPULAN DAN SARAN

5.1 Kesimpulan

Dari hasil penelitian yang telah dilaksanakan terhadap Apotek 99, sistem dapat mempersingkat waktu dalam memperbarui dan menyamai stok produk pada konvensional dan *marketplace*, dalam waktu 40 detik. Fitur transaksi toko yang berada pada sistem lama, telah dikembangkan dengan mengintegrasikannya dengan *API Smartseller*, sehingga stok produk pada konvensional dan *marketplace* akan diperbarui berdasarkan penambahan transaksi konvensional. Dengan ini, apotek dapat memperoleh informasi stok produk yang *up-to-date*, dan meminimalisir kerugian dari terjadinya transaksi penjualan *overselling*.

Selain itu, fitur pengelompokan produk berhasil mempermudah pemilik dalam melacak dan menelusuri histori produk, sehingga apotek dapat meminimalisir kerugian atas barang *expired*, dengan segera meretur dan menggantinya dengan yang terbaru.

5.2 Saran

Sistem *Web API Integrator* yang dirancang mempunyai peluang untuk dapat dikembangkan lebih lanjut. Berikut saran yang dapat diberikan dari pengembangan sistem usulan ini:

- 1) Menyempurnakan proses sinkronisasi produk, agar sistem lebih efisien dan cepat saat dioperasikan.
- 2) Menambahkan fitur *discount* pada produk.
- 3) Menambahkan fitur cetak struk belanja pada transaksi apotek.
- 4) Menambahkan fitur pengiriman struk transaksi apotek melalui email.
- 5) Menyediakan sistem pembayaran dengan e-wallet untuk transaksi toko, seperti OVO, DANA, Go-Pay, dan sejenisnya.

- 6) Mendaftarkan apotek menjadi *official store* atau sejenisnya, agar sistem dapat berinteraksi langsung dengan *API* bawaan *marketplace*.
- 7) Menambahkan fitur refund *batch* untuk melacak produk *expired* yang dikembalikan pada pedagang besar farmasi (PBF).
- 8) Mengkonversi penulisan harga produk dengan tanda titik dan diberikan label Rupiah (Rp).
- 9) Mengkonversi penulisan format tanggal dari "tahun-bulan-tanggal" (Contoh: 2023-01-01) menjadi "tanggal bulan tahun" (Contoh: 01 Januari 2023).



DAFTAR PUSTAKA

- [1] "NGOBROL@TEMPO - Mendorong Transformasi Digital UMKM Melalui E-Commerce," Tempo, 13 August 2021. [Online]. Available: <https://www.youtube.com/watch?v=wYKyEdXxOro&t=845s>. [Accessed 5 December 2021].
- [2] A. A. S. Ani Oktarini Sari, Web Programing, Yogyakarta: Graha Ilmu, 2019.
- [3] K. Lane, "Intro to APIs: What Is an API?," Postman, 5 October 2020. [Online]. Available: <https://blog.postman.com/intro-to-apis-what-is-an-api/>. [Accessed 25 November 2021].
- [4] S. Anhar, Panduan Menguasai PHP & MySQL secara otodidak, Jakarta Selatan: PT TransMedia Pustaka, 2010.
- [5] A. H. I. Y. Supardi, Semua Bisa Menjadi Programmer CodeIgniter Basic, Jakarta: Elex Media Komputindo, 2018.
- [6] H. Alatas, RESPONSIVE WEB DESIGN dengan PHP & BOOTSTRAP, Yogyakarta: Penerbit Lokomedia, 2013.
- [7] A. Y. S. D. Apriadi, "E-Commerce Berbasis Marketplace Dalam Upaya Mempersingkat Distribusi Penjualan Hasil Pertanian," *Jurnal Rekayasa Sistem dan Teknologi Informasi*, vol. 1, no. 2, pp. 131-136, 2017.
- [8] Tokopedia, "Tentang Tokopedia," Tokopedia, [Online]. Available: <https://www.tokopedia.com/about/>. [Accessed 24 November 2021].
- [9] d. H. Rifalika, "Pengaruh Pandemi Virus Corona Pada Pengunjung Tokopedia," *Institut Teknologi dan Bisnis Stikom Bali Jimbaran, Fakultas Teknologi Informasi*, 2021.
- [10] Tokopedia, "Terms & Conditions," Tokopedia, [Online]. Available: <https://www.tokopedia.com/terms>. [Accessed 17 December 2021].
- [11] Shopee, "Tentang Shopee - Karir | Shopee Indonesia," Shopee, [Online]. Available: <https://careers.shopee.co.id/about>. [Accessed 24 November 2021].
- [12] "Peta E-Commerce Indonesia Q2 2021," iPrice, [Online]. Available: <https://iprice.co.id/insights/mapofecommerce/>. [Accessed 17 December 2021].

- [13] Shopee, "Kebijakan Barang Yang Dilarang Dan Dibatasi," Shopee, 1 November 2021. [Online]. Available: <https://help.shopee.co.id/portal/article/71189-Kebijakan-Barang-yang-dilarang-dan-dibatasi>. [Accessed 17 December 2021].
- [14] SmartSeller, "Tentang Kami | SmartSeller," SmartSeller, [Online]. Available: <https://smartseller.co.id/about-us>. [Accessed 20 December 2022].
- [15] SmartSeller, "SmartSeller API," SmartSeller, [Online]. Available: <https://api.smartseller.co.id/doc/index.html#introduction>. [Accessed 12 October 2022].
- [16] A. Solichin, Pemrograman Web dengan PHP dan MYSQL, Jakarta Selatan: Penerbit Budi Luhur, 2016.
- [17] S. R. Fitri, Pemrograman Basis Data Menggunakan MySQL, Banjarmasin: Deepublish, 2020.
- [18] A. Dennis, B. H. Wixom and R. M. Roth, System Analysis & Design with UML, 5th ed., Singapore: John Wiley & Sons, Inc., 2015.
- [19] D. Y. P. Kamariah, "SISTEM INFORMASI TRANSAKSI PADA APOTEK KAB. INHIL RIAU BERBASIS WEB," *Jurnal SISTEMASI*, vol. 5, no. 2, pp. 11-18, 2016.
- [20] Y. S. D. A. A. Abdul Muiz Hasyim, "Rancang Bangun Sistem Informasi Manajemen Gudang Apotek Menggunakan Model Software Development Life Cycle (SDLC) Di Apotek Marifa," *Jurnal Terapan Sains & Teknologi*, vol. 1, no. 4, pp. 11-21, 2019.
- [21] E. Astutik and Mustagrifin, "Sistem Informasi Ketersediaan Obat menggunakan Framework Laravel di Apotek Mugi Sehat Limpung Batang," *Informatika dan RPL*, vol. 2, no. 1, pp. 19-25, 2020.

LAMPIRAN A

HASIL WAWANCARA

Pewawancara 1 (P1) : David Anthony Nathanael
Narasumber 1 (N1) : Pemilik Apotek 99

P1 : Sistem seperti apa yang kira kira ibu perlukan pada perusahaan ibu?

N1 : Permasalahan utamanya adalah sistem kami belum memiliki sinkronisasi penyimpanan data yang baik, dimana segala pencatatan dilakukan pada sebuah program desktop, dalam satu perangkat, dan dilakukan tanpa koneksi internet, jadi saat mencari data yang diperlukan, membutuhkan waktu untuk mencari, belum lagi jika program apotek nya mengalami error/crash/hang, atau komputernya tidak ingin menyala. Selain mengurus pesanan pada toko, kami juga harus mengurus produk dan pesanan melalui sistem marketplace, melalui aplikasi hasil kerjasama dengan mitra kami, pada *smartphone*. Karena sistem mitra tidak sinkron terhadap sistem kami, maka kami harus menginput produk satu per satu, secara manual. Banyaknya produk yang kami miliki, membuat kami tidak dapat memeriksa dan mencocokkan stok tersedia pada seluruh produk satu per satu di marketplace, sehingga pernah adanya sebuah pesanan berisi produk yang telah lama kosong. Hal ini membuat kami harus mengeluarkan usaha lebih dalam mencari produk tersebut, atau bila benar-benar tidak ditemukan, maka apotek terpaksa membatalkan pesanan. Bila kami

terlalu sering membatalkan pesanan, maka reputasi dan kredibilitas toko dapat menurun, dan itu dapat mempengaruhi kepuasan terhadap pelanggan kami di Tokopedia.

P1 : Aspek apa saja yang ibu perlukan untuk menunjang sistem manajemen produk yang ibu inginkan?

N1 : Sebenarnya kami cukup kesulitan untuk mengecek dan menyamai stok satu per satu produk kami, pada gudang dan sistem, karena jumlahnya yang banyak, serta bila sistemnya terpisah, maka kami harus bekerja beberapa kali untuk menyesuaikan setiap stok produk. Biasanya kami prioritaskan *marketplace* dulu, terutama masalah jumlah stok produk, agar tidak terjadi pesanan Tokopedia yang produknya kosong. Sebenarnya, kami punya dua toko yang berbeda di Tokopedia, dengan mitra yang berbeda juga, jadi kami bekerja dengan 3 sistem. Saya pernah dihubungi salah satu mitra, karena jarang mengupdate produk di Tokopedia. Saya sempat mengeluhkan masalah yang saya sampaikan tadi pada Anda, namun setidaknya mereka ada inisiatif mau bantu, dengan mereka kirim spreadsheet, apotek isi, lalu kirim ke mereka, dan mereka yang bakal input secara manual. Kalau di program desktop kami, kami hanya prioritaskan untuk menyesuaikan harga jualnya saja, jadi risikonya, bakal ada beberapa produk dengan stok minus, sehingga tidak merepresentasikan jumlah stok produk yang sebenarnya. Yang kami harapkan, ada sistem yang dapat menyatukan mereka, sehingga

stok produk dapat sinkron di sistem kami, dan kami dapat mengurus pesanan.

