

# SYSTEM REQUIREMENT SPECIFICATION

TOKOBUILDER / SISTEM INFORMASI

LINDA JIE - C14230052

TIMOTHY SEBASTIAN ITAMURTI - C14230071

JOSEPH EVAN TANUJAYA - C14230096

MARCEL HANS SASONGKO - C14230099

EVELIN FELICIA BUNTARAN - C14230100

ADSI

## feasibility study

### **1. Pendahuluan**

#### **a. Latar Belakang**

Sumber permasalahan mitra kami adalah inkonsistensi pencatatan yang menyebabkan perhitungan ulang stok ulang berulang serta pencatatan keuangan yang manual sehingga tidak memungkinkan untuk melihat report berdasarkan suatu interval waktu. Selain masalah pencatatan, mitra kami juga tidak memiliki perlengkapan yang lengkap untuk memperluas pangsa pasar di dunia digital sehingga potensi perkembangan mitra kami terhambat. Dengan menggunakan Sistem Informasi Toko Builder, kami dapat menanggulangi masalah pencatatan keuangan dan stok dengan cara pencatatan mudah tiap terjadi penjualan lewat produk kami sehingga dapat dengan cepat mengecek stok beserta lokasinya. Kemudian untuk memperluas pangsa pasar di dunia digital, sistem informasi kami memungkinkan untuk membuat website yang dapat digunakan sebagai katalog produk beserta newsletter bagi yang berlangganan untuk membagikan flyer produk yang sedang promo atau pengenalan produk baru.

#### **b. Ruang Lingkup**

Setelah kami menginterview pemilik toko, kami dapat menyimpulkan bahwa permasalahan mereka ada pada pencatatan, keberadaan di dunia digital, beserta promosi. Pencatatan tidak dilakukan secara rutin, keberadaan di dunia digital hanya ada pada akun Tokopedia, beserta tidak mampu menangani promosi produk karena di toko ini hanya sang pemilik yang mengurus semuanya beserta dengan seorang nenek. Aplikasi kami batasi perkembangannya untuk fitur yang mudah digunakan serta user friendly karena pemilik toko dan nenek tersebut tidak bisa menggunakan sebuah software yang sangat rumit. Data yang biasanya dicari adalah penghasilan bulan ini dan sisa stok.

#### **c. Studi Kelayakan**

Berdasarkan hasil interview kami dengan pemilik toko, kami dapat menyimpulkan bahwa kemampuan pemilik toko dan nenek tersebut cukup untuk menggunakan aplikasi Sistem Informasi. Pemilik toko dapat dengan lancar menggunakan excel, visio, dan word. Nenek tersebut dapat menggunakan WhatsApp, TikTok, dan YouTube tanpa ada masalah. Toko ini juga memiliki laptop dan HP untuk menggunakan Sistem Informasi kami.

#### **d. Adoption Strategi**

Rencana kami untuk penggunaan aplikasi ini secara efektif adalah dengan melakukan pelatihan selama 1-2 hari. Pelatihan ini mencangkup bagaimana cara mencatat penjualan, mengecek stok, mengeluarkan laporan, mengeluarkan promosi, dan CRUD produk. Kemudian kami akan memberi bantuan chat support selama 3 bulan untuk mempermudah menjawab pertanyaan mereka seputar penggunaan Sistem Informasi kami.

## 2. Definisi Aktor

Aktor		Deskripsi
Kode	Nama	
KSR	Kasir	Kasir orang melayani semua transaksi yang terjadi di toko, menjalankan customer loyalty program, dan generate flyer dan PDF katalog produk berdasarkan ketersediaan produk
ADM	Admin toko	Admin mewarisi semua yang dapat dilakukan kasir, memegang master key untuk CRUD katalog produk, catatan penjualan, dan daftar supplier, generate laporan keuntungan
TMR	Timer	Timer melakukan pengecekan jumlah stok barang secara berkala, menyesuaikan dengan <i>timestamp</i> saat ini akan tenggat hutang dan piutang toko, dan membuat peringatan terhadap stok produk yang rendah atau jatuh tempo tagihan
CLT	Client	Client dapat mengakses keseluruhan daftar dan keterangan produk yang tersedia untuk dibeli di toko

## 3. Kebutuhan Fungsional

Kebutuhan fungsional adalah fitur-fitur yang perlu ada dalam suatu sistem untuk memuaskan pengguna. Untuk lebih memahami kebutuhan dari pengguna aplikasi *Toko Builder* ini kami sudah melakukan interview dengan pemilik usaha. Setelah interview ditemukan bahwa kasir ingin fitur yang membantu dalam pencatatan penjualan agar pencatatan tidak perlu dilakukan secara manual. Selain itu kasir ingin fitur yang dapat membuat sebuah katalog atau flyer baru tanpa perlu campur tangan manusia yang terlalu banyak. Tujuannya adalah supaya memiliki cara untuk melakukan promosi. Kasir juga ingin membuat daftar untuk pelanggan-pelanggan yang loyal. Untuk admin,

admin ingin fitur yang dapat memberikan peringatan ketika stok rendah dan ketika hutang atau piutang sudah mendekati hari jatuh tempo supaya tidak kelupaan. Admin juga ingin fitur untuk memanajemen vendor. Lebih lanjut, admin ingin membuat dan mengecek semua pencatatan penjualan dan pembelian secara digital agar lebih mudah diatur dan hemat kertas. Kita juga melakukan interview dengan beberapa pelanggan dan ditemukan bahwa pelanggan ingin fitur untuk melihat katalog dan stok barang yang tersedia pada waktu tertentu.

Setelah melakukan pendaftaran semua keinginan dari para pengguna, kami sebagai tim membahas lebih lanjut detail-detail penting mengenai sistem yang ingin kami bangun. Pada akhirnya kami menentukan bahwa sistem yang ingin bangun perlu melakukan hal-hal ini.

Admin

1. Admin dapat membuat dan mengecek catatan penjualan, pembelian, dan pendapatan/keuntungan
2. Admin dapat mengurus vendor dan barang yang disediakan oleh vendor
3. Admin dapat membuat, memperbarui, atau menghapus katalog barang
4. Admin akan diingatkan oleh sistem jika stok mulai rendah dan hutang atau piutang mendekati jatuh tempo

Kasir

1. Kasir dapat membuat, membaca, memperbarui, atau menghapus pelanggan dari daftar pelanggan loyal
2. Kasir dapat membuat flyer dan katalog barang
3. Kasir dapat membaca katalog dan stok produk
4. Kasir dapat membuat catatan penjualan

Pelanggan

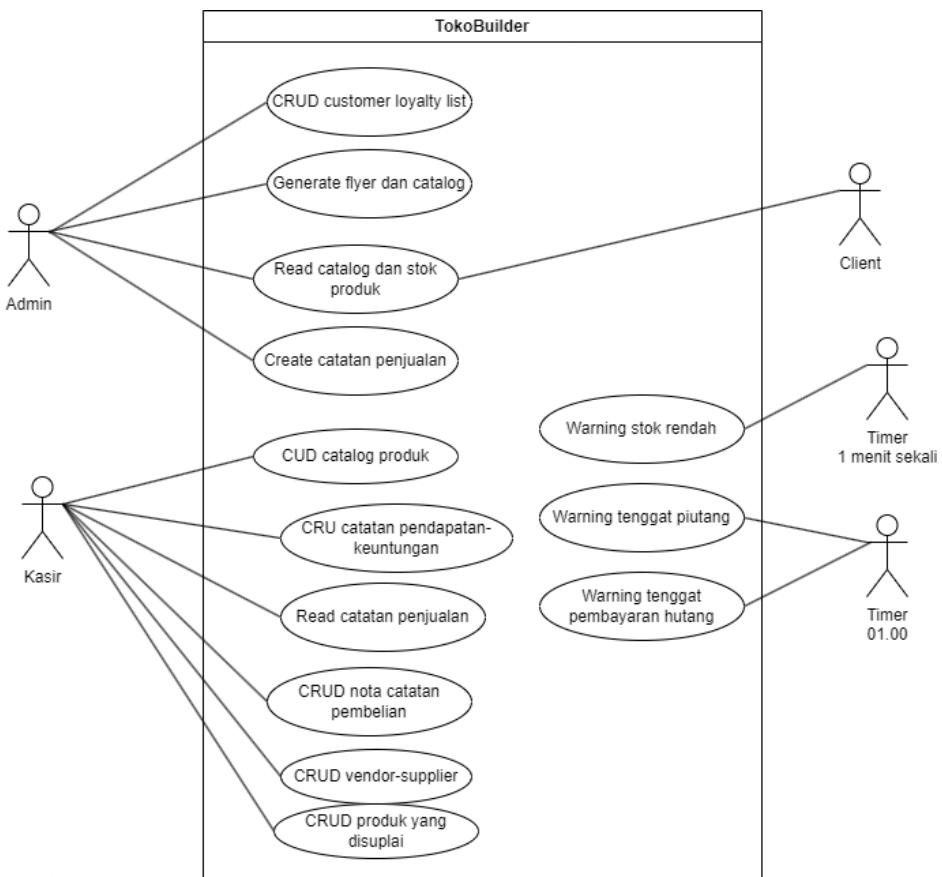
1. Pelanggan dapat membaca katalog dan stok barang yang tersedia saat ini

<b>Id</b>	<b>Id Use Case</b>	<b>Fungsi</b>	<b>Aktor</b>	<b>Deskripsi</b>
TB.01	UC_TB.01, UC_TB.02, UC_TB.12	Manajemen Customer	Kasir	Membuat daftar pelanggan loyal dan membuat flyer atau katalog yang dapat digunakan untuk promosi. Selain itu juga membantu mengingatkan jika piutang sudah dekat jatuh tempo.
TB.02	UC_TB.03, UC_TB.05, UC_TB.06	Manajemen Produk/Stok	Kasir, Admin	Membantu manajemen katalog dan stok produk. Catatan: kasir hanya dapat membaca katalog, sedangkan admin bisa membuat, memperbarui, dan menghapus katalog. Juga mengingatkan jika stok rendah.
TB.03	UC_TB.10, UC_TB.11, UC_TB.13	Manajemen Vendor	Admin	Pendaftaran vendor disertai barang yang disediakan oleh vendor tersebut. Juga peringatan jika hutang sudah dekat jatuh tempo.
TB.04	UC_TB.04, UC_TB.08, UC_TB.09	Manajemen Penjualan dan Pembelian	Kasir, Admin	Pembuatan dan pembacaan catatan penjualan dan pembelian. Catatan: untuk pembuatan catatan penjualan dilakukan oleh kasir dan pembacaan dilakukan oleh admin
TB.05	UC_TB.07	Manajemen Pendapatan	Admin	Pencatatan pendapatan/keuntungan

#### 4. Diagram UML

Bagian ini terdiri 4 bagian utama, yaitu :

##### a. Use Case



ID Usecase	Usecase	Actors			
		KSR	ADM	TMR	CLT
UC_TB.01	CRUD customer loyalty list	v	v		
UC_TB.02	Generate flyer dan catalog	v	v		
UC_TB.03	Read katalog dan stok produk	v	v	v	
UC_TB.04	Create catatan penjualan	v	v		
UC_TB.05	Warning stok rendah		v		
UC_TB.06	CUD katalog produk dan jenisnya		v		
UC_TB.07	CUD catatan pendapatan-keuntungan		v		
UC_TB.08	Read catatan penjualan		v		
UC_TB.09	CRUD nota catatan pembelian		v		
UC_TB.10	CRUD vendor/supplier		v		
UC_TB.11	CRUD produk yang disuplai		v		
UC_TB.12	Warning tenggat pembayaran piutang		v		
UC_TB.13	Warning tenggat pembayaran hutang		v		

**Timothy Sebastian Itamurti - C14230071**

Form Element	Description
Use Case Name	Warning Tenggat Piutang
Description	Sistem mengirim notifikasi lewat email untuk piutang yang belum lunas serta memasang warning di dashboard website. Dicek 01:00
Actors	Timer
Priority	Must Have: untuk mengatasi piutang yang belum lunas supaya uang bisa masuk ke toko dan disirkulasikan lagi

Risk	1. High: Jika piutang tidak segera lunas, maka toko bisa kehilangan keuntungan dan menghambat sirkulasi uang
Pre-conditions and assumptions	Piutang yang belum lunas melewati batas tanggal pelunasan
Extension Points	-
Trigger	Piutang yang belum lunas melewati batas tanggal pelunasan
Flow of Events	<ol style="list-style-type: none"> <li>1. Use case start saat <b>sistem</b> mengecek <b>status</b> semua piutang setiap hari sekali [A1]</li> <li>2. <b>Sistem</b> menemukan <b>piutang</b> yang belum lunas namun sudah lewat batas tanggal pembayaran [A2]</li> <li>3. <b>Sistem</b> mengirim email ke admin dan kasir untuk mengingatkan atas <b>piutang</b> yang belum lunas</li> <li>4. <b>Sistem</b> memasang warning di dashboard terkait <b>piutang</b> belum lunas</li> <li>5. <b>Admin</b> atau <b>kasir</b> akan mengganti <b>status</b> pembayaran menjadi lunas</li> <li>6. Use case ends</li> </ol>
Alternate Flows	A1 : Semua <b>piutang</b> yang belum lunas belum ada yang lewat batas tanggallya pembayaran lalu use case ends A2 : Ada yang belum lunas namun sudah di <b>warning</b> dan sudah dikirim email 7 hari yang lalu lalu use case ends
Post-conditions	Piutang terlunasi
Business Rules	BR1: Batas pembayaran piutang tidak boleh lebih dari 1 bulan per penjualan BR2: Piutang tidak boleh melebihi 1 juta per penjualan
Notes	Timothy Sebastian Itamurti - C14230071 UC_TB.12

Form Element	Description
Use Case Name	Warning Stok Rendah
Description	Sistem mengirim notifikasi untuk stok rendah suatu produk lewat email dan memasang warning di dashboard website. Dicek secara real time
Actors	Timer
Priority	Must Have : Untuk mempermudah pencarian stock mana yang rendah dan perlu di restock
Risk	High: Jika tidak sadar stok rendah dan tidak segera di restock, maka akan menghambat penjualan di toko sehingga menyebabkan kehilangan pembeli saat kekurangan stok
Pre-conditions and assumptions	Jumlah stok yang dibawah standar yang ditentukan untuk produk tersebut
Extension Points	-
Trigger	Jumlah stok yang dibawah standar yang ditentukan untuk produk tersebut
Flow of Events	<ol style="list-style-type: none"> <li>1. Use case start saat <b>sistem</b> membaca jumlah stok semua <b>produk</b> [A1]</li> <li>2. <b>Sistem</b> menemukan <b>produk</b> yang jumlahnya dibawah standar yang ditentukan [A2]</li> <li>3. <b>Sistem</b> mengirim <b>email</b> untuk restock <b>produk</b> tersebut</li> <li>4. <b>Sistem</b> memasang warning di <b>dashboard website</b></li> <li>5. <b>Admin</b> atau <b>kasir</b> akan memesan <b>order</b> restock</li> <li>6. Jumlah restock di masukkan ke <b>stok</b></li> <li>7. Use case ends</li> </ol>
Alternate Flows	<p>A1 : Semua <b>stok produk</b> diatas pada <b>standar</b> yang ditentukan, use case end</p> <p>A2 : Menemukan <b>stok produk</b> yang dibawah <b>standar</b> namun sudah di <b>warning</b> dan <b>warning</b> terakhir adalah kemarin</p>
Post-conditions	Order restock sudah dipesan kemudian jumlah stok akan naik setelah memasukkan pesanan restock

Business Rules	BR 1: Produk harus sudah dipesan order restocknya maksimal sehari setelah menyentuh dibawah jumlah stok yang ditentukan BR 2: Produk harus dipesan dalam jumlah yang dapat memenuhi kebutuhan 2 bulan kedepan
Non-Functional Requirements	NFR 1: Sistem harus menyala 24/7 NFR 2: Sistem harus bisa update secara real time
Notes	Timothy Sebastian Itamurti - C14230071 UC_TB.05

### Linda Jie - c14230052

Form Element	Description
Use Case Name	Create nota catatan pembelian
Description	Admin toko menginput detail pembelian produk dari tiap supplier yang terdaftar. Pendataan dilakukan per nota pembelian. Data pembelian yang disimpan berupa list produk, kuantitas produk, pembayaran, status pembayaran, dan status pengiriman.
Actors	Primary: Admin toko secondary: -
Priority	Must have: Diperlukan sebagai bagian dari pencatatan toko, dan untuk kepentingan update katalog dan stok produk.
Risk	Medium: kompleksitas rincian dan detail nota pembelian yang perlu di-input admin.
Pre-conditions and assumptions	Setidaknya ada satu vendor/supplier yang terdaftar dalam sistem sebagai tempat pembelian barang dilakukan yang tercatat pada nota pembelian. Setidaknya ada satu produk yang terdaftar dalam sistem sebagai produk yang dijual di toko.
Extension Points	EP1: create data vendor/supplier EP2: create data produk
Trigger	Admin toko melakukan pembelian produk untuk stok barang toko.

Flow of Events	<p>1: Mulai ketika admin telah melakukan pembelian/pemesanan barang dan pembelian ingin ditambahkan ke dalam database.</p> <p>1.1: Sistem retrieve list nota pembelian dari database</p> <p>2: Admin memilih add nota pembelian</p> <p>3: Admin menginput data pembelian berupa nomor faktur, tanggal, jenis pembayaran pada form kelengkapan nota pembelian</p> <p>3.1: Sistem retrieve list supplier</p> <p>3.2: Admin memilih vendor/supplier asal pembelian dilangsungkan dari list vendor/supplier [A1] [A2]</p> <p>4: Admin memilih tambah produk yang dibeli sebagai item nota</p> <p>4.1: Sistem retrieve list produk yang dijual di toko</p> <p>4.2: Admin memilih produk dari list produk [A3]</p> <p>4.3: Admin menginput harga satuan dan kuantitas produk</p> <p>4.4: Sistem menghitung keuntungan per satuan penjualan produk berdasarkan harga jual yang terdaftar [A5]</p> <p>4.5: Admin menyimpan data item pembelian [A6]</p> <p>5: Admin memilih simpan data nota pembelian [A8]</p> <p>5.1: Sistem menambahkan nota pembelian ke dalam database</p> <p>6: Use case berakhir</p>
Alternate Flows	<p>A1: Data vendor/supplier tidak ditemukan dan belum terdaftar, masuk EP1</p> <p>A2: Data vendor/supplier tidak sesuai [A8]</p> <p>A3: Produk belum terdaftar pada sistem, masuk EP2</p> <p>A4: Admin mengubah harga jual produk, masuk A8</p> <p>A5: Berdasarkan BR1, sistem menginformasikan admin akan potensi kerugian, pergi ke A4</p> <p>A6: Input-an data nota pembelian tidak disimpan, data dihapus</p> <p>A7: Admin keluar dari halaman form input data nota pembelian, use case berakhir [A9]</p> <p>A8: Input-an data nota pembelian disimpan sementara dalam program, dan dapat ditampilkan kembali pada fields form yang sama ketika vendor/supplier yang sama dipilih kembali selama program belum ditutup.</p> <p>At any time: Admin membatalkan use case, use case berakhir [A6]</p>
Post-conditions	Sebuah nota pembelian ditambahkan.

Business Rules	BR1: Harga beli satuan tidak boleh lebih dari harga jual satuan produk yang terdaftar BR2: Minimal 1 satuan produk ditambahkan pada setiap kuantitas item yang diinput
Non-Functional Requirements	NFR1: UI penjualan harus lengkap dengan fitur yang relevan
Notes	Linda Jie - c14230052 UC_TB.09

Form Element	Description
Use Case Name	Update status pembayaran dan pengiriman nota catatan pembelian
Description	Admin meng-update perkembangan status pembayaran dan status pengiriman pada setiap nota pembelian.
Actors	Primary: Admin toko Secondary: -
Priority	Must have: Diperlukan sebagai bagian dari pendataan toko, dan untuk kepentingan update katalog dan stok produk.
Risk	Low: update status pembayaran dan status pengiriman memiliki persyaratan yang sederhana
Pre-conditions and assumptions	Setidaknya terdapat satu nota pembelian yang telah diinput
Extension Points	-

Trigger	Admin membayar hutang untuk nota dengan status pembayaran unpaid atau partial. Pengiriman barang telah diselesaikan untuk item pembelian dengan status pengiriman on process.
Flow of Events	<p>1: Mulai ketika terjadi pembaruan status pembayaran atau pengiriman.</p> <p>1.1: Sistem retrieve list nota pembelian dari database</p> <p>2: Admin memilih nota pembelian yang membutuhkan pembaruan status</p> <p>Update status pembayaran:</p> <p>3: Admin memperbarui status pembayaran pada nota pembelian</p> <p>3.1: Admin memasukan nominal hutang yang telah dibayarkan untuk suatu nota.</p> <p>3.2: admin memilih save changes</p> <p>3.3: Sistem memperbarui rincian nominal pembayaran sesuai BR3</p> <p>3.4: Sistem memperbarui status pembayaran nota berdasarkan BR1.</p> <p>Update status pengiriman:</p> <p>3: Admin memperbarui status pengiriman pada nota pembelian</p> <p>3.1: Pada status pengiriman, admin mengisi kuantitas item tertentu [A1]</p> <p>3.2: Admin memilih save changes</p> <p>3.3: Status pengiriman item berubah dari <i>pending</i> menjadi <i>delivered</i> terhadap item pembelian yang telah dikirim ke toko, sesuai BR2</p> <p>3.4: Sistem melakukan penyesuaian stok produk toko dengan jumlah item pembelian yang berstatus delivered</p> <p>4: Admin keluar dari list item nota</p> <p>end use case</p>
Alternate Flows	<p>A1: Terdapat item <i>delivered</i> yang perlu direturn, admin memilih return pada item yang dimaksud [A3]</p> <p>A2: Status pengiriman pada sejumlah kuantitas item tersebut berubah kembali menjadi <i>pending</i>, sesuai BR2</p> <p>A3: Admin mengisi kuantitas item yang direturn [A4]</p> <p>A4: Admin memilih save update quantity [A5][A6]</p> <p>A5: Sistem mengurangi kuantitas item pada nota pembelian dengan jumlah item yang direturn,</p>

	Sistem menyesuaikan rincian nominal pada nota pembelian sesuai BR3 A6: admin cancel updating [A7] A7: Sistem clear form return
Post-conditions	Status pembayaran nota pembelian dan status pengiriman item yang dibeli pada nota pembelian telah diperbarui.
Business Rules	BR1: Status pembayaran <i>unpaid</i> ketika ada nominal pembelian belum dibayar dan tersimpan dalam tabel Hutang database dan ketika tidak ada lagi nominal yang harus dibayarkan, hutang nota pada tabel Hutang dihapus BR2: Status pengiriman pada masing-masing item nota pembelian berupa <i>pending</i> ketika terdapat satu atau lebih kuantitas yang belum diterima toko dan <i>Delivered</i> ketika seluruh kuantitas sebuah item sudah diterima BR3: Nominal total pembelian dikurangi nominal yang telah dibayar menghasilkan nominal yang belum dibayar
Non-Functional Requirements	NFR1: UI yang user friendly untuk mempermudah melihat detail nota
Notes	Linda Jie - c14230052 UC_TB.09

Form Element	Description
Use Case Name	Warning Tenggat Hutang
Description	Sistem menampilkan notifikasi pengingat tenggat pembayaran nota yang mendekati jatuh tempo pada dashboard software
Actors	Primary: Timer Secondary: Admin toko
Priority	Must have: untuk membantu manajemen toko
Risk	Medium: sistem memberi peringatan dengan mencocokan tanggal jatuh tempo nota dengan real time

Pre-conditions and assumptions	Setidaknya terdapat satu nota pembelian yang telah diinput dan memiliki status pembayaran unpaid atau partial.
Extension Points	EP1: Update nota pembelian
Trigger	Status pembayaran unpaid atau partial pada nota pembelian barang dengan tenggat pembayaran kurang dari 7 hari.
Flow of Events	1: Mulai ketika sistem retrieve list nota pembelian yang berstatus pembayaran belum lunas setiap hari sekali [A1] 2: Sistem triggered ketika dilakukan update terhadap table hutang dan memberikan peringatan sesuai BR1 3: Sistem menampilkan notifikasi pengingat pada dashboard software SI yang berisi informasi nota pembelian, status pembayaran nota, nominal yang belum dibayar, dan jumlah hari menuju tenggat pembayaran hutang [A2] end use case
Alternate Flows	A1: Tidak ada pembayaran yang belum lunas, use case ends A2: Peringatan lewat batas tenggat pembayaran pada nota yang belum dibayar dan telah melewati batas pembayaran hutang (Hari kurang dari 0)
Post-conditions	Notifikasi pengingat pembayaran hutang tampil pada halaman dashboard software dan pinned pada kolom notifikasi.
Business Rules	BR1: Pengingat dimulai hari H tenggat pembayaran hutang
Non-Functional Requirements	NFR1: Sistem tersedia 24/7 NFR2: Sistem harus bisa update secara real time
Notes	Linda Jie - c14230052 UC_TB.13

<b>Form Element</b>	<b>Description</b>
Use Case Name	Create Customer Loyalty List
Description	Use case ini menangani penambahan pelanggan baru ke dalam loyalty list jika belum terdaftar.
Actors	Primary : admin, kasir
Priority	Tinggi = memastikan katalog produk yang benar dan up-to-date,yang mempengaruhi penjualan,inventaris, dan kepuasan pelanggan.
Risk	Medium: Jika ada kesalahan input atau duplikasi data, program loyalitas tidak berjalan dengan baik.
Pre-conditions and assumptions	Customer belum ada di database, dan data customer harus lengkap dan valid.
Extension Points	Cek duplikat data, validasi format data
Trigger	Ketika admin/kasir ingin menambahkan customer baru ke daftar customer loyalty.
Flow of Events	<ol style="list-style-type: none"> <li>1. Admin/kasir menginput data customer.</li> <li>2. Sistem memverifikasi data customer.(BR1)</li> <li>3. Sistem memeriksa apakah data customer sudah ada dalam database.[A1](BR2)</li> <li>4. Sistem melakukan konfirmasi untuk menambahkan customer.</li> </ol>
Alternate Flows	A1: Jika ada duplikat, sistem memberikan notifikasi bahwa customer sudah ada dan kembali ke halaman input customer.
Post-conditions	Customer berhasil didaftarkan ke dalam loyalty list.
Business Rules	BR1 : Data customer harus lengkap dan sesuai dengan ketentuan sistem. BR2 : Customer tidak bisa didaftarkan jika customer sudah ada di dalam database
Non-Functional Requirements	NFR 1: Sistem harus bisa melayani 5 user sekaligus NFR 2: Sistem harus bisa update secara real time

Notes	Sistem menyarankan Admin/Kasir memeriksa database sebelum mendaftarkan customer baru.  UC_TB.01 Evelin Felicia Buntaran - c14230100
-------	--

Form Element	Description
Use Case Name	Update customer loyalty list
Description	Use case ini menangani pembaruan detail pelanggan yang sudah ada di loyalty list.
Actors	Primary : admin, kasir
Priority	Tinggi: Data customer harus selalu up to date dalam loyalty list.
Risk	Medium: Jika data tidak diperbarui dengan benar maka akan menyebabkan kebingungan dan data pelanggan yang sudah tidak valid.
Pre-conditions and assumptions	Pelanggan sudah ada di database. Data yang diinput harus lengkap dan terverifikasi.
Extension Points	Validasi format data
Trigger	Ketika admin/kasir ingin mengupdate data customer di customer loyalty list.
Flow of Events	<ol style="list-style-type: none"> <li>Admin/kasir menginput data customer.</li> <li>Sistem memverifikasi data.(BR1)</li> <li>Sistem memeriksa apakah customer sudah ada dalam database.[A1](BR2)</li> <li>Sistem mengupdate data customer.</li> <li>Sistem mengkonfirmasi bahwa update berhasil dengan menampilkan notifikasi bahwa data sudah berhasil diupdate.</li> </ol>
Alternate Flows	A1 : Jika tidak ada, sistem memberikan notifikasi bahwa customer tidak ditemukan dan kembali ke halaman input data customer.

Post-conditions	Informasi customer berhasil diperbarui dan muncul notifikasi bahwa informasi customer sudah diperbarui atau jika tidak ada, muncul notifikasi jika customer tidak ada.
Business Rules	BR1 : Data customer harus lengkap dan sesuai dengan ketentuan sistem. BR2: Semua bagian yang diupdate harus diperiksa untuk validasi, untuk memastikan formatnya benar dan tidak salah input
Non-Functional Requirements	NFR 1 : Wajib respon dalam 3 detik jika tidak harus di time out NFR 2 : Sistem harus bisa update secara real time.
Notes	Sistem menyarankan Admin/Kasir memeriksa database sebelum mengupdate data customer.  UC_TB.01 Evelin Felicia Buntaran - c14230100

Form Element	Description
Use Case Name	Delete Customer Loyalty List
Description	Use case ini menangani deletion data pelanggan yang sudah ada di loyalty list.
Actors	Primary : admin, kasir
Priority	Tinggi: Memastikan data pelanggan yang tidak diperlukan atau sudah tidak relevan dihapus dari daftar loyalitas.
Risk	Medium: Risiko penghapusan data pelanggan yang penting secara tidak sengaja.
Pre-conditions and assumptions	Pelanggan sudah ada di database.
Extension Points	-
Trigger	Admin/kasir memilih pelanggan untuk dihapus dari daftar loyalitas.

Flow of Events	<ol style="list-style-type: none"> <li>1. Admin/kasir memilih data customer yang akan dihapus.</li> <li>2. Sistem memeriksa apakah customer ada dalam database.(BR1)[A1]</li> <li>3. Sistem menghapus data customer.(BR3)</li> <li>4. Sistem mengkonfirmasi bahwa data yang sudah dipilih akan benar benar dihapus.(BR2)</li> <li>5. Sistem mengkonfirmasi bahwa deletion berhasil dengan menampilkan notifikasi bahwa data sudah berhasil dihapus.</li> </ol>
Alternate Flows	A1 : Jika tidak ada, sistem memberikan notifikasi bahwa customer tidak ditemukan.
Post-conditions	Pelanggan dihapus dari daftar loyalitas dan muncul notifikasi bahwa pelanggan sudah berhasil dihapus.
Business Rules	<p>BR1 : Pelanggan harus sudah ada di dalam database sebelum dapat dihapus.</p> <p>BR2 : Sistem harus mengkonfirmasi dulu dengan admin bahwa data benar benar akan dihapus untuk menghindari ketidaksengajaan</p> <p>BR3: Penghapusan data bersifat permanen</p>
Non-Functional Requirements	<p>NFR 1: Sistem harus bisa melayani 5 user sekaligus</p> <p>NFR 2: Sistem harus bisa update secara real time</p>
Notes	<p>Sistem harus meminta konfirmasi sebelum melanjutkan penghapusan untuk mencegah penghapusan yang tidak disengaja.</p> <p>UC_TB.01</p>

Form Element	Description
Use Case Name	Create Catatan pendapatan keuntungan
Description	Use case ini menangani pencatatan data pendapatan dan keuntungan baru oleh admin atau kasir.
Actors	Primary : Admin, kasir

Priority	Tinggi : Wajib ada karena digunakan untuk memastikan data pendapatan/keuntungan tercatat dengan benar.
Risk	Medium = resiko kesalahan input data atau format yang tidak sesuai.
Pre-conditions and assumptions	User memiliki izin untuk mencatat data pendapatan.
Extension Points	Notifikasi kesalahan input data.
Trigger	Admin atau kasir menginput data pendapatan/keuntungan.
Flow of Events	<p>1. Sistem memverifikasi apakah user memiliki izin untuk menginput data(BR2)[A2]</p> <p>2. Pengguna memulai proses pencatatan pendapatan/keuntungan.</p> <p>3. Sistem menampilkan formulir untuk input data.</p> <p>4. Pengguna mengisi data pendapatan/keuntungan.</p> <p>5. Sistem memvalidasi data yang diinput.[A1](BR1)</p> <p>6. Sistem menyimpan data ke database.</p> <p>7. Sistem memunculkan notif bahwa pencatatan berhasil.</p>
Alternate Flows	<p>A1 : Jika data tidak lengkap atau salah, sistem memberikan notifikasi kesalahan dan mengarahkan user untuk menginput ulang data.</p> <p>A2 : Jika user tidak memiliki izin maka akan muncul notifikasi dan tidak bisa lanjut ke langkah selanjutnya.</p>
Post-conditions	Data pendapatan/keuntungan tersimpan dalam database.
Business Rules	<p>BR1 : Data yang diinput harus lengkap dan terverifikasi.</p> <p>BR2 : Hanya pengguna yang memiliki izin bisa input data</p>
Non-Functional Requirements	<p>NFR 1: Wajib respon dalam 3 detik jika tidak harus di time out</p> <p>NFR 2: Sistem harus bisa melayani 5 user sekaligus</p> <p>NFR 3: Sistem harus bisa update secara real time.</p>
Notes	<p>Sistem menyimpan log penghapusan untuk referensi di kemudian hari.</p> <p>UC_TB.07</p>

Form Element	Description

Use Case Name	Read Catatan pendapatan keuntungan
Description	Use case ini menangani read laporan pendapatan dan keuntungan berdasarkan data yang tersimpan di database.
Actors	Primary : Admin, Kasir
Priority	Tinggi : Wajib ada untuk memastikan laporan pendapatan dapat diakses secara efisien.
Risk	Rendah = resiko laporan yang tidak lengkap jika ada data yang belum terinput, maka dari itu user harus bisa melihat data.
Pre-conditions and assumptions	Pengguna memiliki akses untuk membaca laporan pendapatan.
Extension Points	Notifikasi jika user tidak memiliki izin untuk akses.
Trigger	Admin atau kasir meminta laporan pendapatan.
Flow of Events	<ol style="list-style-type: none"> <li>1. Pengguna memulai permintaan untuk melihat laporan pendapatan.</li> <li>2. Sistem memverifikasi izin pengguna.[A1](BR1)</li> <li>3. Sistem mengambil data pendapatan dari database.</li> <li>4. Sistem memproses data pendapatan dan keuntungan.</li> <li>5. Sistem menghasilkan laporan pendapatan/keuntungan.</li> <li>6. Sistem menampilkan laporan kepada pengguna.(BR2)[A2]</li> </ol>
Alternate Flows	<p>A1 : Jika user tidak memiliki izin untuk melihat data maka akan memunculkan notifikasi dan tidak bisa lanjut ke langkah berikutnya.</p> <p>A2 : Jika user mengklik ingin ubah format ke bentuk excel, atau pdf maka akan langsung mendownload data yang sedang ditunjukkan ke dalam bentuk pdf atau excel</p>
Post-conditions	Laporan pendapatan berhasil ditampilkan.