P1 : Bisa jelaskan bu, kira-kira cara kerja apotek dalam mengurus Tokopedia, dari app sampai prosedur kirim? Dan, sistem Tokopedia yang apotek urus, itu milik apotek atau bukan?

N1 : Baik, jadi lapak Tokopedia yang apotek urus selama ini, bukan punya saya. Itu punya mitra. Jadi perusahaan mitra ini menawarkan kerjasama kepada banyak Apotek, buat berjualan semua produk obat di Tokopedia, termasuk obat keras. Di apotek, ada beberapa kategori obat, seperti obat bebas, dan obat keras. Obat keras itu adalah obat yang peredarannya dijual terbatas, yang hanya bergantung pada resep dokter. Obat keras itu dulu bisa dijual bebas di Tokopedia dan *marketplace* lain, misalnya Shopee, namun setelahnya, kurang lebih awal 2020, semua obat keras itu ditarik dari lapak *marketplace*. Disitulah, mitra apotek mulai menawarkan jasanya, supaya kami tetap dapat berjualan. Mereka hanya memberi persyaratan kepada apotek, untuk memberikan fotokopi surat izin apotek dan surat izin peredaran obat keras, berbeda kalau mendaftar di Tokopedia sendiri, dimana administrasinya lebih sulit, untuk berjualan. Shopee bahkan setau saya tidak mengizinkan obat keras, sehingga kalau apotek mengupload obat keras, nanti dihapus beberapa jam kemudian, atau bisa saja disuspend tokonya, jika terlalu sering. Ini saya tau dari teman saya, yang pernah berjualan obat di Shopee. Kembali ke Tokopedia, jadi kami urus produk dan pesanan

kami, melalui aplikasi *smartphone*, yang disediakan oleh mitra. Mitra tidak memberikan kami akses ke Tokopedia Seller nya langsung. Kan kita tau kalau Tokopedia punya fitur *two factor authentication* saat mau *login* ke Tokopedia, sementara akses *two factor* nya di mereka, sehingga setiap kita *login* dan meminta akses kode, pasti ditanya sama mitra buat apa, jadi akses nya tidak bisa sembarangan. Dengan begitu, maka semua aktivitas pengelolaan produk dan transaksi dijalankan di sistem buatan mitra. Kadang-kadang, saya pernah mendapatkan chat dari *customer*, bahwa ia telah memesan produk di apotek kami, namun saat cek di sistem mitra, pesannya belum masuk, sehingga terjadi *delay*, dan apotek tidak bisa proses pesanan, sebelum masuk. Lalu, karena bergantung sama sistem mitra, maka data produk nya juga ikut punya mitra. Jadi pernah ya, kami ingin mengupload sebuah produk, namun produknya belum ada di database mitra, jadi kami tidak bisa menambahkan produk tersebut di *marketplace*, sehingga produk itu hanya bisa kami jual di apotek langsung. Kemudian, jika *customer* memesan obat keras, maka *customer* diharuskan mengupload resep dokter. Fiturnya memang dibuat khusus antara Tokopedia dan mitra, namun sebenarnya fitur ini tidak ada pengawasan, dan hanya diperiksa oleh masing-masing apotek, jadi walaupun *customer* upload gambar apapun, meski bukan resep, tetap kami *accept*. Ya, walaupun kami tahu salah, namun sejauh ini, apotek kami masih baik-baik saja dan tidak dianggap melanggar ketentuan, perihal fitur resep. Saat pesanan masuk,

dan produknya tersedia, saya atau karyawan saya mengambil dan mempacking produk secara lengkap, kemudian memanggil kurir. Kurir datang, dan mengambil pack tersebut. Kurirnya bisa instan, pakai gosend atau grab-send, yang langsung diantar ke *customer*, atau kurir biasa, seperti sicepat, anteraja, tiki, jne, dan sebagainya, yang diproses dulu di gudang mereka, kalau jaraknya sangat jauh dari apotek. Begitu produk sampai, maka saldo masuk ke rekening Tokopedia kami, dan mendapatkan rating seperti pada umumnya, bila *customer* memberikan.

P1 : Oke, lalu boleh jelaskan sistem apotek yang sekarang, seperti apa?

N1: Baik, jadi sistem kami dijalankan dalam sebuah program desktop, programnya milik perusahaan teman saya, Marulam. Saya sudah kenal dia dari lama, jadi program ini sudah digunakan dari tahun 2010-2011, kalau tidak salah. Sebenarnya, program ini sudah cukup membantu bisnis kami, terutama dalam mengelola produk dan transaksi di apotek. Namun, seperti kita tahu, komputer itu kan bisa rusak dan mati kapan saja. Paling sederhana, ya tidak mau menyala. Karena tidak terhubung dengan koneksi internet, jadi kalau ada kejadian tersebut, mau tidak mau harus install ulang, panggil orang IT nya lagi, dan datanya dari awal lagi. Mungkin kalau data produk, mereka sudah siapkan cadangannya, namun saya kurang tahu, itu data produk pada tahun berapa, jadi kalau ada jenis produk baru keluaran tahun sekarang, apotek harus tambah data baru lagi. Namun, selain data produk, data penjualan juga hilang, ini yang tidak bisa dibackup oleh mereka, jadi

misalnya data penjualan 3 tahun lalu, bisa saja hilang dan tidak ditemukan lagi. Sebenarnya, kami banyak mendapatkan banyak penawaran sistem sama pihak luar, yang fiturnya lebih canggih dari Marulam. Ada banyak benefit yang mereka tawarkan, sampai ada nyebut sistem cloud, yang saya belum paham. Tapi, yang buat saya berpikir dua kali, untuk menerima sistem mereka, selain masalah *cost*, ya masalah backup dan sinkron produk tadi. Produk kami di apotek lumayan banyak. Kalau sistem nya baru, ya saya dan pegawai harus input satu per satu lagi, yang lumayan menguras tenaga dan waktu, buat cari di etalase. Selain itu, pastinya butuh waktu untuk sosialisasi dan penyesuaian lagi terhadap sistem, dimana saya yakin belum tentu semua fiturnya berguna untuk mendukung proses bisnis apotek. Sebenarnya, soal sistem mitra marketplace, kami juga tidak terlalu paham sepenuhnya, dengan fitur-fitur yang disediakan, tapi kami tahu manfaatnya, agar *customer* tetap dapat membeli produk obat dari kami. Apalagi sekarang masih musim pandemi, yang kadang *customer* belum tentu mau berbelanja di apotek langsung.

P1: Oke, jadi apotek butuh sebuah sistem untuk mensinkronisasi produk dan pesanan di dalam apotek dan *marketplace*? Tapi setelah saya mendengar cerita bisnis dan masalah yang dihadapi ibu, saya berpikir bahwa sistem yang saya rancang, tidak akan dapat menjual semua produk di apotek, bu. Bagaimana? Terutama masalah obat keras, yang

harus upload resep dokter, dimana itu fitur khusus yang hanya disediakan untuk mitra apotek, betul bu?

N1: Ya, betul, tidak apa-apa. Saya rasa itu cukup untuk memmanage produk obat dan pesanan kami. Dengan segala kekurangan pada sistem kami yang sekarang, saya bersyukur bahwa saat ini, apotek kami masih bertahan, karena apotek kami telah berdiri selama hampir 20 tahun, dan telah dikenal oleh masyarakat Kota Bogor, sehingga banyak *customer* yang datang langsung ke kami. Ngomong-ngomong, kamu kuliah dimana, dan semester berapa? Apakah ini buat tugas kuliah kamu?

P1 : Ya betul bu, saya berkuliah di UPH, jurusan sistem informasi, semester 7, saat ini sedang skripsi atau tugas akhir. Rencananya, saya ingin merancang sebuah sistem yang ada menggunakan API, jadi sistem rancangan saya bisa berinteraksi dengan sistem lain. Kalau hubungannya sama apotek dan cerita bisnis tadi, berarti data produk dikelola dalam sistem saya, yang nanti berpengaruh ke *marketplace*.

N1: Baik, semoga lancar ya skripsi kamu, dan sukses, sehingga sistem kamu bisa membuat kami untuk bisa mengelola aset kami dengan lebih baik.

P1 : Sama-sama.

LAMPIRAN B

FORM LEMBAR MONITORING BIMBINGAN TUGAS AKHIR

Nama Mahasiswa / Student Name : David Anthony Nathanael
 NPM / Student ID Number : 01081190015
 Program Studi / Study Program : Sistem Informasi
 Fakultas / Faculty : Ilmu Komputer
 Lokasi Kampus / Campus Location : Lippo Village, Karawaci
 Semester : 10

Nama Dosen Pembimbing/Final Project Advisor : Hery, S.Kom., M.M.S.I.

Minggu Ke- Week of-	Hari Day	Tanggal Date	Materi yang didiskusikan The Material Discussed	Tanda Tangan Dosen Pembimbing Final Project Advisor Signature	Catatan Remarks
1	Jumat	16 September 2022	Pencarian sistem pihak ketiga		Mencari sistem API integrator pihak ketiga, yang dapat berinteraksi antara sistem apotek dan sistem marketplace.
2	Minggu	9 Oktober 2022	Pembahasan platform sistem ketiga		Menyampaikan beberapa platform sistem pihak ketiga yang akan digunakan, seperti Sirclo Store, Berdu, Jubelio, dan SmartSeller. Dan SmartSeller terpilih.
3	Kamis	20 Oktober 2022	Update progress Tugas Akhir		Sistem telah dapat melakukan login/register, CRUD Produk & Kategori Produk, Activity Log, Profile, User, dan Customer.
4	Jumat	28 Oktober 2022	Update progress Tugas Akhir		Sistem telah dilengkapi fitur SMTP Email, Export Product melalui CSV/PDF, Import Product melalui CSV ke database, simpan API key melalui database, serta menyerahkan laporan Table Relationship Diagram dan Use Case.