Business Rules	BR1 : User harus memiliki akses untuk melihat data BR2 : Laporan bisa diexport ke dalam bentuk excel atau pdf
Non-Functional Requirements	NFR 1 : Sistem harus menyala 24/7  NFR 2 : Wajib respon dalam 3 detik jika tidak harus di time out  NFR 3 : UI yang user friendly untuk mempermudah melihat detail nota
Notes	Sistem harus memberikan peringatan kepada pengguna jika data yang diperlukan untuk laporan tidak tersedia.  UC_TB.07

Form Element	Description
Use Case Name	Update Catatan pendapatan keuntungan
Description	Use case ini menangani pembaruan data pendapatan atau keuntungan yang sudah ada di sistem.
Actors	Primary : Admin, kasir
Priority	Tinggi : Wajib ada karena digunakan untuk memastikan data pendapatan/keuntungan diperbarui secara akurat dan tetap relevan.
Risk	Medium = risiko ketidaklengkapan data atau kesalahan dalam penghitungan ulang.
Pre-conditions and assumptions	User memiliki izin untuk mengakses data pendapatan.
Extension Points	Notifikasi kesalahan atau perubahan data yang terkait. Misal format salah.
Trigger	User memilih untuk memperbarui data pendapatan/keuntungan.
Flow of Events	<ol style="list-style-type: none"> <li>1. Sistem memastikan user memiliki izin untuk mengakses data.[A2](BR3)</li> <li>2. Pengguna memilih record pendapatan yang akan diperbarui.</li> </ol>

	<ul style="list-style-type: none"> <li>3. Sistem menampilkan record yang dipilih untuk diperbarui.</li> <li>4. Pengguna mengubah data yang diperlukan.</li> <li>5. Sistem memvalidasi data baru yang dimasukkan.[A1](BR1)(BR2)</li> <li>6. Sistem memeriksa apakah ada data lain yang terpengaruh oleh perubahan.</li> <li>7. Jika tidak ada masalah, sistem memperbarui data di database.</li> <li>8. Sistem menghitung ulang data yang telah berubah dan memberikan konfirmasi.</li> </ul>
Alternate Flows	<p>A1 : Jika data tidak lengkap atau salah, sistem memberikan notifikasi kesalahan dan mengarahkan user untuk menginput ulang data.</p> <p>A2 : Jika user tidak memiliki izin untuk melihat data maka akan memunculkan notifikasi dan tidak bisa lanjut ke langkah berikutnya.</p>
Post-conditions	Data pendapatan/keuntungan diupdate dalam database.
Business Rules	<p>BR1 : Data yang diperbarui harus divalidasi ulang dan tidak boleh memengaruhi data lain yang relevan secara negatif.</p> <p>BR2 : Data yang diinput harus lengkap dan sesuai format</p> <p>BR3 : Hanya user yang memiliki izin bisa melihat dan mengupdate data.</p>
Non-Functional Requirements	<p>NFR 1: Sistem harus bisa melayani 5 user sekaligus</p> <p>NFR 2 : Wajib respon dalam 3 detik jika tidak harus di time out</p> <p>NFR 3 : Sistem harus bisa update secara real time</p>
Notes	<p>Sistem harus memberikan notifikasi kepada pengguna jika ada kesalahan dalam input data dan jika ada perubahan signifikan yang mempengaruhi record lain.</p> <p>UC_TB.07</p>

**Marcel Hans Sasongko - c14230099**

<b>Form Element</b>	<b>Description</b>
Use Case Name	CRUD Vendor-Supplier
Description	Admin bisa memeriksa, menambahkan Vendor, Menghapus Vendor, Meng-Update Vendor
Actor(s)	Admin
Priority	Harus Ada : Essential to this system
Risk	High: Karena jika tidak ada Vendor-Supplier, Maka tidak ada yang di jual
Trigger	Ketika admin ingin memeriksa, menambahkan, menghapus, meng-update Vendor/Supplier
Pre-Conditions	Vendor harus terdaftar (Minimal 1) dalam sistem ini untuk dapat melihat, menghapus, meng-update, atau menambahkan Vendor dengan vendor-supplier baru
Post-Conditions	Sudah ditambahkan Vendor/Supplier Sudah dikurangi Vendor/Supplier Sudah di update menjadi Alamat atau nama Vendor – Supplier
Flow of Event	<p>Create Vendor-Supplier</p> <ol style="list-style-type: none"> <li>1. Admin harus memulai menambah Vendor-Supplier</li> <li>2. Admin dapat mengakses halaman tambah Vendor</li> <li>3. Admin mengisi nama Vendor, Alamat vendor sesuai dengan Vendornya [BR1][BR2][A1]</li> <li>4. Admin bisa langsung menekan Save apabila formnya terisi semua</li> <li>5. End case</li> </ol> <p>Read Vendor-Supplier</p> <ol style="list-style-type: none"> <li>1. Admin harus memulai melihat Vendor-Supplier</li> <li>2. Admin dapat mengakses halaman Vendor</li> <li>3. Admin dapat memantau atau melihat Vendor manapun saja</li> <li>4. Admin dapat melakukan “Back” untuk Kembali ke halaman utama</li> <li>5. End case</li> </ol> <p>Update Vendor-Supplier</p> <ol style="list-style-type: none"> <li>1. Admin harus memulai memperbarui Vendor-Supplier</li> <li>2. Admin dapat mengakses halaman Vendor</li> <li>3. Admin mengisi yang sudah ada nama Vendor atau Alamat Vendor yang salah. Lalu mengisi Alamat Vendor atau nama Vendor yang benar (BR1)(BR)(A1)</li> </ol>

	<p>4. Admin bisa langsung menekan Save apabila formnya terisi semua</p> <p>5. End case</p> <p>Delete Vendor-Supplier</p> <ol style="list-style-type: none"> <li>1. Admin harus memulai menghapus Vendor-Supplier</li> <li>2. Admin dapat mengakses halaman Vendor</li> <li>3. Admin bisa memilih vendor mana yang sekiranya tidak cocok, Lalu tekan delete [A2]</li> <li>4. Admin bisa langsung menekan Save</li> <li>5. End case</li> </ol>
Non-Functional Requirement	NFR 1: Sistem harus bisa update secara real time NFR 2: UI yang user friendly untuk mempermudah melihat detail nota
Notes	UC_TB.11 Marcel Hans Sasongko - c14230099
Alternate Flow	A1: Jika mengisi formnya tidak lengkap, akan muncul pop up "Lengkapi form". pergi ke step Create Vendor step 3 A2: Jika tidak jadi dihapus, Maka akan kembali ke halaman Vendor. Pergi ke step Delete Vendor step 3
Business Rules	BR1: Harus diisi semua form Vendor-Supplier BR2: Jika ingin mengisi Alamat vendor harus sesuai Maps

Marcel Hans Sasongko - c14230099

Form Element	Description
Use Case Name	CRUD Produk yang Disuplai Vendor
Description	<b>Admin</b> bisa memeriksa, menambahkan, menghapus <b>produk</b> yang disuplai oleh <b>Vendor-Supplier</b>
Actor(s)	<b>Admin</b>
Priority	Bisa Ada : Essential to this system
Risk	Medium: Karena jika adminnya lupa membeli produknya, Bisa melihat system fitur ini
Trigger	Ketika admin ingin memeriksa, menambahkan, menghapus, meng-update Produk yang disuplai oleh vendor maupun supplier
Pre-Conditions	Produk Harus terdaftar (Minimal 1) dalam sistem ini untuk dapat melihat, menghapus, meng-update, atau menambahkan produk baru dengan vendor-supplier baru
Post-Conditions	Sudah ditambahkan produk Baru Sudah dihapus produknya Sudah di update menjadi produk dan vendor mana saja
Flow of Event	<p>Create <b>Produk</b> yang Disuplai <b>Vendor</b></p> <ol style="list-style-type: none"> <li>1. <b>Admin</b> bisa memulai menambahkan <b>produk</b> dan <b>vendor</b> dari mana saja</li> <li>2. <b>Admin</b> mengakses halaman tambah <b>produk</b></li> <li>3. <b>Admin</b> dapat mengisi halaman <b>formulir</b> (Nama, deskripsi, stok, kategori dan lain sebagainya) (BR1)(A1)</li> <li>4. <b>Admin</b> dapat menekan Save, apabila sudah <b>formulir</b> tersebut sudah di isi semua</li> <li>5. End case</li> </ol> <p>Read <b>Produk</b> yang Disuplai <b>Vendor</b></p> <ol style="list-style-type: none"> <li>1. <b>Admin</b> bisa memulai melihat <b>produk</b> dan <b>vendor</b> dari mana saja</li> <li>2. Admin mengakses halaman <b>daftar produk</b> yang disuplai</li> <li>3. <b>Sistem</b> akan menampilkan <b>produk</b> yang terdaftar dan <b>vendornya</b> dari mana saja</li> <li>4. <b>Admin</b> juga dapat mencari atau memfilter berdasarkan nama <b>produk</b> atau kategori <b>produk</b></li> <li>5. <b>Admin</b> dapat menekan "back" apabila sudah <b>selesai</b> melihat <b>produk</b> dan vendornya</li> <li>6. End case</li> </ol> <p>Update <b>Produk</b> yang disuplai <b>Vendor</b></p>

	<ol style="list-style-type: none"> <li>1. <b>Admin</b> bisa memulai mengubah <b>informasi produk</b></li> <li>2. <b>Admin</b> bisa mengakses halaman <b>daftar produk</b> yang disuplai</li> <li>3. <b>Admin</b> mengubah informasi produk (misalnya harga, deskripsi, dan lain lain)</li> <li>4. <b>Admin</b> dapat menekan “Update” bila sudah selesai diubah</li> <li>5. End case</li> </ol> <p>Delete <b>Produk</b> yang disuplai <b>Vendor</b></p> <ol style="list-style-type: none"> <li>1. <b>Admin</b> bisa memulai menghapus <b>informasi produk</b></li> <li>2. <b>Admin</b> bisa mengakses <b>daftar produk</b> yang disuplai</li> <li>3. <b>Admin</b> memilih <b>produk</b> yang dihapus dan menekan “Delete”</li> <li>4. <b>Sistem</b> akan meminta konfirmasi penghapusan [A2]</li> <li>5. Setelah dikonfirmasi, <b>produk</b> yang dihapus tadi akan menghilang</li> <li>6. <b>Admin</b> dapat menekan “Back” bila sudah selesai penghapusan</li> <li>7. End case</li> </ol>
Non-Functional Requirement	NFR 1: Sistem harus bisa update secara real time NFR 2: UI yang user friendly untuk mempermudah melihat detail nota
Notes	Jika lupa beli produk di mana, bisa melihat system ini UC_TB.10 Marcel Hans Sasongko - c14230099
Alternate Flow	A1: jika <b>formulir</b> tidak dilengkapi, Maka akan muncul pop-up. pergi ke step Create <b>Produk</b> step 3 A2: Jika tidak jadi dihapus, Maka akan kembali ke halaman Vendor. Pergi ke step Delete Produk step 4
Business Rules	BR1: Harus di isi semua <b>formulir</b> produk <b>Vendor-Supplier</b>

**Joseph Evan Tanujaya - c14230096**

Form Element	Description
Use Case Name	Generate Flyer dan Catalog
Description	Kasir dapat melakukan pembuatan flyer dan katalog barang dengan tujuan untuk promosi produk kepada pelanggan. Selain itu kasir dapat memeriksa stok produk yang ada. Pembuatan flyer dan katalog barang akan disesuaikan dengan stok saat ini
Actor(s)	Primary: Kasir
Priority	Tinggi: karena pengalaman pelanggan toko bangunan akan terpengaruhi jika fitur tidak ada
Risk	Sedang: karena kasir harus memiliki kemampuan untuk menganalisis stok barang dengan teliti agar dapat membuat flyer dan katalog barang yang sesuai
Trigger	Terjadi ketika kasir ingin membuat flyer baru atau memperbarui katalog barang yang lama
Extension Point	Sebelum membuat flyer atau katalog dapat melakukan pengecekan stok barang saat ini
Pre-condition	Barang harus tersedia minimal 1 di stok
Post-condition	Terbuat flyer atau katalog baru
Non-Functional Requirement	NFR1 (Banyak users sekaligus) NFR2 (Kecepatan respon UI) NFR3 (Format Flyer dan Katalog)
Notes	Kalau perusahaan semakin berkembang, dapat membuat fitur untuk memberikan rekomendasi produk untuk pembuatan flyer agar tidak perlu mengandalkan analisis manual  UC_TB.02 Joseph Evan Tanujaya / c14230096
Flow of Events	1: Use case dimulai ketika kasir ingin membuat flyer atau katalog barang yang baru 2: Kasir membuat flyer [2.1] atau katalog [2.2] baru

	<p>2.1: Flyer, pembuatan sebuah flyer harus mengikuti BR2</p> <p>2.1.1: Extension Point (read katalog dan stok produk)</p> <p>2.1.2: Kasir memasukkan barang yang ingin dijadikan bahan flyer</p> <p>2.1.3: Kasir menentukan promosi apa yang ingin dibuat</p> <p>2.1.4: Flyer berhasil dibuat [A1][A2]</p> <p>2.1.5: Hasil flyer disimpan dalam bentuk .jpg [A5][A6]</p> <p>2.2: Katalog, pembuatan katalog harus sesuai dengan BR3</p> <p>2.2.1: Extension Point (read katalog dan stok produk)</p> <p>2.2.2: Membuat katalog baru [A3][A4]</p> <p>3: Mengirim Flyer melalui email</p> <p>4: Use case ends</p>
Alternate Flow	<p>A1: Flyer yang dibuat sesuai BR1, lanjut ke step 2.1.5</p> <p>A2: Admin kurang setuju dengan flyer, use case ends</p> <p>A3: Katalog baru berhasil dibuat, use case ends</p> <p>A4: Dalam pembuatan katalog baru terjadi kesalahan, pergi ke step 2</p> <p>A5: Flyer disimpan dulu, use case ends</p> <p>A6: Flyer ingin dikirim ke target customer sekarang, pergi ke step 3</p>
Business Rules (BR)	<p>BR1: Flyer yang dibuat harus disetujui oleh admin</p> <p>BR2: Jika ingin menjadikan barang menjadi bahan flyer, perlu dipastikan barang tersedia di stok</p> <p>BR3: Barang yang akan dimasukkan ke dalam stok sudah harus tersedia secara fisik di gudang, tidak boleh barang masih dalam pengantaran.</p>

Form Element	Description
Use Case Name	Create Catatan Penjualan
Description	Kasir dapat membuat catatan penjualan berdasarkan interval 1 hari, 1 minggu, 1 bulan, atau 1 tahun. Total penjualan akan dikalkulasi oleh system dan hasilnya akan disimpan dalam bentuk pdf atau excel.
Actor(s)	Primary: Kasir
Priority	Tinggi: karena berhubungan dengan keuangan dan penghasilan

Risk	Sedang: perhitungan tidak terlalu kompleks sehingga sistem dapat menyelesaiannya dengan mudah
Trigger	Terjadi setiap kali kasir ingin membuat sebuah catatan penjualan
Pre-condition	Perlu terjadi minimal 1 kali penjualan
Post-condition	Terbuat catatan penjualan per hari, per minggu, per bulan, atau per tahun
Non-Functional Requirement	NFR1 (Banyak users sekaligus) NFR2 (Kecepatan respon UI) NFR3 (Kalkulasi harga) NFR4 (UI penjualan) NFR5 (UI user friendly)
Notes	Mengembangkan cara untuk melakukan catatan penjualan tanpa perlu intervensi manusia  UC_TB.04 Joseph Evan Tanujaya / c14230096
Flow of Events	1: Use case dimulai ketika kasir ingin membuat catatan penjualan 2: Kasir menentukan jenis catatan penjualan [A1][A2][A3][A4] 3: Simpan hasil catatan penjualan [A9][A10] 4: Use case ends
Alternate Flow	A1: System membuat catatan penjualan per hari, pergi ke step 3 [A5] A2: Membuat catatan penjualan per minggu, pergi ke step 3 [A6] A3: Membuat catatan penjualan per bulan, pergi ke step 3 [A7] A4: Membuat catatan penjualan per tahun, pergi ke step 3 [A8] A5: Catatan penjualan yang dibuat melanggar BR1, kembali ke step 2, berikan peringatan kepada user A6: Catatan penjualan yang dibuat melanggar BR2, kembali ke step 2, berikan peringatan kepada user A7: Catatan penjualan yang dibuat melanggar BR3, kembali ke step 2, berikan peringatan kepada user

	A8: Catatan penjualan yang dibuat melanggar BR4, kembali ke step 2, berikan peringatan kepada user A9: Simpan dalam bentuk PDF, lanjut ke step 4 A10: Simpan dalam bentuk excel, lanjut ke step 4
Business Rules (BR)	BR1: Catatan penjualan per hari hanya boleh dibuat diatas jam 4 BR2: Catatan penjualan per minggu hanya boleh dibuat pada hari jumat atau sabtu BR3: Catatan penjualan per bulan hanya boleh dibuat pada tanggal 25 ke atas BR4: Catatan penjualan per tahun hanya boleh dibuat pada bulan desember tanggal 20 ke atas

Form Element	Description
Use Case Name	Buat Katalog Produk dan Jenisnya
Description	Menangani penambahan
Actors	Primary : Admin
Priority	Must Have = memastikan katalog produk yang benar dan up-to-date,yang mempengaruhi penjualan,inventaris, dan kepuasan pelanggan.
Risk	Sedang = risiko kesalahan input data produk.
Pre-conditions and assumptions	Pengguna memiliki izin untuk membuat katalog produk.
Extension Points	Notifikasi kesalahan input data.
Trigger	Admin menginput data produk baru ke sistem.
Flow of Events	<ol style="list-style-type: none"> <li>1. Sistem memastikan user memiliki izin[A1](BR1)</li> <li>2. Pengguna memulai proses penambahan produk.</li> <li>3. Sistem menampilkan formulir input data dan stok produk.</li> <li>4. Pengguna mengisi informasi produk dan kategori.</li> <li>5. Sistem memvalidasi data input.[A2](BR2)</li> <li>6. Sistem mengonfirmasi bahwa produk berhasil dicatat.</li> </ol>

Alternate Flows	A1 : Jika user tidak memiliki izin maka akan muncul notifikasi dan user tidak bisa lanjut ke langkah selanjutnya A2 : Jika data tidak valid, maka akan muncul notifikasi bahwa data tidak valid dan mengarahkan user untuk menginput ulang data.(BR3)
Post-conditions	Data produk berhasil disimpan dalam database.
Business Rules	BR1 : Hanya user yang memiliki izin yang bisa mengakses data BR2 : Data yang dimasukkan harus sesuai dengan format BR3 : Sistem memberi notifikasi data yang diinput tidak sesuai format yang diminta
Non-Functional Requirements	NFR 1: Sistem harus bisa melayani 5 user sekaligus NFR 2: Sistem harus bisa update secara real time
Notes	Sistem harus memberikan notifikasi kepada pengguna jika ada kesalahan dalam input data.  UC_TB.06

Form Element	Description
Use Case Name	Perbarui Katalog Produk dan Jenisnya
Description	Menangani pembaruan informasi produk dan jenisnya.
Actors	Primary : Admin
Priority	Must Have = pembaruan informasi katalog yang tepat diperlukan untuk menampilkan produk yang tersedia dan menghindari informasi yang salah tentang stok atau harga, yang bisa menyebabkan gangguan operasional dan kekecewaan pelanggan.
Risk	Sedang = risiko salah input data atau dampak pada data lain.
Pre-conditions and assumptions	Pengguna memiliki izin untuk memperbarui data produk..
Extension Points	Notifikasi kesalahan input data,notifikasi perubahan berdampak pada data lain.

Trigger	Admin memperbarui data produk atau jenisnya.
Flow of Events	<ol style="list-style-type: none"> <li>1. Sistem memastikan user memiliki izin untuk memperbarui data[A1](BR1)</li> <li>2. Pengguna memilih produk yang akan diperbarui.</li> <li>3. Sistem menampilkan informasi produk yang akan diperbarui.</li> <li>4. Pengguna mengubah data produk.</li> <li>5. Sistem memvalidasi data input.[A2](BR2)</li> <li>6. Sistem memperbarui data</li> <li>7. Sistem mengonfirmasi pembaruan berhasil.</li> </ol>
Alternate Flows	A1 : Jika user tidak memiliki izin maka akan muncul notifikasi dan user tidak bisa lanjut ke langkah selanjutnya A2 : Jika data tidak valid, maka akan muncul notifikasi bahwa data tidak valid dan mengarahkan user untuk menginput ulang data.(BR3)
Post-conditions	Data produk berhasil disimpan dalam database.
Business Rules	<p>BR1 : Hanya user yang memiliki izin yang bisa mengakses data</p> <p>BR2 : Data yang dimasukkan harus sesuai dengan format</p> <p>BR3 : Sistem memberi notifikasi data yang diinput tidak sesuai format yang diminta</p>
Non-Functional Requirements	<p>NFR 1: Wajib respon dalam 3 detik jika tidak harus di time out</p> <p>NFR 2: Sistem harus bisa update secara real time</p>
Notes	<p>Sistem harus memberikan notifikasi kepada pengguna jika ada kesalahan dalam input data.</p> <p>UC_TB.06</p>

Form Element	Description
Use Case Name	Hapus Katalog Produk dan Jenisnya
Description	Menangani penghapusan produk dan jenisnya dari sistem.
Actors	Primary : Admin

Priority	Must Have = memastikan bahwa produk yang tidak lagi tersedia atau usang dihapus dari katalog, agar informasi tetap relevan dan terkini, menghindari kecacuan informasi yang bisa mempengaruhi pengalaman pengguna.
Risk	Sedang = risiko dampak pada data lain.
Pre-conditions and assumptions	Pengguna memiliki izin untuk menghapus produk dan jenisnya.
Extension Points	Notifikasi jika penghapusan memengaruhi transaksi.
Trigger	Admin memilih untuk menghapus produk dan jenisnya.
Flow of Events	1. Sistem memastikan user memiliki izin untuk menghapus data[A1] 2. Pengguna memilih produk atau jenis yang akan dihapus. 3. Sistem menampilkan konfirmasi penghapusan. (BR2) 5. Sistem menghapus produk dari database. 6. Sistem mengonfirmasi penghapusan berhasil.
Alternate Flows	A1 : Jika user tidak memiliki izin maka akan muncul notifikasi dan user tidak bisa lanjut ke langkah selanjutnya
Post-conditions	Data produk berhasil dihapus.
Business Rules	BR1 : Hanya user yang memiliki izin yang bisa mengakses data BR2 : Harus memastikan bahwa user ingin menghapus data supaya menghindari ketidaksengajaan
Non-Functional Requirements	NFR 1 : Sistem harus bisa update secara real time
Notes	Sistem menyimpan log penghapusan untuk referensi di kemudian hari.  UC_TB.06

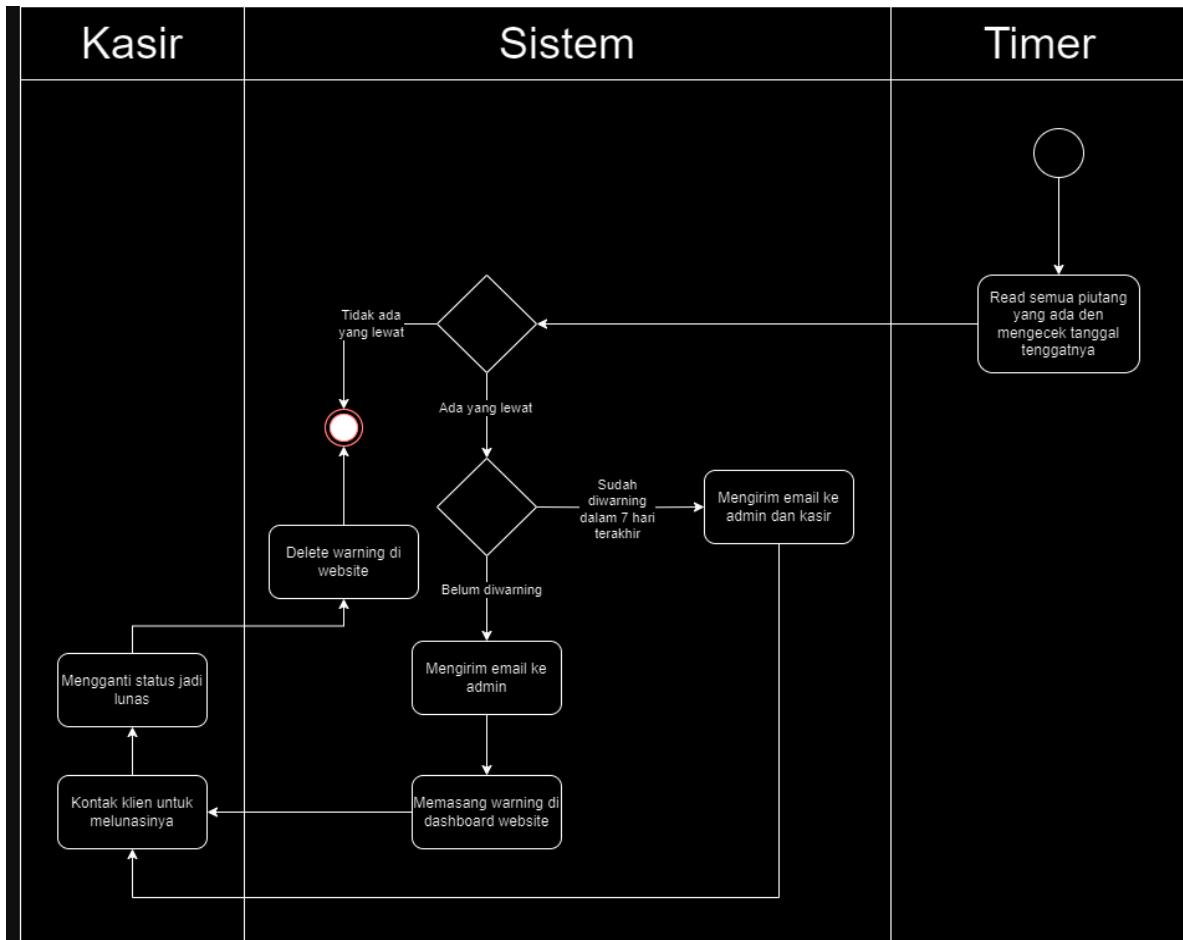
Form Element	Description
Use Case Name	Read Catatan Penjualan

Description	Menangani pembacaan catatan penjualan.
Actors	Primary : admin, kasir
Priority	Must Have : laporan keuangan yang akurat dan lengkap menjadi dasar bagi pengambilan keputusan manajemen dan pemenuhan kewajiban hukum terkait pelaporan keuangan.
Risk	Sedang = risiko data yang tidak lengkap atau salah.
Pre-conditions and assumptions	Pengguna memiliki izin untuk membaca catatan penjualan.
Extension Points	Notifikasi jika catatan penjualan tidak lengkap.
Trigger	Pengguna ingin melihat catatan penjualan.
Flow of Events	<ol style="list-style-type: none"> <li>1. Pengguna memulai permintaan untuk melihat laporan pendapatan.</li> <li>2. Sistem memverifikasi izin pengguna.[A1](BR1)</li> <li>3. Sistem mengambil data pendapatan dari database.</li> <li>4. Sistem memproses data pendapatan dan keuntungan.</li> <li>5. Sistem menghasilkan laporan pendapatan/keuntungan.</li> <li>6. Sistem menampilkan laporan kepada pengguna.(BR2)[A2]</li> </ol>
Alternate Flows	<p>A1 : Jika user tidak memiliki izin untuk melihat data maka akan memunculkan notif, dan tidak akan bisa lanjut ke langkah berikutnya</p> <p>A2 : Jika user mengklik ingin ubah format ke bentuk excel, atau pdf maka akan langsung mendownload data yang sedang ditunjukkan ke dalam bentuk pdf atau excel</p>
Post-conditions	Catatan penjualan berhasil ditampilkan.
Business Rules	<p>BR1 : User harus memiliki akses untuk melihat data</p> <p>BR2 : Laporan bisa diexport ke dalam bentuk excel atau pdf</p>
Non-Functional Requirements	<p>NFR 1: Sistem harus bisa melayani 5 user sekaligus</p> <p>NFR 5: Sistem harus menyala 24/7</p> <p>NFR 6: Sistem harus bisa update secara real time</p>
Notes	Sistem harus memberikan peringatan kepada pengguna jika data yang diperlukan untuk laporan tidak tersedia.

	UC_TB.08
--	----------

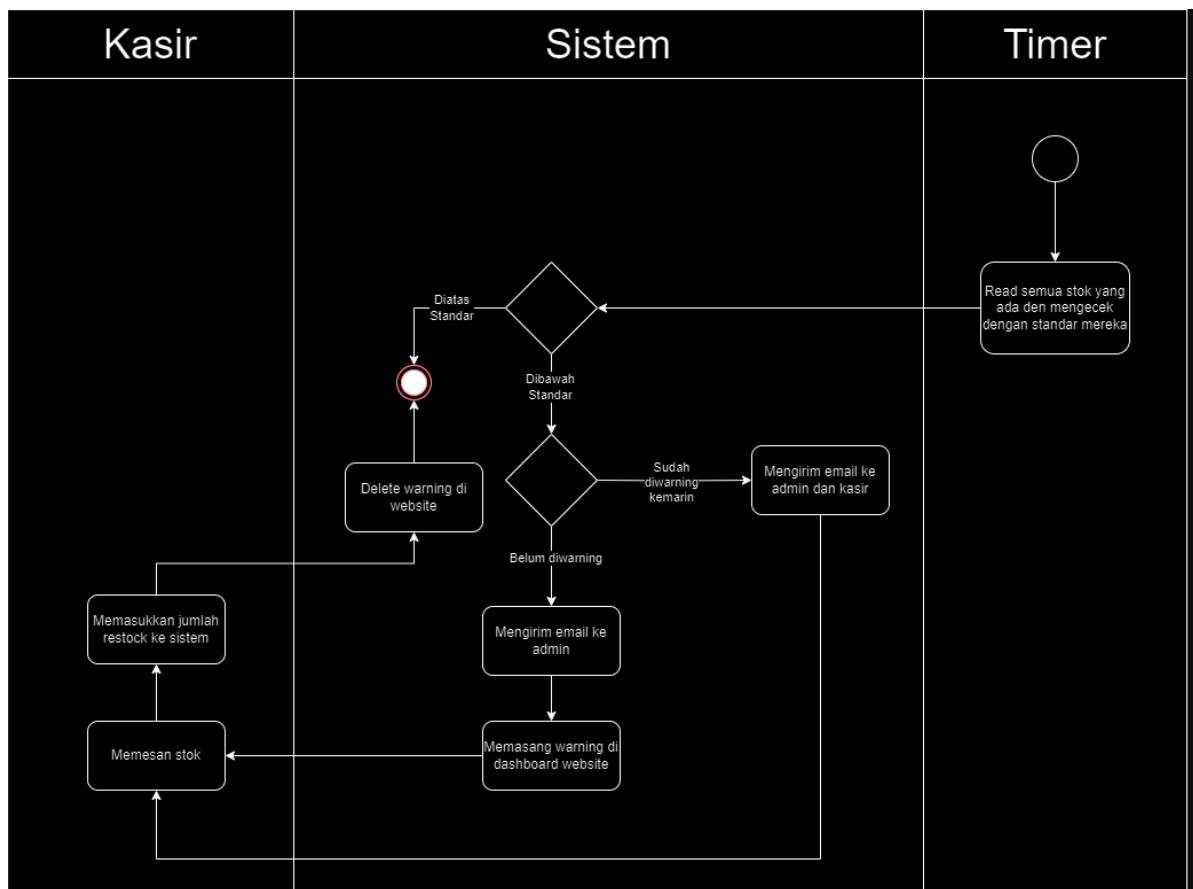
Form Element	Description
Use Case Name	Read Catalog dan Stok produk
Description	Use case ini menangani pembacaan katalog produk dan stoknya.
Actors	Primary : admin, kasir
Priority	Must Have : Informasi stok dan produk yang akurat adalah dasar bagi operasional penjualan dan manajemen inventaris. Tanpa akses yang tepat, kesalahan dalam pemesanan, penjualan, atau perencanaan dapat terjadi.
Risk	Rendah = risiko data yang tidak akurat atau tidak terupdate.
Pre-conditions and assumptions	Pengguna memiliki izin untuk mengakses katalog produk.
Extension Points	Notifikasi stok produk habis atau hampir habis.
Trigger	Pengguna ingin melihat daftar produk dan stok.
Flow of Events	<ol style="list-style-type: none"> <li>1. Pengguna memulai permintaan melihat katalog.</li> <li>2. Sistem memverifikasi izin pengguna.[A1](BR1)</li> <li>3. Sistem mengambil data produk dan stok.(BR2)</li> <li>4. Sistem menampilkan katalog produk beserta stoknya.</li> </ol>
Alternate Flows	A1 : Jika user tidak memiliki izin untuk melihat maka akan memunculkan notif dan tidak akan bisa lanjut ke langkah selanjutnya
Post-conditions	Katalog produk dan stok berhasil ditampilkan.
Business Rules	BR1 : Hanya user yang memiliki izin bisa melihat data BR2 : Informasi mengenai produk dan stok harus selalu akurat dan up to date.
Non-Functional Requirements	NFR 1: Sistem harus bisa melayani 5 user sekaligus NFR 2: Sistem harus menyala 24/7 NFR 3:Sistem harus bisa update secara real time, Flyer dan catalog harus memiliki resolusi 1080p