5	Jumat	25 November 2022	Update progress Tugas Akhir		Sistem telah dapat menambahkan/menghapus produk dengan API SmartSeller, dan fitur penambahan/perubahan key API akan melalui tahap tes koneksi API(bila menampilkan pesan gagal, maka query tidak dijalankan).
6	Senin	5 Desember 2022	Update progress Tugas Akhir		Fitur pengelolaan produk telah berhasil terintegrasi dengan SmartSeller, dari menambahkan, mengubah, dan menghapus produk. Hal ini juga berlaku untuk kategori dan gambar, serta unit dan supplier. Fitur transaksi toko, transaksi marketplace, dan laporan sedang on progress. Website tengah diuji coba pada hosting apotek99.my.id.
7	Selasa	12 Desember 2022	Update progress Tugas Akhir		Pembuatan transaksi toko sedang on progress, perihal AJAX untuk pengelolaannya. Kelola Order dan laporan sedang on progress. Stok produk dan cari produk di dashboard berjalan dengan baik.
8	Rabu	21 Desember 2022	Update progress Tugas Akhir		Paper sudah mulai diisi dengan screenshot UI dan diagram, sedang dalam progress. Fitur yang penting namun belum selesai, seperti laporan dan kelola pesanan marketplace.
9	Kamis	29 Desember 2022	Update progress Tugas Akhir		Melaporkan progress TA, yang telah mencapai 70-80% untuk sistem.
10	Sabtu	31 Desember 2022	Update progress Tugas Akhir		Menanyakan tentang lampiran UAT, namun dikarenakan deadline sudah dekat, maka dapat dilewatkan terlebih dahulu, dan akan ditambahkan dalam bentuk lampiran saat revisi nanti.

Tangerang, 2 Januari 2023

Disetujui oleh/*Approved by:*
Dosen Pembimbing/*Final Project Advisor*

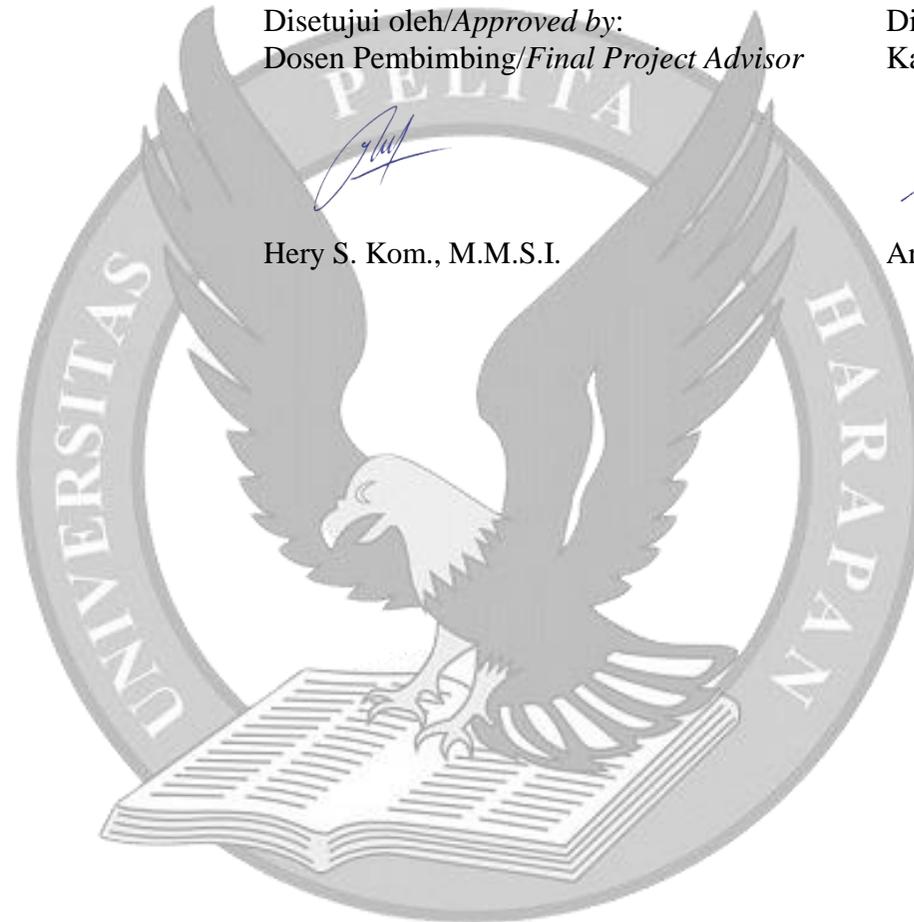


Hery S. Kom., M.M.S.I.

Diketahui oleh/*Acknowledged by:*
Kaprodi/*Department Chair*



Arnold Aribowo, S.T., M.T.



LAMPIRAN C
LINK VIDEO DEMO

Link Video Demo : [DemoProgram_01081190015](#)



LAMPIRAN D

TAHAP PENGUJIAN

Role : Admin

Fungsi :

- 1) Mengelola informasi produk

Tabel D. 1 *Test Case* Fitur Mengelola Informasi Produk oleh Admin sebagai Aktor

<i>Step Description</i>	<i>Test Steps</i>	<i>Expected Results</i>	<i>Result (Pass/Fail)</i>
Mengakses informasi produk	Mengakses submenu "Informasi" pada menu "Product"	Informasi nama produk, berat produk, kategori produk, dan unit produk muncul.	Pass
		Dapat melihat rincian informasi produk pada baris tabel masing-masing	Pass
		Dapat mengakses "Batch Produk" dengan daftar <i>batch</i> produk dengan stok tersedia	Pass
Mengelola informasi produk	Menambahkan informasi produk baru dengan data nama produk yang sudah ada	Muncul pesan error: "Mohon untuk tidak menginput nama produk yang telah tersedia"	Pass

Tabel D. 1 *Test Case* Fitur Mengelola Informasi Produk oleh Admin sebagai Aktor (Lanjutan)

	Menambahkan informasi produk baru dengan data nama produk yang belum ada	Data produk tersimpan pada tabel produk	<i>Pass</i>
	Merubah informasi produk	Informasi produk berhasil dirubah dan muncul notifikasi “Produk berhasil diubah”	<i>Pass</i>
	Menghapus data informasi produk	Data pada tabel <i>product</i> terhapus	<i>Pass</i>

2) Mengelola *batch* produk

Tabel D. 2 *Test Case* Fitur Mengelola *Batch* Produk oleh Admin sebagai Aktor

<i>Step Description</i>	<i>Test Steps</i>	<i>Expected Results</i>	<i>Result (Pass/Fail)</i>
Mengakses <i>batch</i> produk	Mengakses <i>batch</i> pada tabel “Informasi Produk”	Informasi harga beli, harga jual, qty awal, qty saat ini, expired, dan tanggal beli muncul, pada produk yang dipilih.	<i>Pass</i>
	Mengakses <i>batch</i> dengan ID Produk yang tidak valid	Muncul pesan <i>error</i> : “Batch tidak valid, silakan buka batch lain.”	<i>Pass</i>
Mengelola <i>batch</i> produk	Menekan tombol Tambah	Menampilkan formulir penambahan <i>batch</i> baru	<i>Pass</i>
	Menambahkan <i>batch</i> dengan mengosongkan input pada formulir penambahan <i>batch</i> produk	Menghasilkan pesan <i>error</i> “Batch tidak berhasil dibuat.”	<i>Pass</i>

Tabel D. 2 *Test Case* Fitur Mengelola *Batch* Produk oleh Admin sebagai Aktor (Lanjutan)

	Menambahkan <i>batch</i> dengan mengisi input yang valid pada formulir penambahan <i>batch</i> produk	Menambahkan data <i>batch</i> produk terbaru	<i>Pass</i>
	Merubah <i>batch</i> produk	Data <i>batch</i> berubah sesuai input	<i>Pass</i>
	Merubah stok saat ini pada <i>batch</i> produk	Data stok saat ini berubah sesuai input	<i>Pass</i>
	Menghapus <i>batch</i> produk	Penghapusan <i>batch</i> berhasil	<i>Pass</i>

3) Mengelola kategori produk

Tabel D. 3 *Test Case* Fitur Mengelola Kategori Produk oleh Admin sebagai Aktor

<i>Step Description</i>	<i>Test Steps</i>	<i>Expected Results</i>	<i>Result (Pass/Fail)</i>
Mengakses kategori produk	Mengakses submenu kategori produk pada menu <i>Product</i>	Informasi kategori produk	<i>Pass</i>
Menambah kategori produk	Menekan tombol Tambah	Menampilkan formulir penambahan kategori baru	<i>Pass</i>
	Mengisi data kosong pada formulir penambahan kategori	Menampilkan pesan <i>error</i> "Mohon melengkapi kategori"	<i>Pass</i>
	Mengisi formulir tambah dengan kategori yang sudah tersedia	Menampilkan pesan <i>error</i> " Mohon untuk tidak menginput kategori yang sudah ada."	<i>Pass</i>

Tabel D. 3 *Test Case* Fitur Mengelola Kategori Produk oleh Admin sebagai Aktor (Lanjutan)