**b. Activity Diagram**



**UC\_TB.12 (Warning Tenggat Piutang)**

**Timothy Sebastian Itamurti**

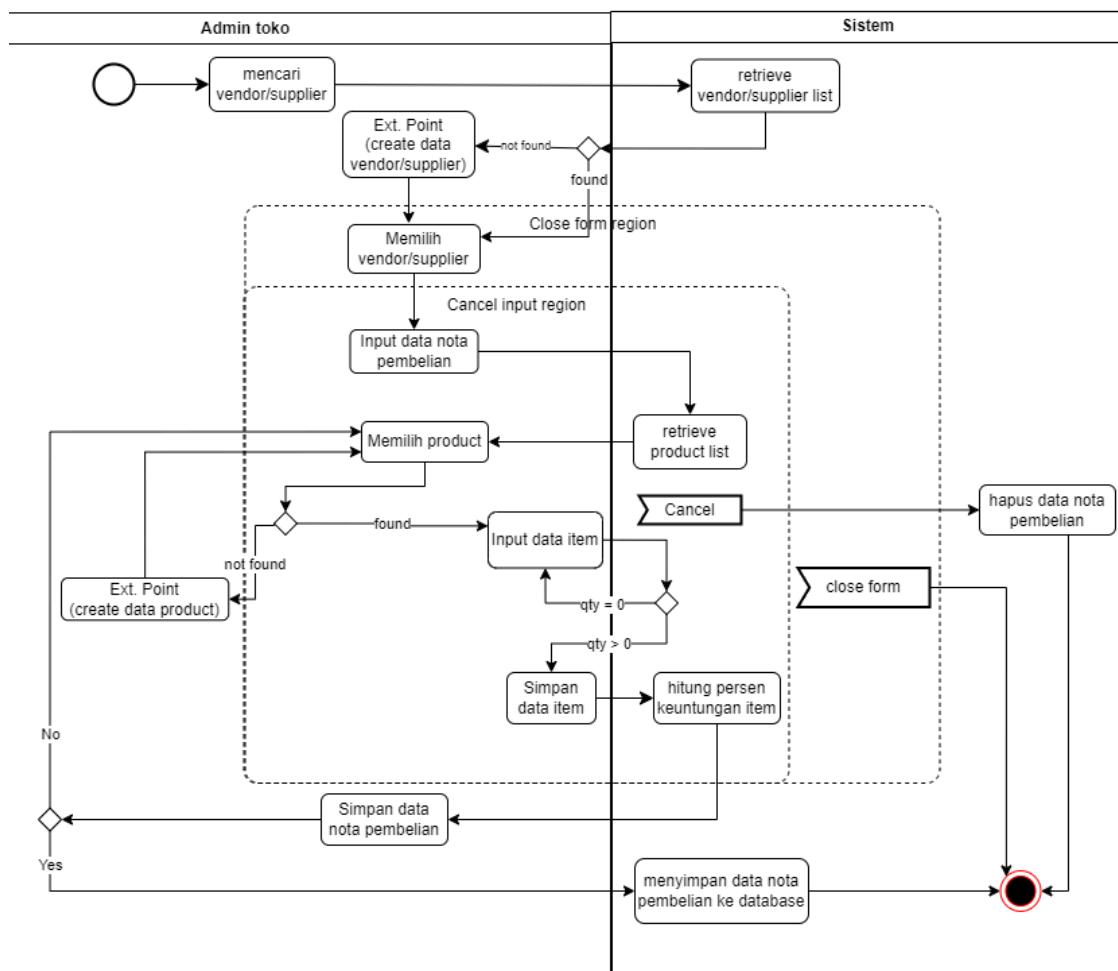


**UC\_TB.11 (Warning Stok Rendah)**

**Timothy Sebastian Itamurti**

Linda Jie - C14230052

### Create Nota Pembelian

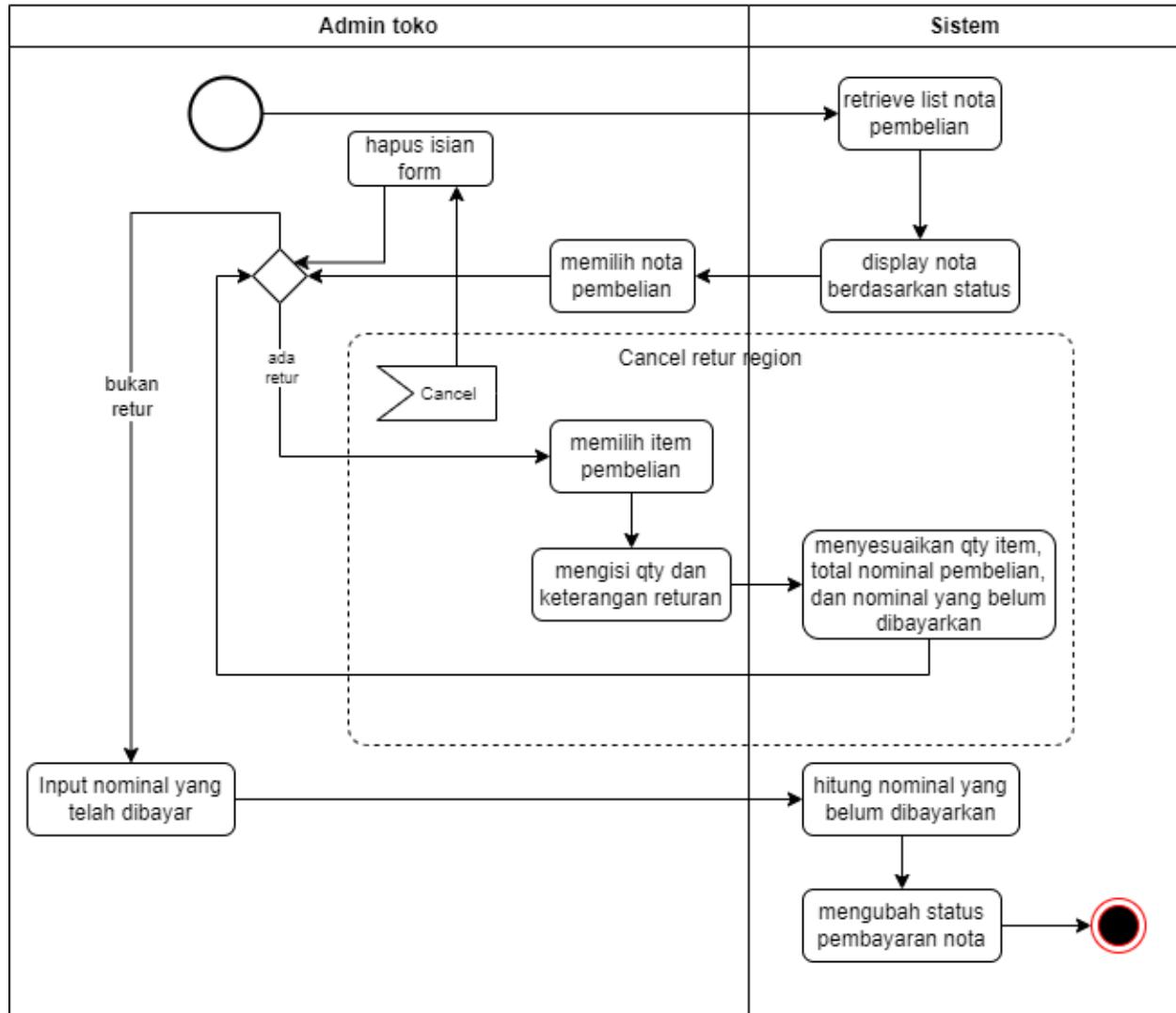


**UC\_TB.09**

Linda Jie - c14230052

## Update Status Pembayaran

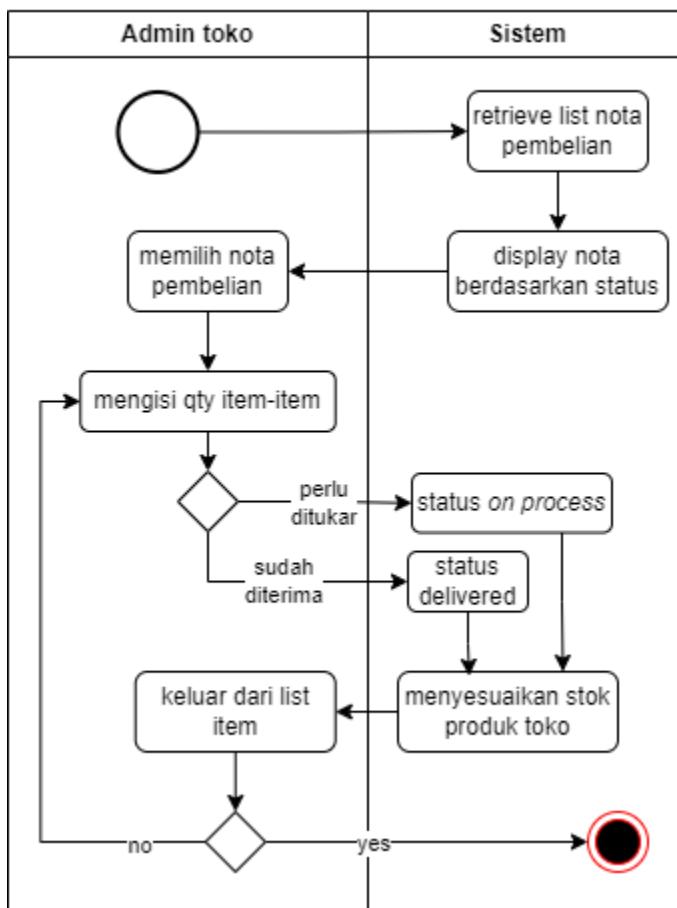
Linda Jie - C14230052



**UC\_TB.09**

Linda Jie - c14230052

Linda Jie - C14230052 Update Status Pengiriman

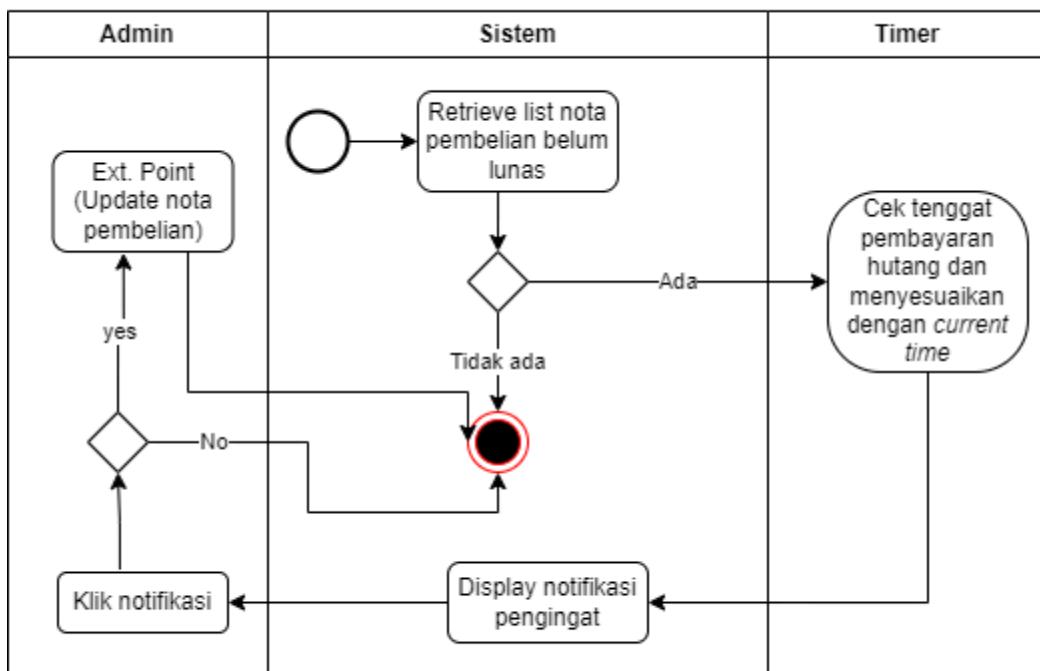


**UC\_TB.09**

Linda Jie - c14230052

Linda Jie - C14230052

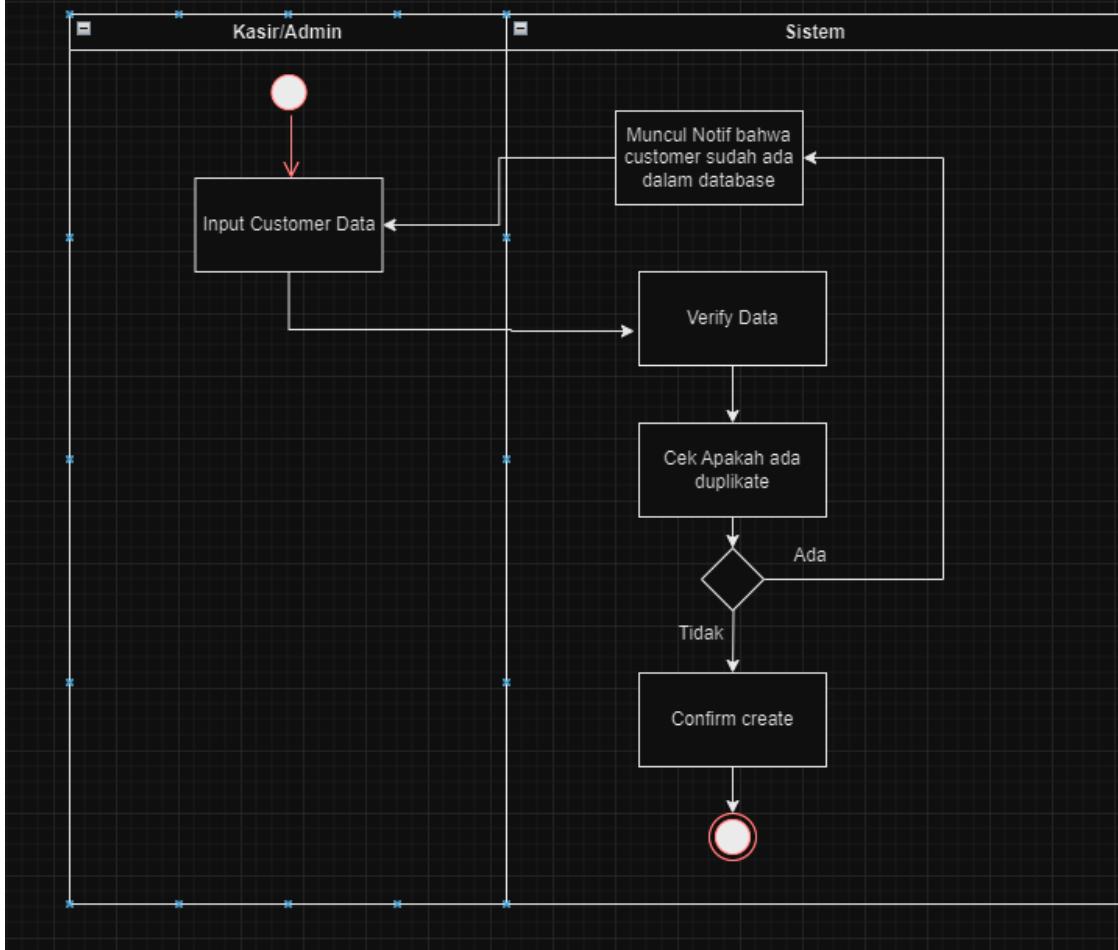
### Warning Tenggat Hutang



UC\_TB.13

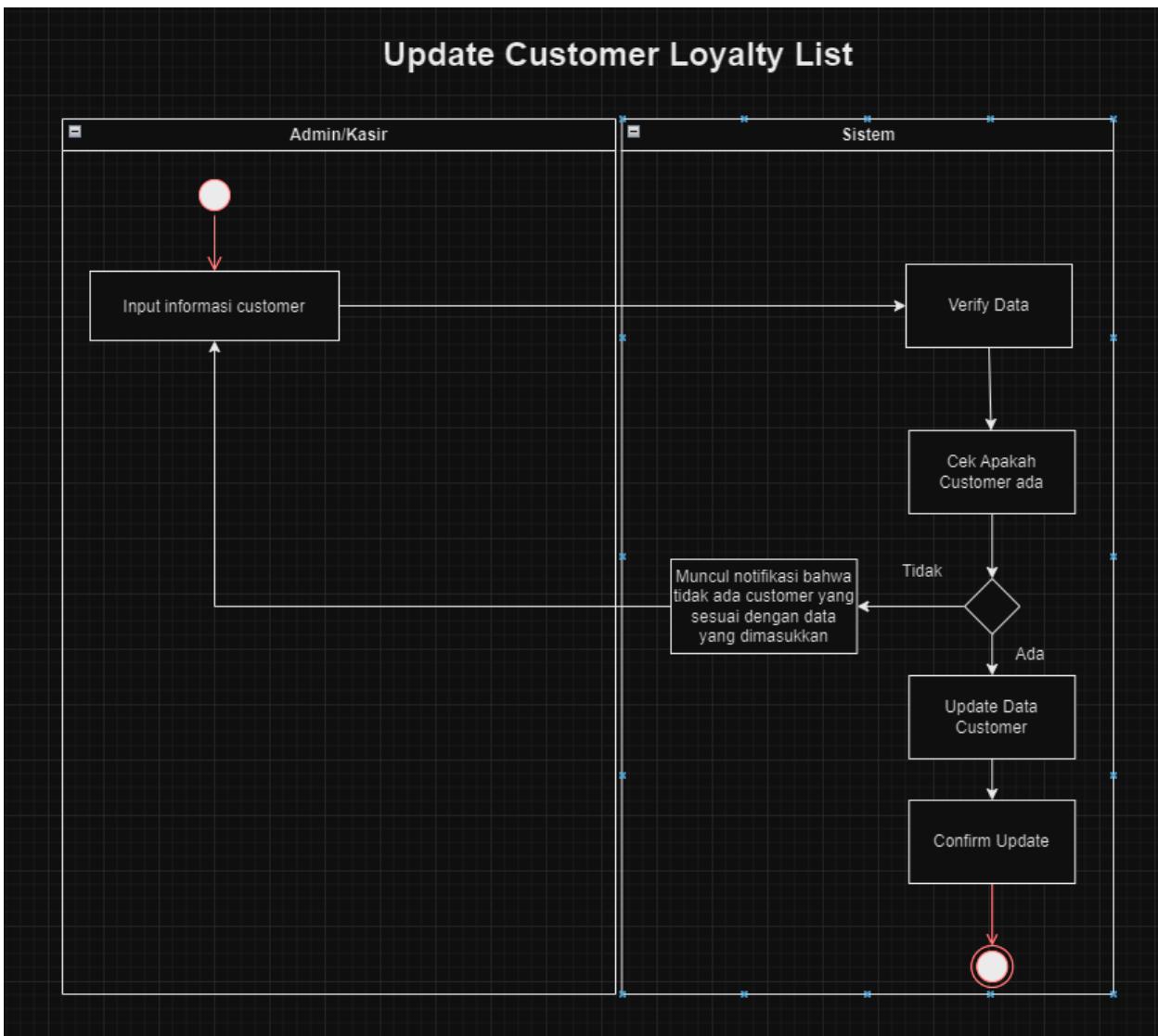
Linda Jie - c14230052

## Create Customer Loyalty List



UC\_TB.01 (Create Customer Loyalty List)  
Evelin Felicia Buntaran - C14230100

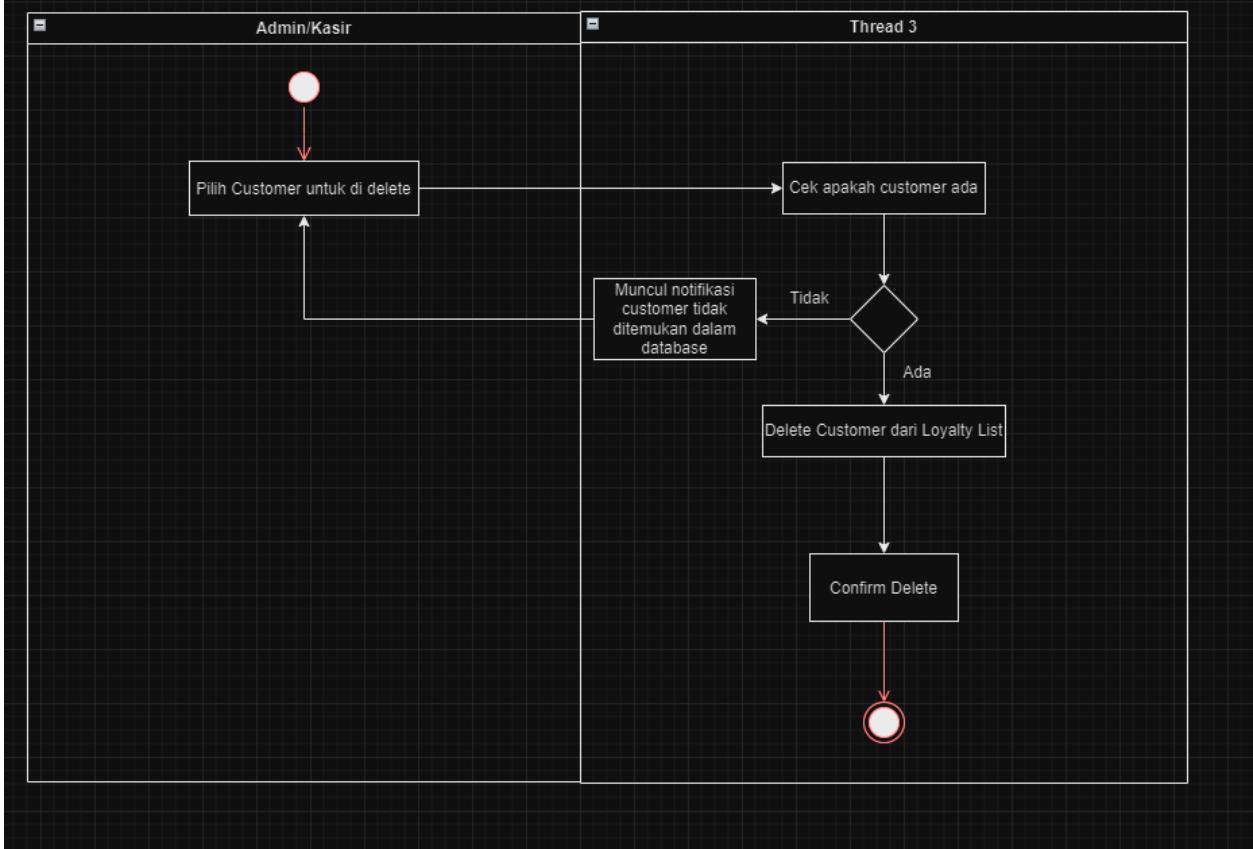
## Update Customer Loyalty List



**UC\_TB.01 (Update Customer Loyalty List)**

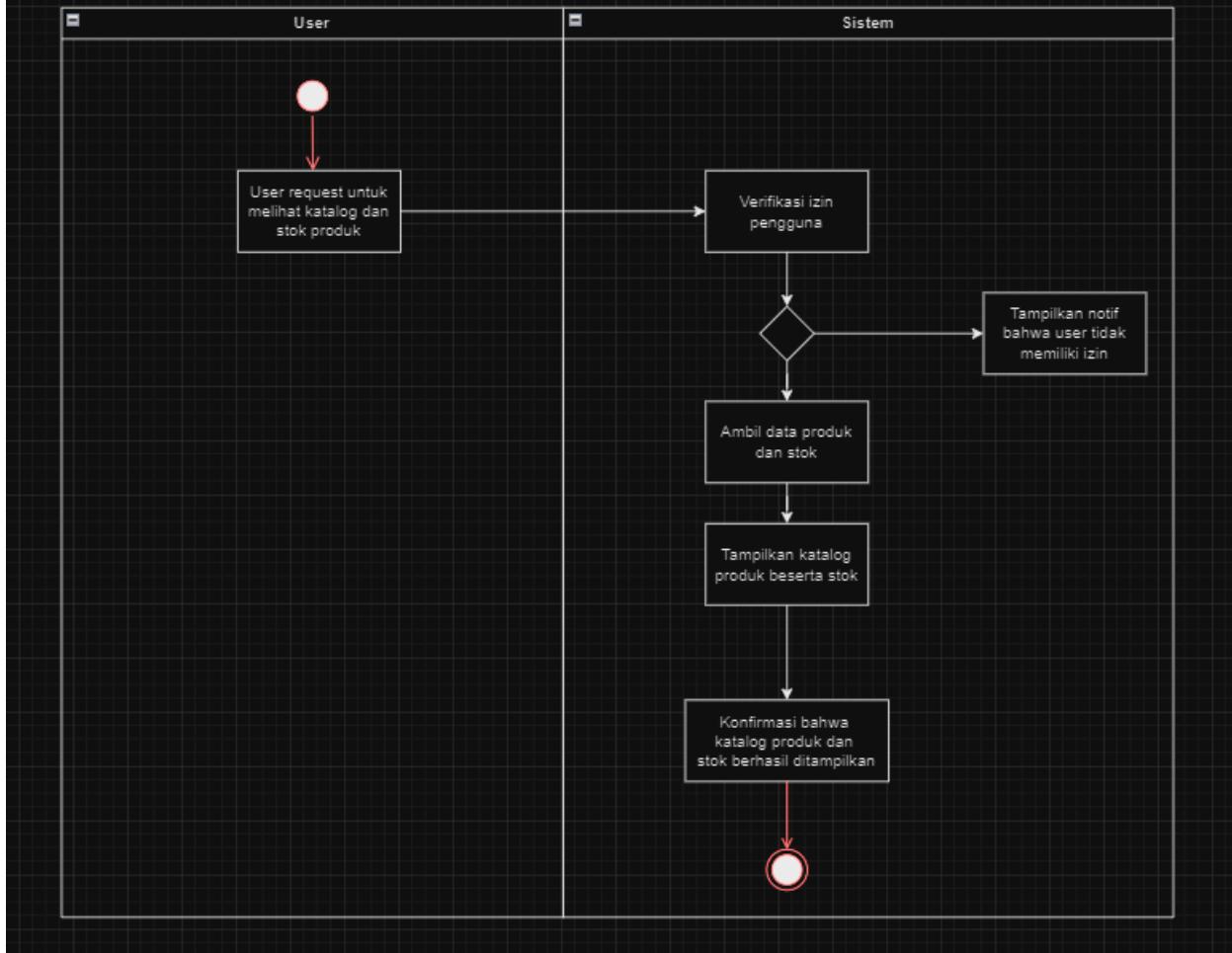
Evelin Felicia Buntaran - C14230100

## Delete Customer Loyalty List

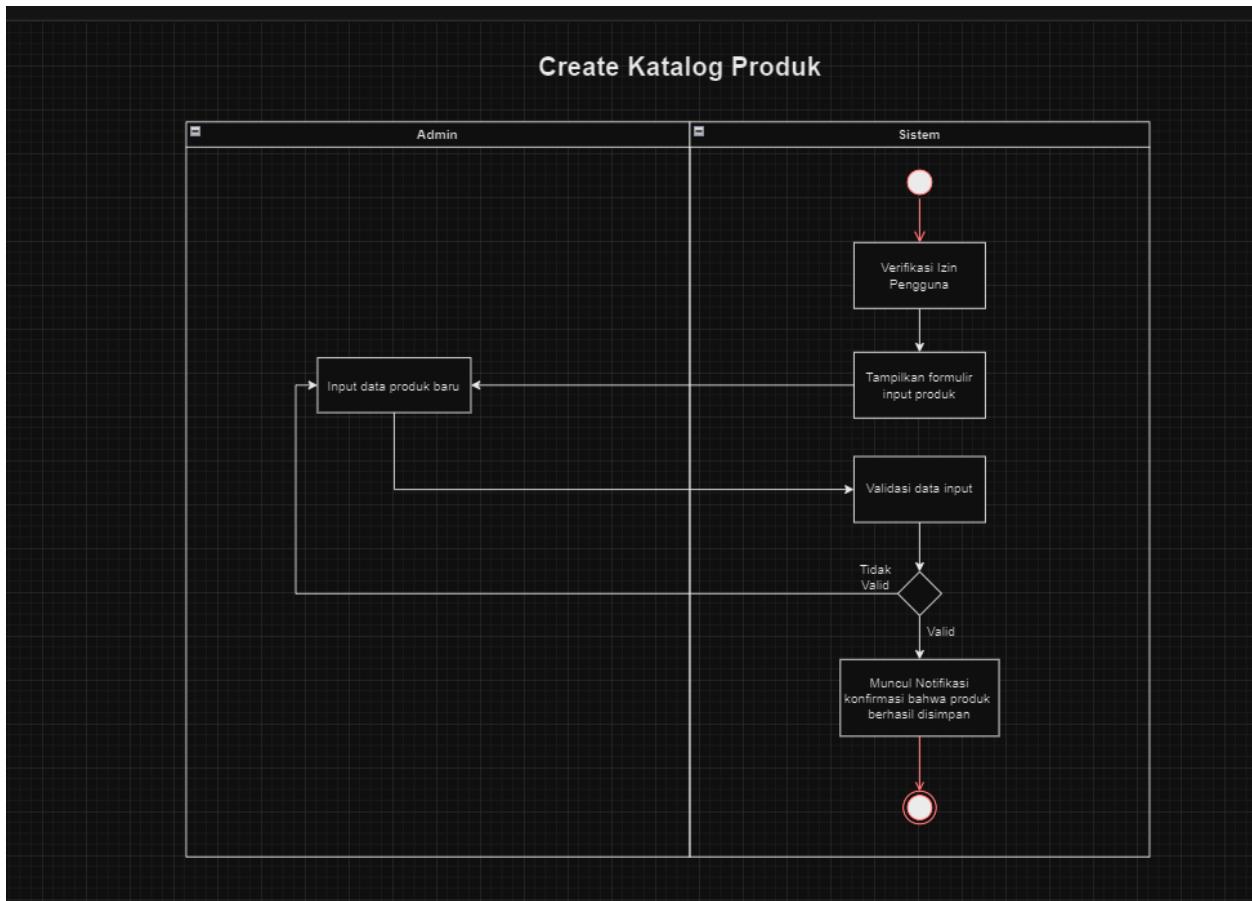


**UC\_TB.01 (Delete Customer Loyalty)**  
Evelin Felicia Buntaran - C14230100

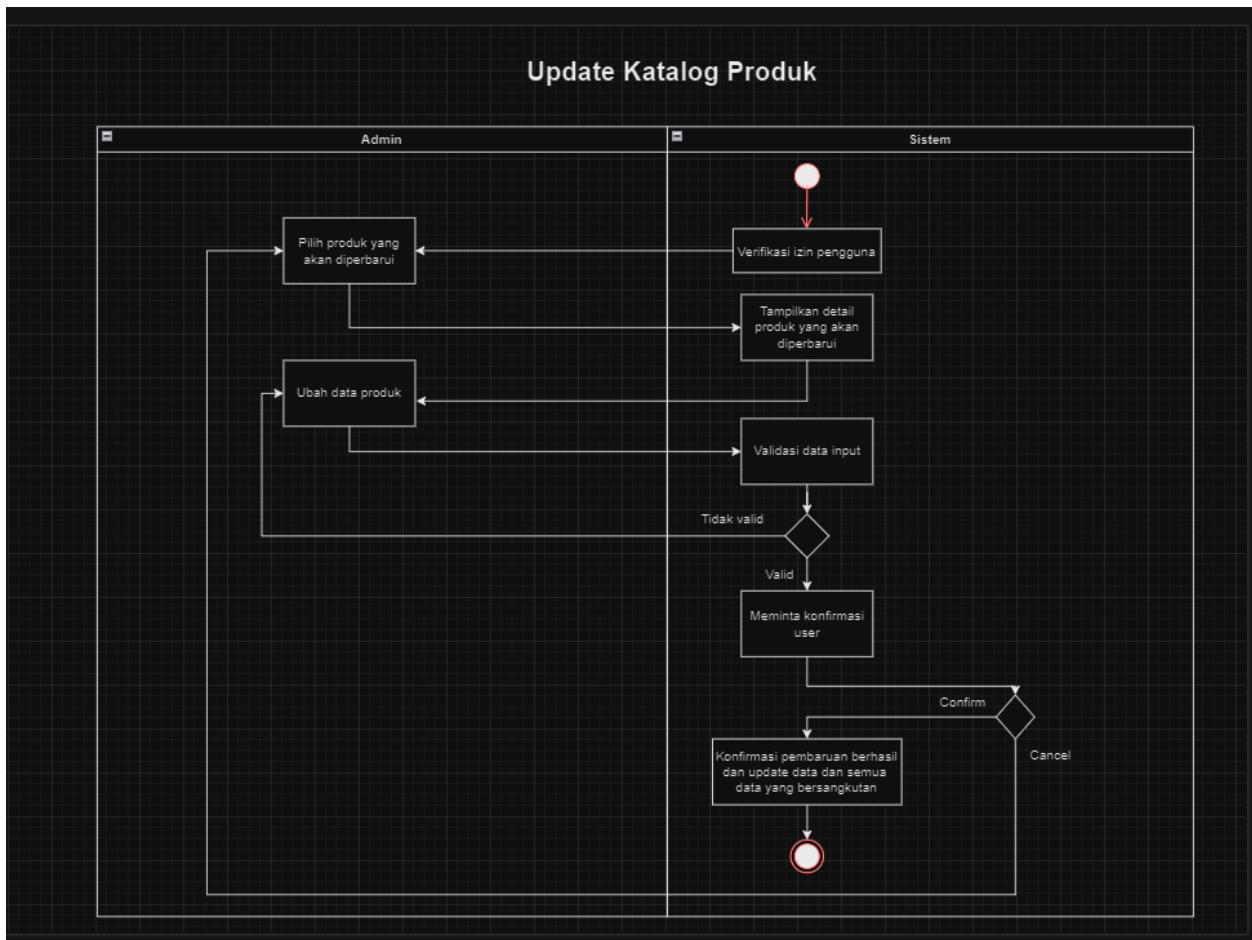
## Read Katalog dan Stok Produk



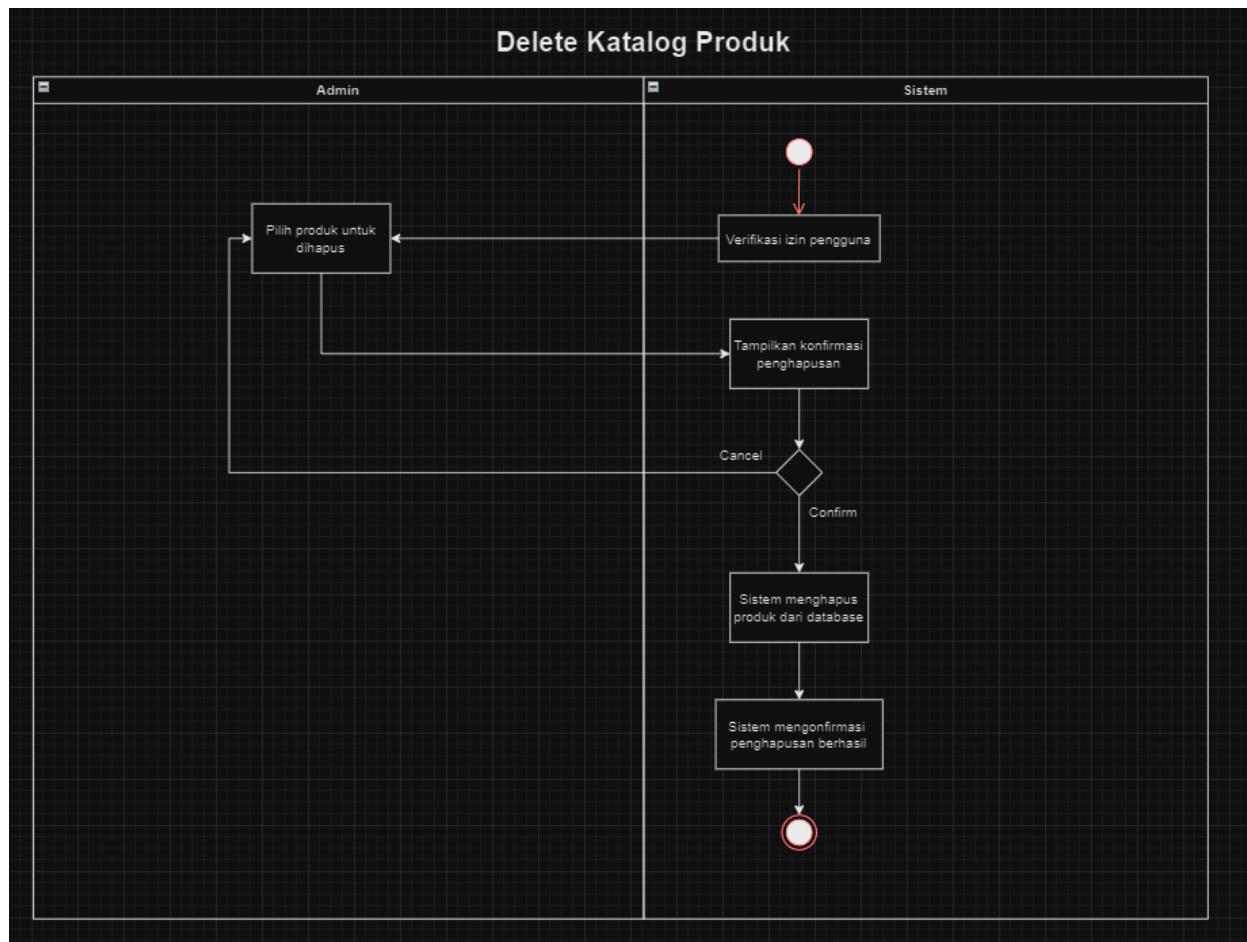
**UC\_TB.03 (Read Catalog and Stock Product)**  
**Evelin Felicia Buntaran - C14230100**