	Mengisi formulir tambah dengan kategori yang belum tersedia	Menyimpan data kategori terbaru	<i>Pass</i>
Mengubah kategori produk	Menekan tombol Edit pada baris kategori yang akan diubah	Menampilkan formulir perubahan kategori	<i>Pass</i>
	Mengisi formulir ubah dengan kategori yang sudah tersedia	Menampilkan pesan <i>error</i> " Mohon untuk tidak menginput kategori yang sudah ada."	<i>Pass</i>
	Mengisi formulir ubah dengan kategori yang belum tersedia	Merubah data kategori sesuai input pada formulir ubah	<i>Pass</i>
Menghapus kategori	Menekan tombol Hapus pada baris kategori yang akan dihapus	Menampilkan pesan konfirmasi penghapusan kategori	<i>Pass</i>
	Mengonfirmasi penghapusan kategori	Menghapus data kategori	<i>Pass</i>

4) Mengelola unit produk

Tabel D. 4 *Test Case* Fitur Mengelola Unit Produk oleh Admin sebagai Aktor

<i>Step Description</i>	<i>Test Steps</i>	<i>Expected Results</i>	<i>Result (Pass/Fail)</i>
Mengakses unit produk	Mengakses submenu unit produk pada menu <i>Product</i>	Informasi unit produk	<i>Pass</i>
Menambah unit produk	Menekan tombol Tambah	Menampilkan formulir penambahan unit baru	<i>Pass</i>

Tabel D. 4 *Test Case* Fitur Mengelola Unit Produk oleh Admin sebagai Aktor (Lanjutan)

	Mengisi data kosong pada formulir penambahan unit	Menampilkan pesan <i>error</i> "Mohon melengkapi unit"	<i>Pass</i>
	Mengisi formulir tambah dengan unit yang sudah tersedia	Menampilkan pesan <i>error</i> " Mohon untuk tidak menginput unit yang sudah ada."	<i>Pass</i>
	Mengisi formulir tambah dengan unit yang belum tersedia	Menyimpan data unit terbaru	<i>Pass</i>
Mengubah unit produk	Menekan tombol Edit pada baris unit yang akan diubah	Menampilkan formulir perubahan unit	<i>Pass</i>
	Mengisi formulir ubah dengan unit yang sudah tersedia	Menampilkan pesan <i>error</i> " Mohon untuk tidak menginput unit yang sudah ada."	<i>Pass</i>
	Mengisi formulir ubah dengan unit yang belum tersedia	Merubah data unit sesuai input pada formulir ubah	<i>Pass</i>
Menghapus unit	Menekan tombol Hapus pada baris unit yang akan dihapus	Menampilkan pesan konfirmasi penghapusan unit	<i>Pass</i>
	Mengonfirmasi penghapusan unit	Menghapus data unit	<i>Pass</i>

5) Mengelola laporan

Tabel D. 5 *Test Case* Fitur Mengelola Laporan oleh Admin sebagai Aktor

<i>Step Description</i>	<i>Test Steps</i>	<i>Expected Results</i>	<i>Result (Pass/Fail)</i>
Mengakses laporan	Mengakses submenu laporan laba jual beli pada menu Laporan	Menampilkan grafik dan tabel laporan penjualan dan pembelian setiap bulan, per tahun sekarang	<i>Pass</i>
	Mengubah tahun laporan penjualan dan pembelian	Merubah dan menampilkan data grafik dan tabel berdasarkan tahun yang dipilih	<i>Pass</i>

6) Mengelola *user*

Tabel D. 6 *Test Case* Fitur Mengelola *User* oleh Admin sebagai Aktor

<i>Step Description</i>	<i>Test Steps</i>	<i>Expected Results</i>	<i>Result (Pass/Fail)</i>
Mengakses daftar <i>user</i>	Mengakses submenu <i>user</i> pada menu Pengaturan	Terdapat daftar user dengan informasi sebagai berikut: <ul style="list-style-type: none"> - Nama - Email - Gambar Profil - Peran - Status <i>user</i> 	<i>Pass</i>

Tabel D. 6 *Test Case* Fitur Mengelola *User* oleh Admin sebagai Aktor (Lanjutan)

Menambahkan <i>user</i> baru	Mengklik tombol "Tambah"	Mengakses halaman <i>pop-up</i> untuk menambahkan informasi <i>user</i>	<i>Pass</i>
	Mengisi email <i>user</i> dengan email yang sudah ada	Muncul pesan error "Email telah terdaftar, silakan gunakan Email lain"	<i>Pass</i>
	Menambahkan <i>user</i> terbaru	<i>User</i> baru berhasil ditambahkan. Muncul notifikasi "Data berhasil dibuat! Segera Aktivasi melalui email."	<i>Pass</i>
Merubah <i>user</i>	Mengklik tombol "Ubah <i>user</i> "	Mengakses halaman <i>pop-up</i> untuk merubah informasi <i>user</i> : Nama, Email(<i>readonly</i>), Role, Active	<i>Pass</i>
	Mengosongkan input nama pada formulir ubah <i>user</i>	Menampilkan pesan <i>error</i> "Mohon mengisi nama"	<i>Pass</i>
	Merubah data <i>user</i> dengan input lengkap pada formulir	Menyimpan data <i>user</i> sesuai input pada formulir edit <i>user</i>	<i>Pass</i>
Menghapus <i>user</i>	Mengklik tombol Hapus pada baris <i>user</i>	Muncul notifikasi konfirmasi untuk menghapus <i>user</i>	<i>Pass</i>
	Mengonfirmasi penghapusan <i>user</i>	Menghapus data <i>user</i>	<i>Pass</i>

7) Mengelola *role*

Tabel D. 7 *Test Case* Fitur Mengelola *Role* oleh Admin sebagai Aktor

<i>Step Description</i>	<i>Test Steps</i>	<i>Expected Results</i>	<i>Result (Pass/Fail)</i>
Mengakses daftar <i>role</i>	Mengakses submenu <i>role management</i> pada menu “Menu”	Terdapat daftar <i>role</i> dengan informasi sebagai berikut: - <i>Title Role</i> - Tombol Akses - <i>Check box</i> pada “ <i>Can Be Registered?</i> ”	<i>Pass</i>
Menambah <i>role</i>	Mengklik tombol “ <i>Add Role</i> ”	Mengakses halaman <i>pop-up</i> untuk menambahkan informasi <i>role</i>	<i>Pass</i>
	Mengosongkan formulir tambah <i>role</i>	Muncul pesan error “ <i>Role</i> tidak berhasil dibuat”	
	Menambah <i>role</i> yang sudah ada	Muncul pesan error “ <i>Role</i> tidak berhasil dibuat. Mohon untuk tidak menginput <i>role</i> yang telah tersedia”	<i>Pass</i>
	Menambah <i>role</i> yang belum ada	<i>Role</i> baru berhasil ditambahkan	<i>Pass</i>
Menghapus <i>role</i>	Menekan hapus pada baris <i>role</i> yang akan dihapus	Muncul notifikasi konfirmasi untuk menghapus <i>role</i>	<i>Pass</i>
	Mengkonfirmasi penghapusan admin	Penghapusan <i>role</i> tidak berhasil, muncul pesan error “Admin tidak dapat dihapus, demi menjaga sistem tetap diakses normal”	<i>Pass</i>
	Mengkonfirmasi penghapusan <i>role</i>	Penghapusan <i>role</i> berhasil	<i>Pass</i>

8) Mengelola menu

Tabel D. 8 *Test Case* Fitur Mengelola Menu oleh Admin sebagai Aktor

<i>Step Description</i>	<i>Test Steps</i>	<i>Expected Results</i>	<i>Result (Pass/Fail)</i>
Mengakses menu	Mengakses submenu “ <i>menu management</i> ” pada menu “Menu”	Terdapat daftar menu dengan informasi sebagai berikut: <ul style="list-style-type: none"> - Menu - <i>From Title</i> - URL - <i>Icon</i> - Kata “<i>Yes</i>” apabila status menu adalah aktif - Kata “<i>No</i>” apabila status menu adalah tidak aktif - <i>dataTarget</i> 	<i>Pass</i>
Menambahkan menu baru	Mengklik tombol “Add Menu”	Mengakses halaman <i>pop-up</i> untuk menambahkan informasi menu	<i>Pass</i>
	Mengosongkan satu atau lebih <i>field</i> pada formulir informasi menu baru	Muncul pesan error “ Menu tidak berhasil dibuat.” dengan pesan error pada <i>field</i> yang belum terpenuhi	<i>Pass</i>
	Menambahkan menu dengan data menu yang sudah tersedia	Muncul pesan error “Menu tidak berhasil dibuat.”	<i>Pass</i>
	Menambahkan menu dengan data menu yang tidak tersedia	Menu baru berhasil ditambahkan. Muncul pesan “Menu berhasil dibuat.”	<i>Pass</i>
Merubah menu	Mengklik tombol “Edit” pada Menu yang akan dirubah	Mengakses halaman <i>pop-up</i> untuk merubah informasi menu	<i>Pass</i>

Tabel D. 8 *Test Case* Fitur Mengelola Menu oleh Admin sebagai Aktor (Lanjutan)

	Merubah menu dengan data menu yang sudah tersedia	Muncul pesan error " Menu tidak berhasil diubah." dengan pesan error pada <i>field</i> yang belum terpenuhi	<i>Pass</i>
	Merubah menu dengan data menu yang belum tersedia	Menu berhasil dirubah. Muncul pesan "Menu berhasil diubah."	<i>Pass</i>
Menghapus menu	Menekan hapus pada baris menu yang akan dihapus	Muncul notifikasi konfirmasi untuk menghapus menu	<i>Pass</i>
	Mengkonfirmasi penghapusan menu	Penghapusan menu berhasil	<i>Pass</i>