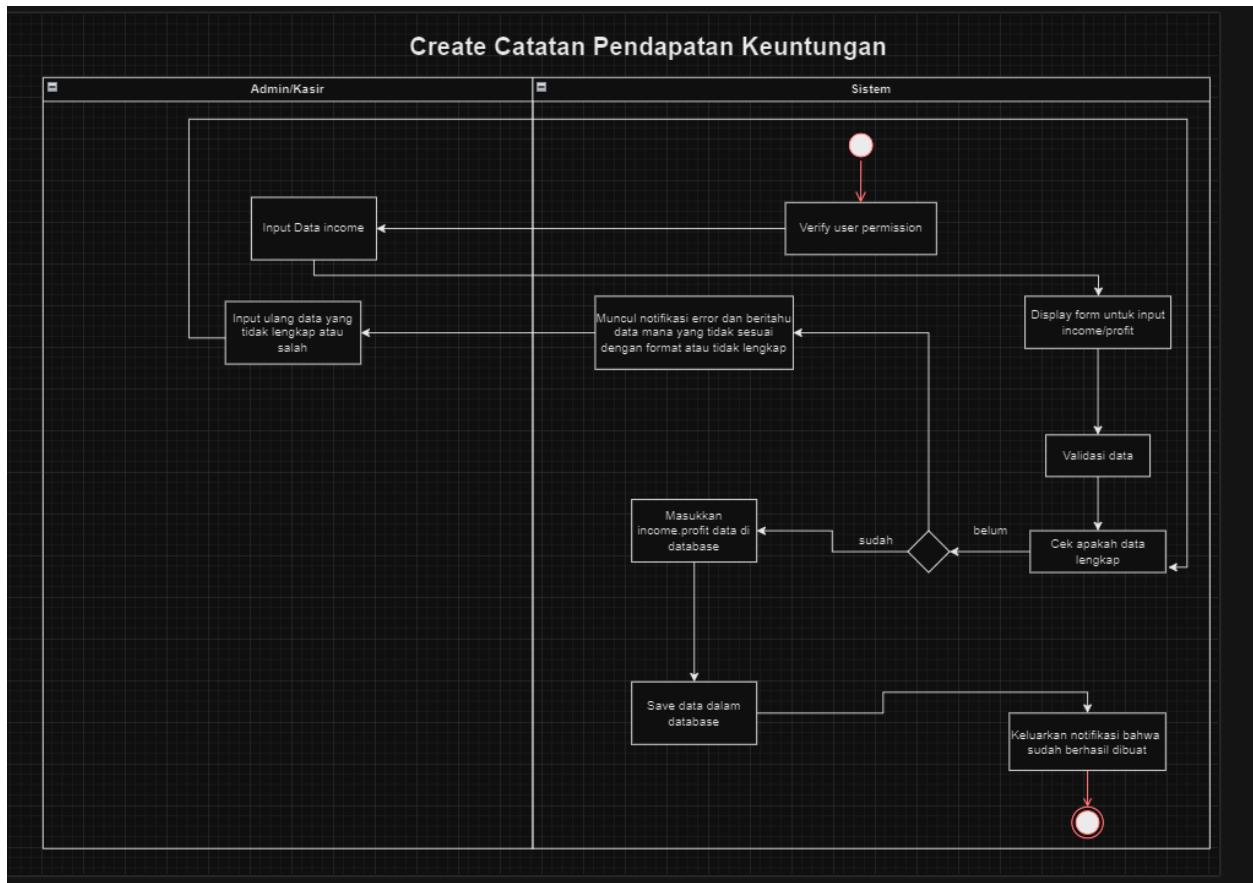
**UC\_TB.06 (Create Katalog Produk)**  
**Evelin Felicia Buntaran - C14230100**



**UC\_TB.06 (Update Katalog Produk)**  
**Evelin Felicia Buntaran - C14230100**



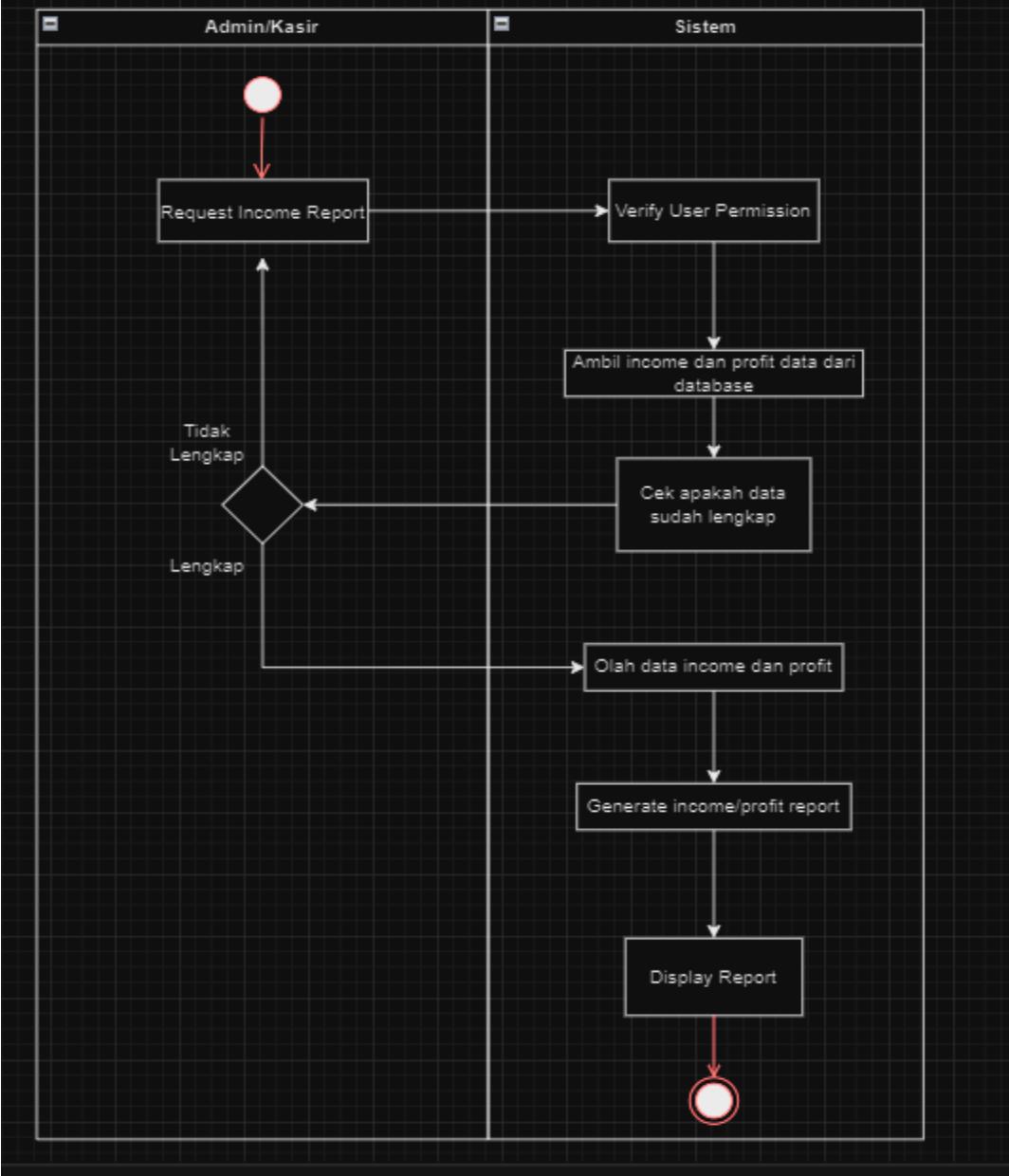
**UC\_TB.06 (Delete Catalog Produk)**  
**Evelin Felicia Buntaran - C14230100**



**UC\_TB.07 (Create Catatan Pendapatan Keuntungan)**

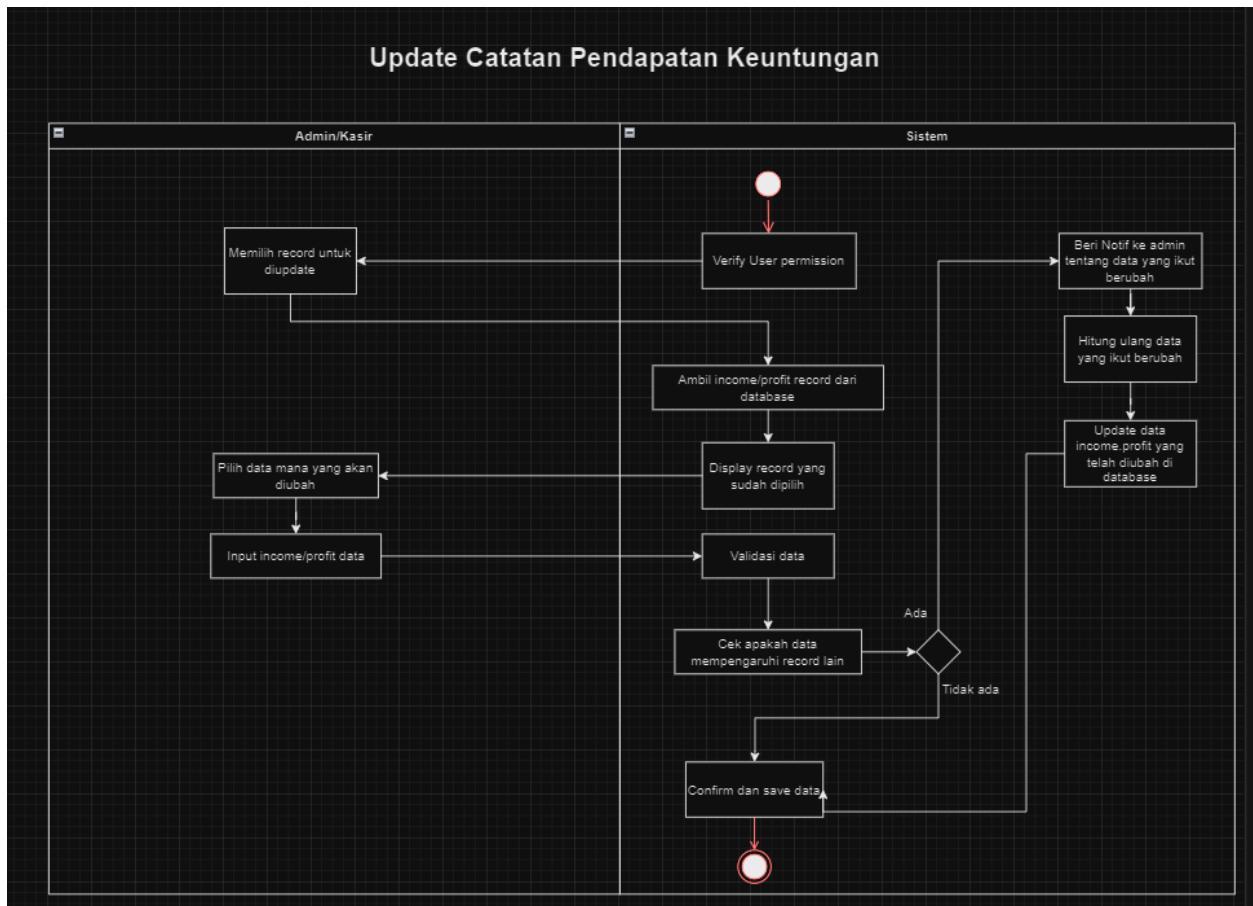
Evelin Felicia Buntaran - C14230100

## Read Catatan Pendapatan Keuntungan



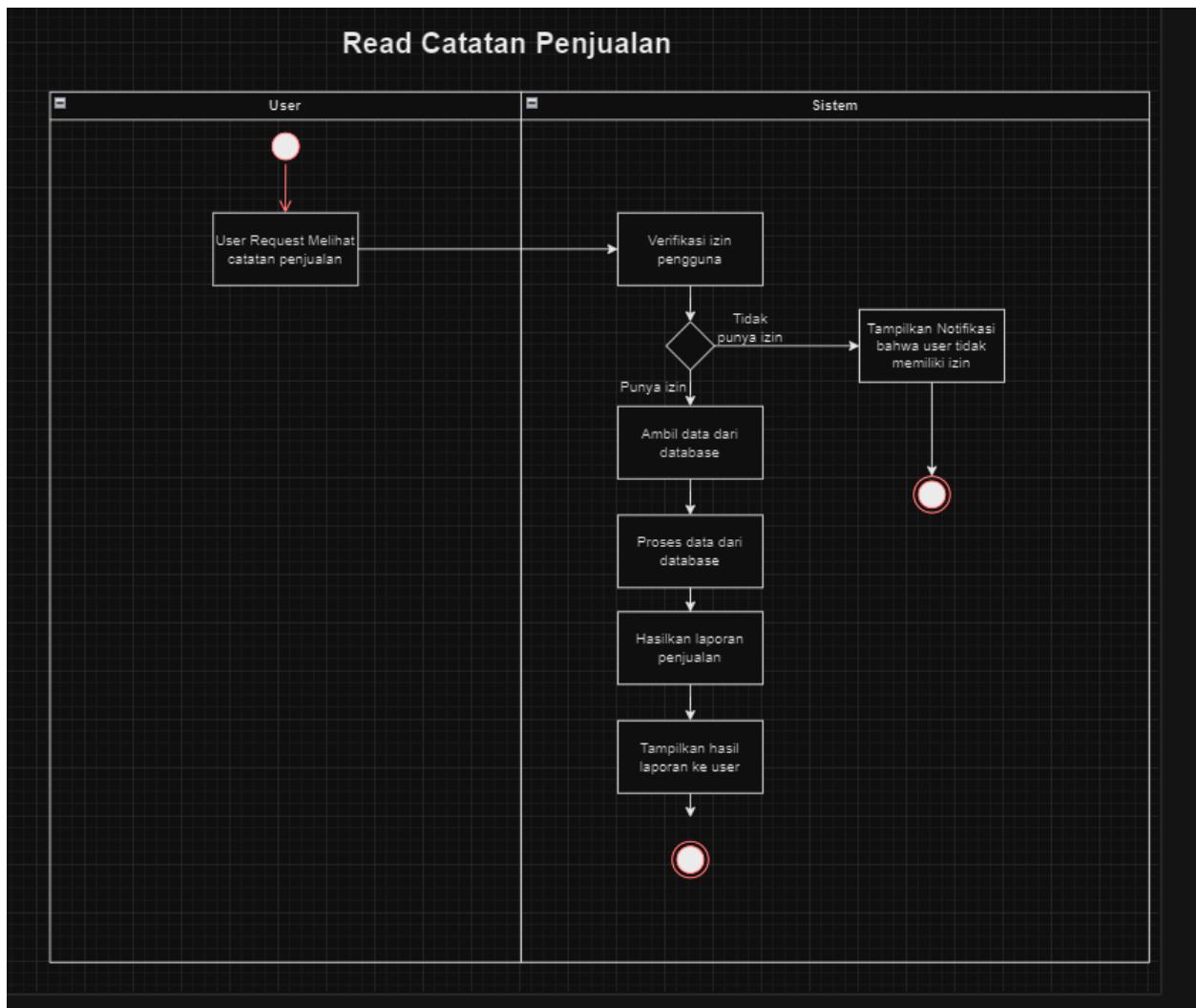
UC\_TB.07 (Read Catatan Pendapatan Keuntungan)