9) Mengelola submenu

Tabel D. 9 *Test Case* Fitur Mengelola Submenu oleh Admin sebagai Aktor

<i>Step Description</i>	<i>Test Steps</i>	<i>Expected Results</i>	<i>Result (Pass/Fail)</i>
Mengakses submenu	Mengakses submenu " <i>sub-menu management</i> " pada menu "Menu"	Terdapat daftar submenu dengan informasi sebagai berikut: - Sub-Menu - <i>From Menu</i> - URL - Urutan Submenu - Status Submenu	<i>Pass</i>
Menambahkan submenu baru	Mengklik tombol " <i>Add New Sub-Menu</i> "	Mengakses halaman <i>pop-up</i> untuk menambahkan informasi submenu	<i>Pass</i>

Tabel D. 9 *Test Case* Fitur Mengelola Submenu oleh Admin sebagai Aktor (Lanjutan)

	Mengosongkan satu atau lebih <i>field</i> pada formulir informasi submenu baru	Muncul pesan error "Sub-Menu tidak berhasil dibuat." dengan pesan error pada <i>field</i> yang belum terpenuhi	<i>Pass</i>
	Menambahkan submenu dengan data submenu yang sudah tersedia	Muncul pesan error "Submenu tidak berhasil dibuat."	<i>Pass</i>
	Menambahkan submenu dengan data submenu yang tidak tersedia	Submenu baru berhasil ditambahkan. Muncul pesan "Submenu berhasil dibuat."	<i>Pass</i>
Merubah submenu	Mengklik tombol "Edit" pada Submenu yang akan dirubah	Mengakses halaman <i>pop-up</i> untuk merubah informasi submenu	<i>Pass</i>
	Merubah submenu dengan data submenu yang sudah tersedia	Muncul pesan error "Submenu tidak berhasil diubah." dengan pesan error pada <i>field</i> yang belum terpenuhi	<i>Pass</i>
	Merubah submenu dengan data submenu yang belum tersedia	Submenu berhasil dirubah. Muncul pesan "Submenu berhasil diubah."	<i>Pass</i>
Menghapus submenu	Menekan hapus pada baris submenu yang akan dihapus	Muncul notifikasi konfirmasi untuk menghapus submenu	<i>Pass</i>
	Mengkonfirmasi penghapusan submenu	Penghapusan submenu berhasil	<i>Pass</i>

10) Mengelola profil

Tabel D. 10 *Test Case* Fitur Mengelola Profil Oleh Admin Sebagai Aktor

<i>Step Description</i>	<i>Test Steps</i>	<i>Expected Results</i>	<i>Result (Pass/Fail)</i>
Mengelola profil	Mengakses menu <i>profile</i>	Menampilkan halaman profil dengan informasi sebagai berikut: - Nama - Email - Role - Profile Picture	Pass
	Menyimpan data profil dengan mengosongkan formulir nama	Menampilkan pesan <i>error</i> pada formulir nama	Pass
	Mengupload file bukan gambar	Menampilkan pesan <i>error</i> "Upload gambar gagal"	Pass
	Mengupload file gambar dengan ukuran di atas 2 MB	Menampilkan pesan <i>error</i> "Upload gambar gagal"	Pass
	Mengupload file gambar dengan ukuran tidak melebihi 2 MB	Menyimpan data gambar pada <i>database</i>	Pass
Merubah <i>password</i>	Mengklik tombol "Ubah <i>Password</i> "	Menampilkan formulir perubahan <i>password</i>	Pass
	Mengosongkan satu atau lebih <i>field</i> pada formulir perubahan <i>password</i>	Menampilkan pesan <i>error</i> pada kolom input yang belum terpenuhi	Pass
	Mengisi <i>password</i> dibawah 8 karakter	Menampilkan pesan <i>error</i> "The New Password field must be at least 8 characters in length."	Pass

Tabel D. 10 *Test Case* Fitur Mengelola Profil Oleh Admin Sebagai Aktor (Lanjutan)

	Mengisi <i>password</i> lama dengan <i>password</i> yang salah	Menampilkan pesan "Password lama salah, silakan coba lagi"	<i>Pass</i>
	Mengisi input <i>password</i> baru dan konfirmasi <i>password</i> dengan <i>password</i> yang berbeda	Menampilkan pesan " <i>The field does not match the other field</i> "	<i>Pass</i>
	Mengisi input <i>password</i> lama dengan <i>password</i> yang sesuai, dan mengisi <i>password</i> yang sama pada <i>password</i> baru dan konfirmasi <i>password</i>	Menyimpan data <i>password</i> terbaru	<i>Pass</i>

11) Menambah transaksi toko

Tabel D. 11 *Test Case* Fitur Menambah Transaksi Toko oleh Admin sebagai Aktor

<i>Step Description</i>	<i>Test Steps</i>	<i>Expected Results</i>	<i>Result (Pass/Fail)</i>
Mengakses penambahan transaksi toko	Mengakses submenu "Tambah Penjualan Toko" pada menu Transaksi	Menampilkan halaman transaksi penjualan toko	<i>Pass</i>
Mengelola keranjang transaksi toko	Mengetik kolom nama produk	Menampilkan <i>dropdown</i> produk dan <i>batch</i>	<i>Pass</i>
	Menekan tombol Tambah Produk dengan data produk dan batch yang tidak valid	Menampilkan pesan <i>error</i> "Produk Batch tidak ditemukan"	<i>Pass</i>

Tabel D. 11 *Test Case* Fitur Menambah Transaksi Toko oleh Admin sebagai Aktor (Lanjutan)

	Menekan tombol Tambah Produk dengan data produk dan batch yang valid	Data <i>batch</i> bertambah ke keranjang transaksi	<i>Pass</i>
	Merubah kuantitas pada keranjang transaksi	Informasi jumlah dan total transaksi berubah	<i>Pass</i>
Mengelola pembayaran pada transaksi	Mengisi input diskon dibawah 100 persen dari jumlah transaksi	Informasi total transaksi berubah sesuai potongan diskon	<i>Pass</i>
	Mengisi input diskon diatas 100 persen dari jumlah transaksi	Input diskon berubah menjadi nol, dan total transaksi berubah menjadi sama dengan jumlah transaksi	<i>Pass</i>
	Mengisi input lain-lain dibawah jumlah transaksi	Informasi total transaksi berubah sesuai potongan lain-lain	<i>Pass</i>
	Mengisi input lain-lain diatas jumlah transaksi	Input lain-lain berubah menjadi nol, dan total transaksi berubah menjadi sama dengan jumlah transaksi	<i>Pass</i>
	Mengisi input jumlah pembayaran dibawah total transaksi	Menampilkan data uang kembali dalam jumlah negatif	<i>Pass</i>
	Mengisi input jumlah pembayaran diatas total transaksi	Menampilkan data uang kembali dalam jumlah positif	<i>Pass</i>
Menyimpan transaksi	Menekan tombol simpan transaksi dengan jumlah pembayaran dibawah total transaksi	Menampilkan pesan <i>error</i> "Mohon mengisi kolom bayar melebihi total belanja"	<i>Pass</i>
	Menekan tombol "simpan transaksi" dengan jumlah pembayaran diatas total transaksi	Menyimpan transaksi dan mengurangi stok <i>batch</i> produk sesuai detail transaksi	<i>Pass</i>

12) Melihat transaksi toko

Tabel D. 12 *Test Case* Fitur Melihat Transaksi Toko oleh Admin sebagai Aktor

<i>Step Description</i>	<i>Test Steps</i>	<i>Expected Results</i>	<i>Result (Pass/Fail)</i>
Mengakses histori transaksi toko	Mengakses submenu "Lihat Penjualan Toko" pada menu Transaksi	Menampilkan histori transaksi toko dengan informasi sebagai berikut: - <i>Receipt Number</i> - Total Transaksi - <i>User</i> - <i>Time</i>	<i>Pass</i>
Melihat rincian informasi produk dalam transaksi toko	Menekan tombol <i>receipt</i> pada baris transaksi yang ingin dilihat	Menampilkan rincian produk dalam transaksi toko dengan informasi sebagai berikut: - Nama Obat - <i>Batch</i> - Harga satuan - Banyak - Subtotal - Grand Total	<i>Pass</i>
Menghapus transaksi toko	Menekan tombol hapus pada baris transaksi yang ingin dihapus	Menampilkan halaman konfirmasi penghapusan transaksi	<i>Pass</i>
	Mengonfirmasi penghapusan transaksi	Menghapus transaksi, dan menambahkan qty dari transaksi ke stok <i>batch</i> produk	<i>Pass</i>

13) Melihat produk kadaluwarsa

Tabel D. 13 *Test Case* Fitur Melihat Produk Kadaluwarsa oleh Admin sebagai Aktor

<i>Step Description</i>	<i>Test Steps</i>	<i>Expected Results</i>	<i>Result (Pass/Fail)</i>
Mengakses produk kadaluwarsa	Mengakses submenu "Batch Kadaluwarsa" pada menu <i>Product</i>	Menampilkan histori produk kadaluwarsa dengan informasi sebagai berikut: - Nama Produk - <i>Batch</i> - <i>QTY</i> Beli - <i>QTY</i> Saat Ini (dengan stok diatas nol) - <i>Expired</i> - <i>Last Update</i>	<i>Pass</i>
Mengosongkan <i>batch</i> kadaluwarsa	Menekan tombol hapus pada <i>batch</i> yang ingin dihapus	Menampilkan halaman konfirmasi pengosongan stok <i>batch</i>	<i>Pass</i>
	Mengonfirmasi pengosongan <i>batch</i>	Mengubah stok <i>batch</i> produk menjadi nol	<i>Pass</i>

14) Melihat produk kosong

Tabel D. 14 *Test Case* Fitur Melihat Produk Kosong oleh Admin sebagai Aktor

<i>Step Description</i>	<i>Test Steps</i>	<i>Expected Results</i>	<i>Result (Pass/Fail)</i>
Mengakses produk kadaluwarsa	Mengakses submenu "Produk kosong" pada menu <i>Product</i>	Menampilkan histori produk kadaluwarsa dengan informasi sebagai berikut: - Nama Produk - Berat - Kategori - Unit	<i>Pass</i>
Menambah <i>batch</i> produk baru	Menekan tombol tambah pada baris produk yang akan ditambah	Menampilkan formulir penambahan stok produk terbaru	<i>Pass</i>
	Menambahkan <i>batch</i> dengan mengosongkan input pada formulir penambahan <i>batch</i> produk	Menghasilkan pesan <i>error</i> "Batch tidak berhasil dibuat."	<i>Pass</i>
	Menambahkan <i>batch</i> dengan mengisi input yang valid pada formulir penambahan <i>batch</i> produk	Menambahkan data <i>batch</i> produk terbaru	<i>Pass</i>

LAMPIRAN E

USE CASE DESCRIPTION

E.1 Mengelola Keranjang Transaksi Toko

Tabel E. 1 *Use Case Description* Mengelola Keranjang Transaksi Toko

<i>Use Case Name:</i> Mengelola Keranjang Transaksi Toko	ID: 1	<i>Importance Level:</i> High
<i>Primary Actor:</i> Admin	<i>Use Case Type:</i> Detail, essential	
<i>Stakeholders and Interests:</i> Admin – Ingin menambah atau mengubah data keranjang transaksi toko		
<i>Brief Description:</i> Use case ini menjelaskan bagaimana admin mengelola keranjang transaksi toko		
<i>Trigger:</i> Admin memilih <i>master data</i> keranjang <i>Type:</i> External		
<i>Relationships:</i> <ul style="list-style-type: none">• <i>Association:</i> Admin• <i>Include:</i>• <i>Extend:</i>• <i>Generalize:</i>		
<i>Normal Flows of Events:</i> <ol style="list-style-type: none">1) Admin memilih <i>master data</i> keranjang2) Sistem menampilkan <i>master data</i> keranjang Jika admin ingin menambah data baru ke keranjang S-1: Tambah data produk ke keranjang Jika admin ingin mengubah keranjang S-2: <i>Edit</i> data keranjang Jika admin ingin menghapus data pada keranjang S-3: Hapus data keranjang3) Sistem menyimpan data		
<i>Sub Flows:</i> S-1: Tambah data keranjang <ol style="list-style-type: none">1) Admin memasukkan data produk S-2: Edit data keranjang <ol style="list-style-type: none">1) Admin mengubah data produk di keranjang		

Tabel E. 1 *Use Case Description* Mengelola Keranjang Transaksi Toko (Lanjutan)

Alternative/Exceptional Flows:

E.2 Menghapus transaksi toko

Tabel E. 2 *Use Case Description* Menghapus Transaksi Toko

<i>Use Case Name:</i> Mengelola Keranjang Transaksi Toko	ID: 2	<i>Importance Level:</i> High
<i>Primary Actor:</i> Admin	<i>Use Case Type:</i> Detail, essential	
<i>Stakeholders and Interests:</i> Admin – Ingin menghapus transaksi toko		
<i>Brief Description:</i> Use case ini menjelaskan bagaimana admin menghapus transaksi toko		
<i>Trigger:</i> Admin memilih <i>master data</i> transaksi <i>Type:</i> External		
<i>Relationships:</i> <ul style="list-style-type: none"> • <i>Association:</i> Admin • <i>Include:</i> • <i>Extend:</i> • <i>Generalize:</i> 		
<i>Normal Flows of Events:</i> <ol style="list-style-type: none"> 1) Admin memilih <i>master data</i> transaksi 2) Sistem menampilkan <i>master data</i> transaksi Jika admin ingin menghapus transaksi S-1: Hapus transaksi 3) Sistem menghapus data 		
<i>Sub Flows:</i>		
<i>Alternative/Exceptional Flows:</i>		

E.3 Mengelola profil

Tabel E. 3 *Use Case Description* Mengelola Profil

<i>Use Case Name:</i> Mengelola Profil	ID: 3	<i>Importance Level:</i> High
<i>Primary Actor:</i> Kasir	<i>Use Case Type:</i> Detail, essential	
<i>Stakeholders and Interests:</i> Kasir – Ingin mengubah data profil		
<i>Brief Description:</i> Use case ini menjelaskan bagaimana kasir mengelola profil		
<i>Trigger:</i> Kasir mengakses data profil <i>Type:</i> External		
<i>Relationships:</i> <ul style="list-style-type: none"> • <i>Association:</i> Kasir • <i>Include:</i> • <i>Extend:</i> • <i>Generalize:</i> 		
<i>Normal Flows of Events:</i> <ol style="list-style-type: none"> 1) Kasir mengakses profil 2) Sistem menampilkan data profil Jika kasir ingin mengubah data S-1: <i>Edit</i> data profil 3) Sistem menyimpan data 		
<i>Sub Flows:</i> S-1: Edit data keranjang <ol style="list-style-type: none"> 1) Kasir mengubah data profil 		
<i>Alternative/Exceptional Flows:</i>		

E.4 Mengelola Informasi Produk

Tabel E. 4 *Use Case Description* Mengelola Informasi Produk

<i>Use Case Name:</i> Mengelola Informasi Produk	ID: 4	<i>Importance Level:</i> High
<i>Primary Actor:</i> Admin	<i>Use Case Type:</i> Detail, essential	
<i>Stakeholders and Interests:</i> Admin – Ingin menambah, mengubah, atau menghapus data informasi produk		
<i>Brief Description:</i> Use case ini menjelaskan bagaimana admin mengelola informasi produk		
<i>Trigger:</i> Admin memilih <i>master data</i> info produk <i>Type:</i> External		
<i>Relationships:</i> <ul style="list-style-type: none"> • <i>Association:</i> Admin • <i>Include:</i> • <i>Extend:</i> • <i>Generalize:</i> 		
<i>Normal Flows of Events:</i> <ol style="list-style-type: none"> 1) Admin memilih <i>master data</i> info produk 2) Sistem menampilkan <i>master data</i> info produk Jika admin ingin menambah data info produk baru S-1: Tambah data info produk Jika admin ingin mengubah info produk S-2: <i>Edit</i> data info produk Jika admin ingin menghapus info produk S-3: Hapus data info produk 3) Sistem menyimpan data 		
<i>Sub Flows:</i> <p>S-1: Tambah data info produk</p> <ol style="list-style-type: none"> 1) Admin memasukkan data info produk <p>S-2: Edit data info produk</p> <ol style="list-style-type: none"> 1) Admin mengubah data info produk 		
<i>Alternative/Exceptional Flows:</i>		

E.5 Mengelola *Batch* Produk

Tabel E. 5 *Use Case Description* Mengelola *Batch* Produk

<i>Use Case Name:</i> Mengelola <i>Batch</i> Produk	ID: 5	<i>Importance Level:</i> High
<i>Primary Actor:</i> Admin	<i>Use Case Type:</i> Detail, essential	
<i>Stakeholders and Interests:</i> Admin – Ingin menambah, mengubah, atau menghapus data <i>batch</i> produk		
<i>Brief Description:</i> Use case ini menjelaskan bagaimana admin mengelola <i>batch</i> produk		
<i>Trigger:</i> Admin memilih <i>master data batch</i> produk <i>Type:</i> External		
<i>Relationships:</i> <ul style="list-style-type: none"> • <i>Association:</i> Admin • <i>Include:</i> • <i>Extend:</i> • <i>Generalize:</i> 		
<i>Normal Flows of Events:</i> <ol style="list-style-type: none"> 1) Admin memilih <i>master data</i> produk 2) Sistem menampilkan <i>master data batch</i> produk Jika admin ingin menambah data <i>batch</i> baru S-1: Tambah data <i>batch</i> produk Jika admin ingin mengubah <i>batch</i> produk S-2: <i>Edit data batch</i> produk Jika admin ingin menghapus <i>batch</i> produk S-3: Hapus data <i>batch</i> produk 3) Sistem menyimpan data 		
<i>Sub Flows:</i> <p>S-1: Tambah data <i>batch</i> produk</p> <ol style="list-style-type: none"> 1) Admin memasukkan data <i>batch</i> produk <p>S-2: <i>Edit data batch</i> produk</p> <ol style="list-style-type: none"> 1) Admin mengubah data <i>batch</i> produk 		
<i>Alternative/Exceptional Flows:</i>		

E.6 Mengakses Rangkuman Data *Dashboard*

Tabel E. 6 *Use Case Description* Mengakses Rangkuman Data *Dashboard*

<i>Use Case Name:</i> Mengakses Rangkuman Data <i>Dashboard</i>	ID: 6	<i>Importance Level:</i> High
<i>Primary Actor:</i> Admin	<i>Use Case Type:</i> Detail, essential	
<i>Stakeholders and Interests:</i> Admin – Ingin mengakses rangkuman data <i>dashboard</i>		
<i>Brief Description:</i> Use case ini menjelaskan bagaimana admin mengakses rangkuman data <i>dashboard</i>		
<i>Trigger:</i> Admin membuka <i>dashboard</i> <i>Type:</i> External		
<i>Relationships:</i> <ul style="list-style-type: none"> • <i>Association:</i> Admin • <i>Include:</i> • <i>Extend:</i> • <i>Generalize:</i> 		
<i>Normal Flows of Events:</i> <ol style="list-style-type: none"> 1) Admin membuka dasbor 2) Sistem menampilkan rangkuman data <i>dashboard</i> 		
<i>Sub Flows:</i>		
<i>Alternative/Exceptional Flows:</i>		

E.7 Mengelola Akses *Backdoor*

Tabel E. 7 *Use Case Description* Mengelola Akses *Backdoor*

<i>Use Case Name:</i> Mengelola Akses <i>Backdoor</i>	ID: 7	<i>Importance Level:</i> High
<i>Primary Actor:</i> Admin	<i>Use Case Type:</i> Detail, essential	
<i>Stakeholders and Interests:</i> Admin – Ingin mengubah data <i>akses backdoor</i>		
<i>Brief Description:</i> <i>Use case ini menjelaskan bagaimana admin mengelola akses backdoor</i>		
<i>Trigger:</i> Admin memilih <i>master data akses backdoor</i> <i>Type:</i> External		
<i>Relationships:</i> <ul style="list-style-type: none"> • <i>Association:</i> Admin • <i>Include:</i> • <i>Extend:</i> • <i>Generalize:</i> 		
<i>Normal Flows of Events:</i> <ol style="list-style-type: none"> 1) Admin memilih <i>master data akses backdoor</i> 2) Sistem menampilkan <i>master data akses backdoor</i> Jika admin ingin mengaktifkan <i>akses backdoor</i> S-1: Tambah <i>akses backdoor</i> Jika admin ingin menonaktifkan <i>akses backdoor</i> S-2: Hapus <i>akses backdoor</i> 3) Sistem menyimpan data 		
<i>Sub Flows:</i>		
<i>Alternative/Exceptional Flows:</i>		

E.8 Mengelola Akses Menu

Tabel E. 8 *Use Case Description* Mengelola Akses Menu

<i>Use Case Name:</i> Mengelola Akses Menu	ID: 8	<i>Importance Level:</i> High
<i>Primary Actor:</i> Admin	<i>Use Case Type:</i> Detail, essential	
<i>Stakeholders and Interests:</i> Admin – Ingin mengubah data akses menu		
<i>Brief Description:</i> Use case ini menjelaskan bagaimana admin mengelola akses menu		
<i>Trigger:</i> Admin memilih <i>master data</i> akses menu <i>Type:</i> External		
<i>Relationships:</i> <ul style="list-style-type: none"> • <i>Association:</i> Admin • <i>Include:</i> • <i>Extend:</i> • <i>Generalize:</i> 		
<i>Normal Flows of Events:</i> <ol style="list-style-type: none"> 1) Admin memilih <i>master data</i> akses menu 2) Sistem menampilkan <i>master data</i> akses menu Jika admin ingin mengaktifkan akses menu S-1: Tambah akses menu Jika admin ingin menonaktifkan akses menu S-2: Hapus akses menu 3) Sistem menyimpan data 		
<i>Sub Flows:</i> S-1: Tambah akses menu <ol style="list-style-type: none"> 1) Admin memilih <i>role</i> untuk menu 		
<i>Alternative/Exceptional Flows:</i>		

E.9 Mengelola Kategori Produk

Tabel E. 9 *Use Case Description* Mengelola Kategori Produk

<i>Use Case Name:</i> Mengelola Informasi Produk	ID: 9	<i>Importance Level:</i> High
<i>Primary Actor:</i> Admin	<i>Use Case Type:</i> Detail, essential	
<i>Stakeholders and Interests:</i> Admin – Ingin menambah, mengubah, atau menghapus data kategori produk		
<i>Brief Description:</i> Use case ini menjelaskan bagaimana admin mengelola kategori produk		
<i>Trigger:</i> Admin memilih <i>master data</i> kategori produk <i>Type:</i> External		
<i>Relationships:</i> <ul style="list-style-type: none"> • <i>Association:</i> Admin • <i>Include:</i> • <i>Extend:</i> • <i>Generalize:</i> 		
<i>Normal Flows of Events:</i> <ol style="list-style-type: none"> 1) Admin memilih <i>master data</i> kategori produk 2) Sistem menampilkan <i>master data</i> kategori produk Jika admin ingin menambah data baru S-1: Tambah data kategori Jika admin ingin mengubah kategori produk S-2: <i>Edit</i> data kategori Jika admin ingin menghapus kategori produk S-3: Hapus data kategori 3) Sistem menyimpan data 		
<i>Sub Flows:</i> <p>S-1: Tambah data info produk</p> <ol style="list-style-type: none"> 1) Admin memasukkan data kategori <p>S-2: Edit data info produk</p> <ol style="list-style-type: none"> 1) Admin mengubah data kategori 		
<i>Alternative/Exceptional Flows:</i>		

E.10 Mengelola Unit Produk

Tabel E. 10 *Use Case Description* Mengelola Unit Produk

<i>Use Case Name:</i> Mengelola Unit Produk	ID: 10	<i>Importance Level:</i> High
<i>Primary Actor:</i> Admin	<i>Use Case Type:</i> Detail, essential	
<i>Stakeholders and Interests:</i> Admin – Ingin menambah, mengubah, atau menghapus data unit produk		
<i>Brief Description:</i> Use case ini menjelaskan bagaimana admin mengelola unit produk		
<i>Trigger:</i> Admin memilih <i>master data</i> unit produk <i>Type:</i> External		
<i>Relationships:</i> <ul style="list-style-type: none"> • <i>Association:</i> Admin • <i>Include:</i> • <i>Extend:</i> • <i>Generalize:</i> 		
<i>Normal Flows of Events:</i> <ol style="list-style-type: none"> 1) Admin memilih <i>master data</i> unit produk 2) Sistem menampilkan <i>master data</i> unit produk Jika admin ingin menambah data baru S-1: Tambah data unit produk Jika admin ingin mengubah unit produk S-2: <i>Edit</i> data unit Jika admin ingin menghapus unit produk S-3: Hapus data unit 3) Sistem menyimpan data 		
<i>Sub Flows:</i> <p>S-1: Tambah data unit produk</p> <ol style="list-style-type: none"> 1) Admin memasukkan data unit <p>S-2: Edit data unit produk</p> <ol style="list-style-type: none"> 1) Admin mengubah data unit 		
<i>Alternative/Exceptional Flows:</i>		

E.11 Mencari Produk di Dashboard

Tabel E. 11 *Use Case Description* Mencari Produk di *Dashboard*

<i>Use Case Name:</i> Mencari Produk di Dashboard	ID: 11	<i>Importance Level:</i> High
<i>Primary Actor:</i> Admin	<i>Use Case Type:</i> Detail, essential	
<i>Stakeholders and Interests:</i> Admin – Ingin mencari produk di halaman <i>dashboard</i>		
<i>Brief Description:</i> <i>Use case ini menjelaskan bagaimana admin mencari produk di dashboard</i>		
<i>Trigger:</i> Admin membuka <i>dashboard</i> <i>Type:</i> External		
<i>Relationships:</i> <ul style="list-style-type: none"> • <i>Association:</i> Admin • <i>Include:</i> • <i>Extend:</i> • <i>Generalize:</i> 		
<i>Normal Flows of Events:</i> <ol style="list-style-type: none"> 1) Admin membuka dasbor 2) Sistem menampilkan dasbor Admin ingin mencari data produk S-1: Cari data produk 3) Sistem menampilkan data 		
<i>Sub Flows:</i> S-1: Cari data produk <ol style="list-style-type: none"> 1) Admin memasukkan data produk 		
<i>Alternative/Exceptional Flows:</i>		

E.12 Mengakses Laporan Laba

Tabel E. 12 *Use Case Description* Mengakses Laporan Laba

<i>Use Case Name:</i> Mengakses laporan laba	ID: 12	<i>Importance Level:</i> High
<i>Primary Actor:</i> Admin	<i>Use Case Type:</i> Detail, essential	
<i>Stakeholders and Interests:</i> Admin – Ingin mengakses laporan laba		
<i>Brief Description:</i> Use case ini menjelaskan bagaimana admin mengakses laporan laba		
<i>Trigger:</i> Admin membuka <i>dashboard</i> <i>Type:</i> External		
<i>Relationships:</i> <ul style="list-style-type: none"> • <i>Association:</i> Admin • <i>Include:</i> • <i>Extend:</i> • <i>Generalize:</i> 		
<i>Normal Flows of Events:</i> <ol style="list-style-type: none"> 1) Admin membuka laporan laba 2) Sistem menampilkan laporan laba Admin ingin mencari laporan laba berdasarkan waktu S-1: Cari data waktu 3) Sistem menampilkan data 		
<i>Sub Flows:</i> S-1: Cari data waktu <ol style="list-style-type: none"> 1) Admin memasukkan data waktu 		
<i>Alternative/Exceptional Flows:</i>		

E.13 Mengelola *API Key*

Tabel E. 13 *Use Case Description* Mengelola *API Key*

<i>Use Case Name:</i> Mengelola <i>API key</i>	ID: 13	<i>Importance Level:</i> High
<i>Primary Actor:</i> Admin	<i>Use Case Type:</i> Detail, essential	
<i>Stakeholders and Interests:</i> Admin – Ingin menambah atau mengubah data <i>API key</i>		
<i>Brief Description:</i> Use case ini menjelaskan bagaimana admin mengelola data <i>API key</i>		
<i>Trigger:</i> Admin memilih <i>master data API key</i> <i>Type:</i> External		
<i>Relationships:</i> <ul style="list-style-type: none"> • <i>Association:</i> Admin • <i>Include:</i> • <i>Extend:</i> • <i>Generalize:</i> 		
<i>Normal Flows of Events:</i> <ol style="list-style-type: none"> 1) Admin memilih <i>master data API key</i> 2) Sistem menampilkan <i>master data API key</i> Jika admin ingin menambah <i>key</i> baru S-1: Tambah data <i>key</i> Jika admin ingin mengubah <i>key</i> S-2: <i>Edit data key</i> 3) Sistem menyimpan data 		
<i>Sub Flows:</i> <p>S-1: Tambah data <i>key</i></p> <ol style="list-style-type: none"> 1) Admin memasukkan data <i>key</i> baru <p>S-2: <i>Edit data key</i></p> <ol style="list-style-type: none"> 1) Admin mengubah data <i>key</i> 		
<i>Alternative/Exceptional Flows:</i>		

E.14 Mengelola *User*

Tabel E. 14 *Use Case Description* Mengelola *User*

<i>Use Case Name:</i> Mengelola <i>User</i>	ID: 14	<i>Importance Level:</i> High
<i>Primary Actor:</i> Admin	<i>Use Case Type:</i> Detail, essential	
<i>Stakeholders and Interests:</i> Admin – Ingin menambah atau mengubah data <i>user</i>		
<i>Brief Description:</i> <i>Use case</i> ini menjelaskan bagaimana admin mengelola data <i>user</i>		
<i>Trigger:</i> Admin memilih <i>master data user</i> <i>Type:</i> External		
<i>Relationships:</i> <ul style="list-style-type: none"> • <i>Association:</i> Admin • <i>Include:</i> • <i>Extend:</i> • <i>Generalize:</i> 		
<i>Normal Flows of Events:</i> <ol style="list-style-type: none"> 1) Admin memilih <i>master data user</i> 2) Sistem menampilkan <i>master data user</i> Jika admin ingin menambah data baru S-1: Tambah data <i>user</i> Jika admin ingin mengubah data S-2: <i>Edit data user</i> 3) Sistem menyimpan data 		
<i>Sub Flows:</i> <ol style="list-style-type: none"> S-1: Tambah data <i>user</i> <ol style="list-style-type: none"> 2) Admin memasukkan data <i>user</i> baru S-2: <i>Edit data user</i> <ol style="list-style-type: none"> 2) Admin mengubah data <i>user</i> 		
<i>Alternative/Exceptional Flows:</i>		

E.15 Mengelola *Role*

Tabel E. 15 *Use Case Description* Mengelola *Role*

<i>Use Case Name:</i> Mengelola <i>Role</i>	ID: 15	<i>Importance Level:</i> High
<i>Primary Actor:</i> Admin	<i>Use Case Type:</i> Detail, essential	
<i>Stakeholders and Interests:</i> Admin – Ingin menambah atau mengubah <i>role</i>		
<i>Brief Description:</i> <i>Use case</i> ini menjelaskan bagaimana admin mengelola data <i>role</i>		
<i>Trigger:</i> Admin memilih <i>master data role</i> <i>Type:</i> External		
<i>Relationships:</i> <ul style="list-style-type: none"> • <i>Association:</i> Admin • <i>Include:</i> • <i>Extend:</i> • <i>Generalize:</i> 		
<i>Normal Flows of Events:</i> <ol style="list-style-type: none"> 1) Admin memilih <i>master data role</i> 2) Sistem menampilkan <i>master data role</i> Jika admin ingin menambah <i>role</i> baru S-1: Tambah data <i>role</i> Jika admin ingin mengubah data S-2: <i>Edit data role</i> 3) Sistem menyimpan data 		
<i>Sub Flows:</i> <p>S-1: Tambah data <i>role</i></p> <ol style="list-style-type: none"> 1) Admin memasukkan data <i>role</i> baru <p>S-2: Edit data <i>user</i></p> <ol style="list-style-type: none"> 1) Admin mengubah data <i>role</i> 		
<i>Alternative/Exceptional Flows:</i>		

E.16 Mengakses Transaksi Toko

Tabel E. 16 *Use Case Description* Mengakses Transaksi Toko

<i>Use Case Name:</i> Mengakses Transaksi Toko	ID: 16	<i>Importance Level:</i> High
<i>Primary Actor:</i> Kasir	<i>Use Case Type:</i> Detail, essential	
<i>Stakeholders and Interests:</i> Kasir – Ingin mengakses data transaksi		
<i>Brief Description:</i> Use case ini menjelaskan bagaimana kasir mengakses data transaksi toko		
<i>Trigger:</i> Kasir membuka data transaksi toko <i>Type:</i> External		
<i>Relationships:</i> <ul style="list-style-type: none"> • <i>Association:</i> Kasir • <i>Include:</i> • <i>Extend:</i> • <i>Generalize:</i> 		
<i>Normal Flows of Events:</i> <ol style="list-style-type: none"> 1) Kasir memilih <i>master</i> data transaksi toko 2) Sistem menampilkan <i>master</i> data transaksi toko 		
<i>Sub Flows:</i>		
<i>Alternative/Exceptional Flows:</i>		

E.17 Mengakses *Batch* Kadaluwarsa

Tabel E. 17 *Use Case Description* Mengakses *Batch* Kadaluwarsa

<i>Use Case Name:</i> Mengakses <i>Batch</i> Kadaluwarsa	ID: 17	<i>Importance Level:</i> High
<i>Primary Actor:</i> Kasir	<i>Use Case Type:</i> Detail, essential	
<i>Stakeholders and Interests:</i> Kasir – Ingin mengakses data <i>batch</i> produk kadaluwarsa		
<i>Brief Description:</i> Use case ini menjelaskan bagaimana kasir mengakses data <i>batch</i> kadaluwarsa		
<i>Trigger:</i> Kasir membuka data produk kadaluwarsa <i>Type:</i> External		
<i>Relationships:</i> <ul style="list-style-type: none"> • <i>Association:</i> Kasir • <i>Include:</i> • <i>Extend:</i> • <i>Generalize:</i> 		
<i>Normal Flows of Events:</i> <ol style="list-style-type: none"> 1) Kasir memilih <i>master</i> data <i>batch</i> kadaluwarsa 2) Sistem menampilkan <i>master</i> data <i>batch</i> produk kadaluwarsa 		
<i>Sub Flows:</i>		
<i>Alternative/Exceptional Flows:</i>		

E.18 Mengakses Produk Kosong

Tabel E. 18 *Use Case Description* Mengakses Produk Kosong

<i>Use Case Name:</i> Mengakses Produk Kosong	ID: 18	<i>Importance Level:</i> High
<i>Primary Actor:</i> Kasir	<i>Use Case Type:</i> Detail, essential	
<i>Stakeholders and Interests:</i> Kasir – Ingin mengakses data produk kosong		
<i>Brief Description:</i> Use case ini menjelaskan bagaimana kasir mengakses data produk kosong		
<i>Trigger:</i> Kasir membuka halaman data produk kosong <i>Type:</i> External		
<i>Relationships:</i> <ul style="list-style-type: none"> • <i>Association:</i> Kasir • <i>Include:</i> • <i>Extend:</i> • <i>Generalize:</i> 		
<i>Normal Flows of Events:</i> <ol style="list-style-type: none"> 1) Kasir memilih <i>master</i> data produk kosong 2) Sistem menampilkan <i>master</i> data produk kosong 		
<i>Sub Flows:</i>		
<i>Alternative/Exceptional Flows:</i>		

E.19 Mengakses Informasi Produk

Tabel E. 19 *Use Case Description* Mengakses Informasi Produk

<i>Use Case Name:</i> Mengakses Informasi Produk	ID: 19	<i>Importance Level:</i> High
<i>Primary Actor:</i> Kasir	<i>Use Case Type:</i> Detail, essential	
<i>Stakeholders and Interests:</i> Kasir – Ingin mengakses data informasi produk		
<i>Brief Description:</i> Use case ini menjelaskan bagaimana kasir mengakses data informasi produk		
<i>Trigger:</i> Kasir membuka halaman informasi produk <i>Type:</i> External		
<i>Relationships:</i> <ul style="list-style-type: none"> • <i>Association:</i> Kasir • <i>Include:</i> • <i>Extend:</i> • <i>Generalize:</i> 		
<i>Normal Flows of Events:</i> <ol style="list-style-type: none"> 1) Kasir memilih <i>master</i> data informasi produk 2) Sistem menampilkan <i>master</i> data informasi produk 		
<i>Sub Flows:</i>		
<i>Alternative/Exceptional Flows:</i>		

E.20 Mengakses *Batch* Produk

Tabel E. 20 *Use Case Description* Mengakses *Batch* Produk

<i>Use Case Name:</i> Mengakses <i>Batch</i> Produk	ID: 20	<i>Importance Level:</i> High
<i>Primary Actor:</i> Kasir	<i>Use Case Type:</i> Detail, essential	
<i>Stakeholders and Interests:</i> Kasir – Ingin mengakses data <i>batch</i> produk		
<i>Brief Description:</i> Use case ini menjelaskan bagaimana kasir mengakses <i>batch</i> produk		
<i>Trigger:</i> Kasir membuka halaman <i>batch</i> produk <i>Type:</i> External		
<i>Relationships:</i> <ul style="list-style-type: none"> • <i>Association:</i> Kasir • <i>Include:</i> • <i>Extend:</i> • <i>Generalize:</i> 		
<i>Normal Flows of Events:</i> <ol style="list-style-type: none"> 1) Kasir memilih <i>master</i> data <i>batch</i> produk 2) Sistem menampilkan <i>master</i> data <i>batch</i> produk 		
<i>Sub Flows:</i>		
<i>Alternative/Exceptional Flows:</i>		