Evelin Felicia Buntaran - C14230100

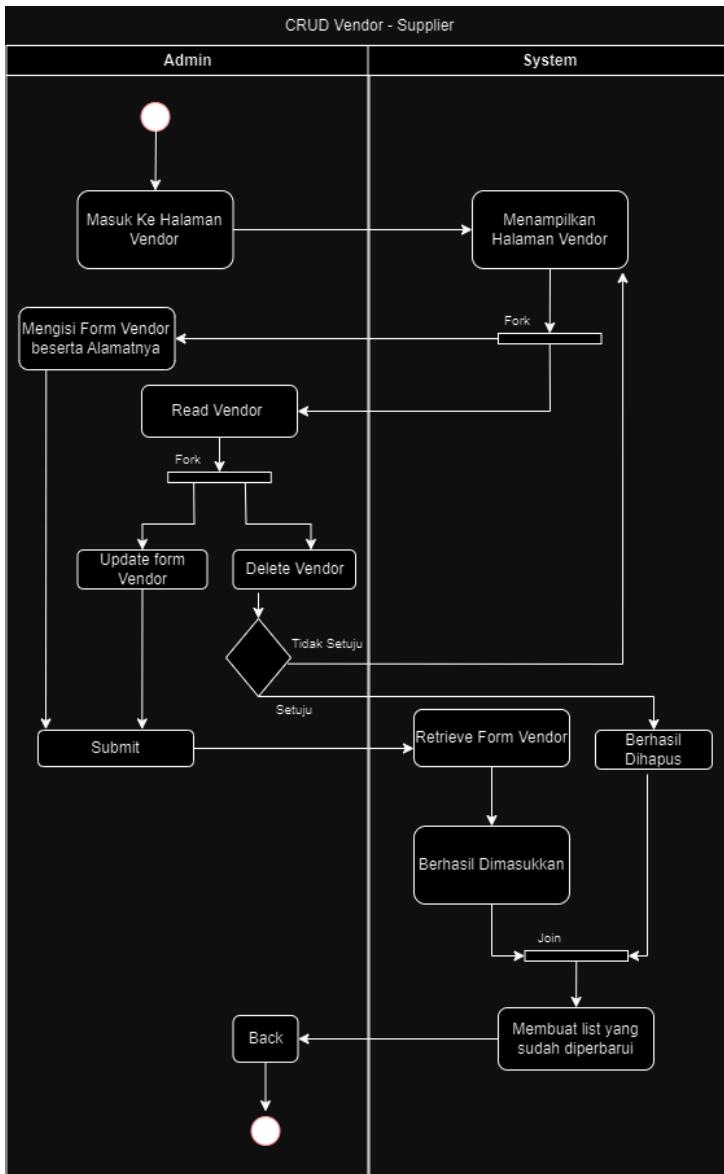


**UC\_TB.07 (Update Catatan Pendapat Keuntungan)**

**Evelin Felicia Buntaran - C14230100**

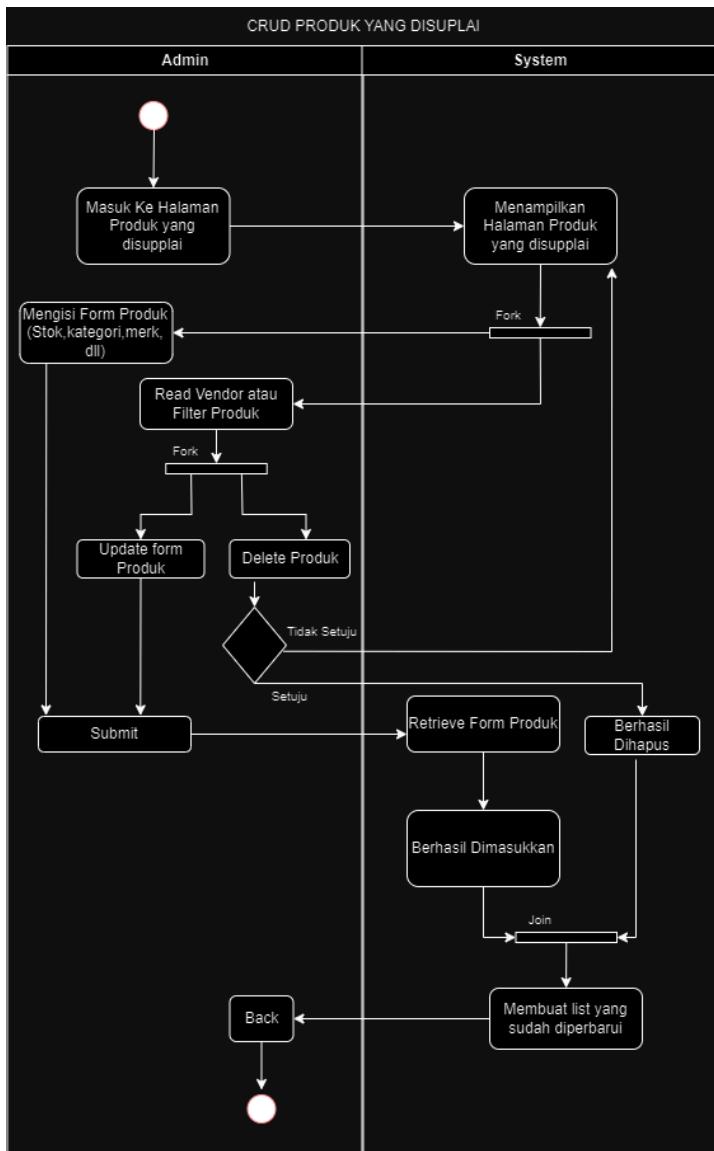


**UC\_TB.08 (Read Catatan Penjualan)**  
**Evelin Felicia Buntaran - C14230100**



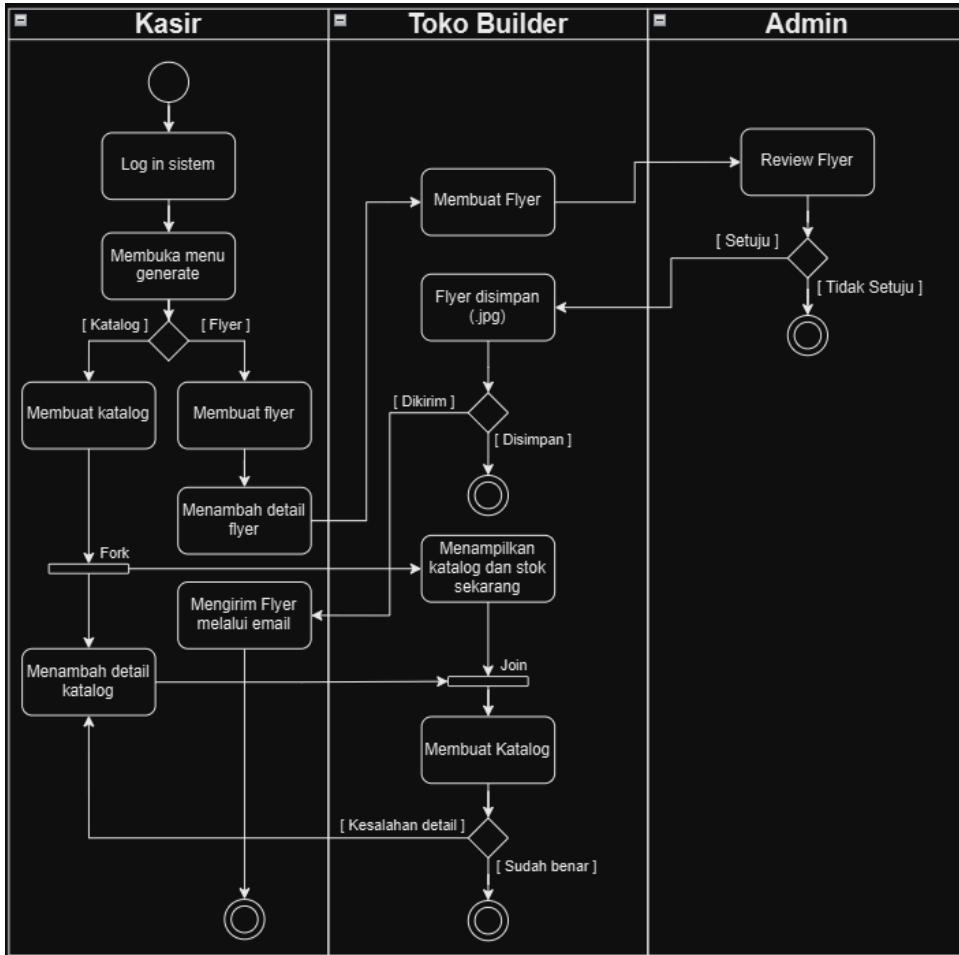
**UC\_TB.11 (CRUD Vendor - Supplier)**

Marcel Hans Sasongko / C14230099



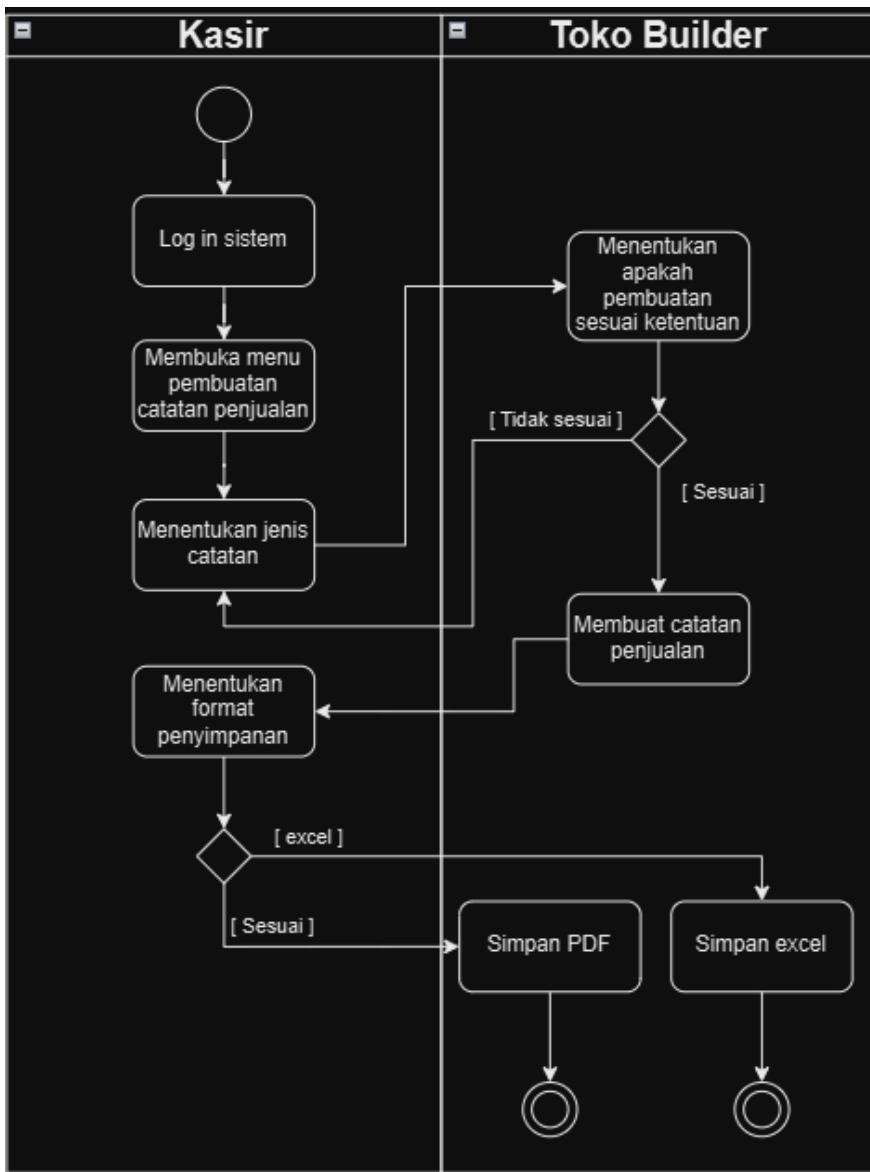
**UC\_TB.10 (CRUD Produk yang disupplai oleh Vendor)**

Marcel Hans Sasongko / C14230099



**UC\_TB.02 (generate flyer dan katalog)**

Joseph Evan Tanujaya /c14230096

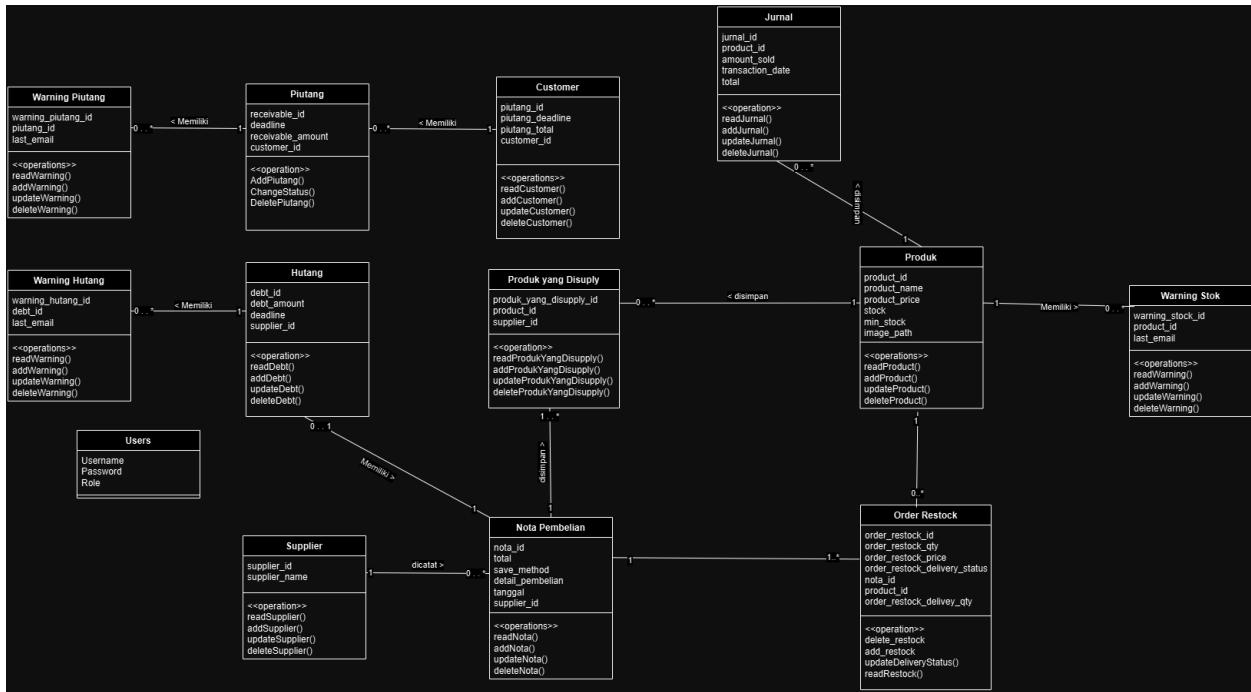


**UC\_TB.04 (Create catatan penjualan)**

Joseph Evan Tanujaya / c14230096

### c. Dokumen

#### a. Class diagram



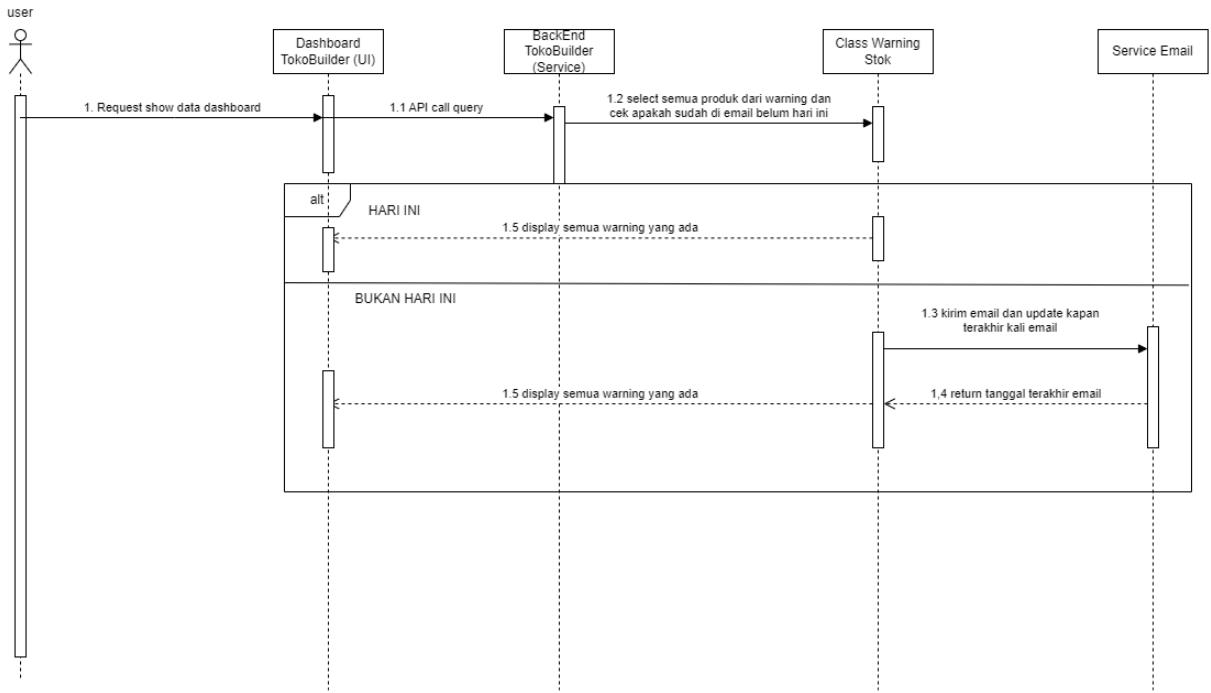
#### d. Sequence Diagram

Sequence diagram yang menggambarkan interaksi antar objek dan menunjukkan (memberi tanda atau petunjuk) komunikasi diantara objek-objek tersebut.

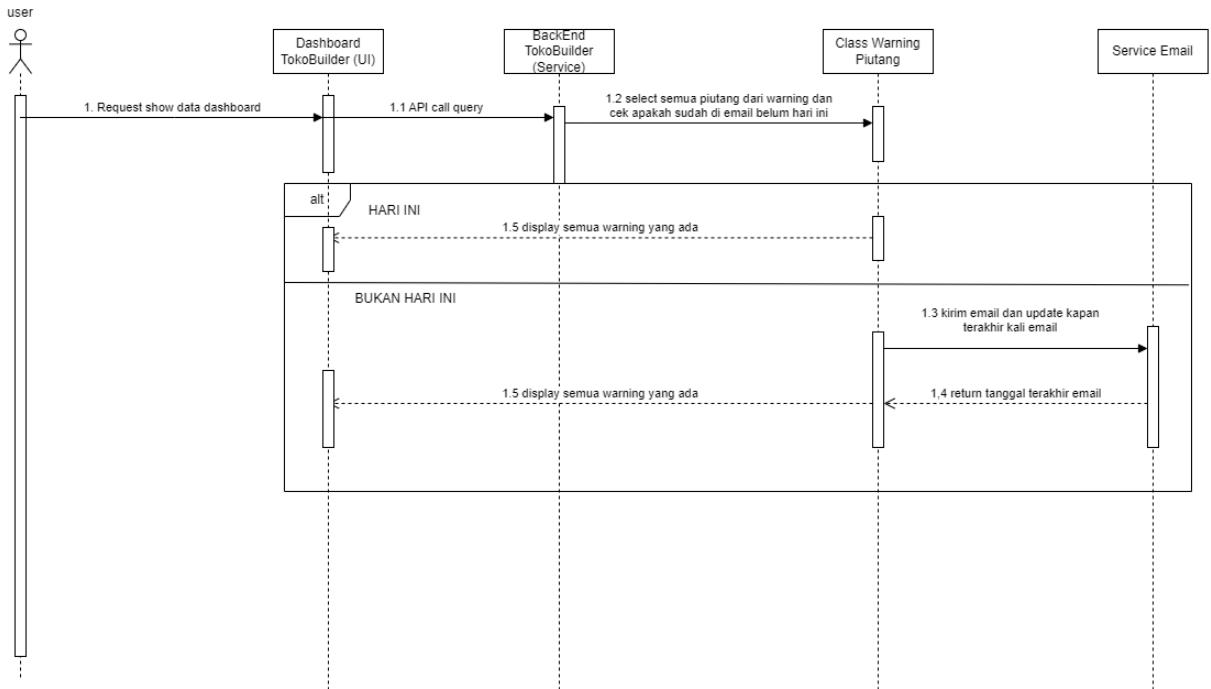
Gunakan format pengkodean untuk ID dari daftar sequence diagram (misal menggunakan rumus SEQ\_<Nama Modul / Aplikasi>.<2 digit numbering>, contoh: SEQ\_OBE.01).

**Timothy Sebastian Itamurti / c14230071**

**SEQ\_TB.05 (Warning stok rendah)**



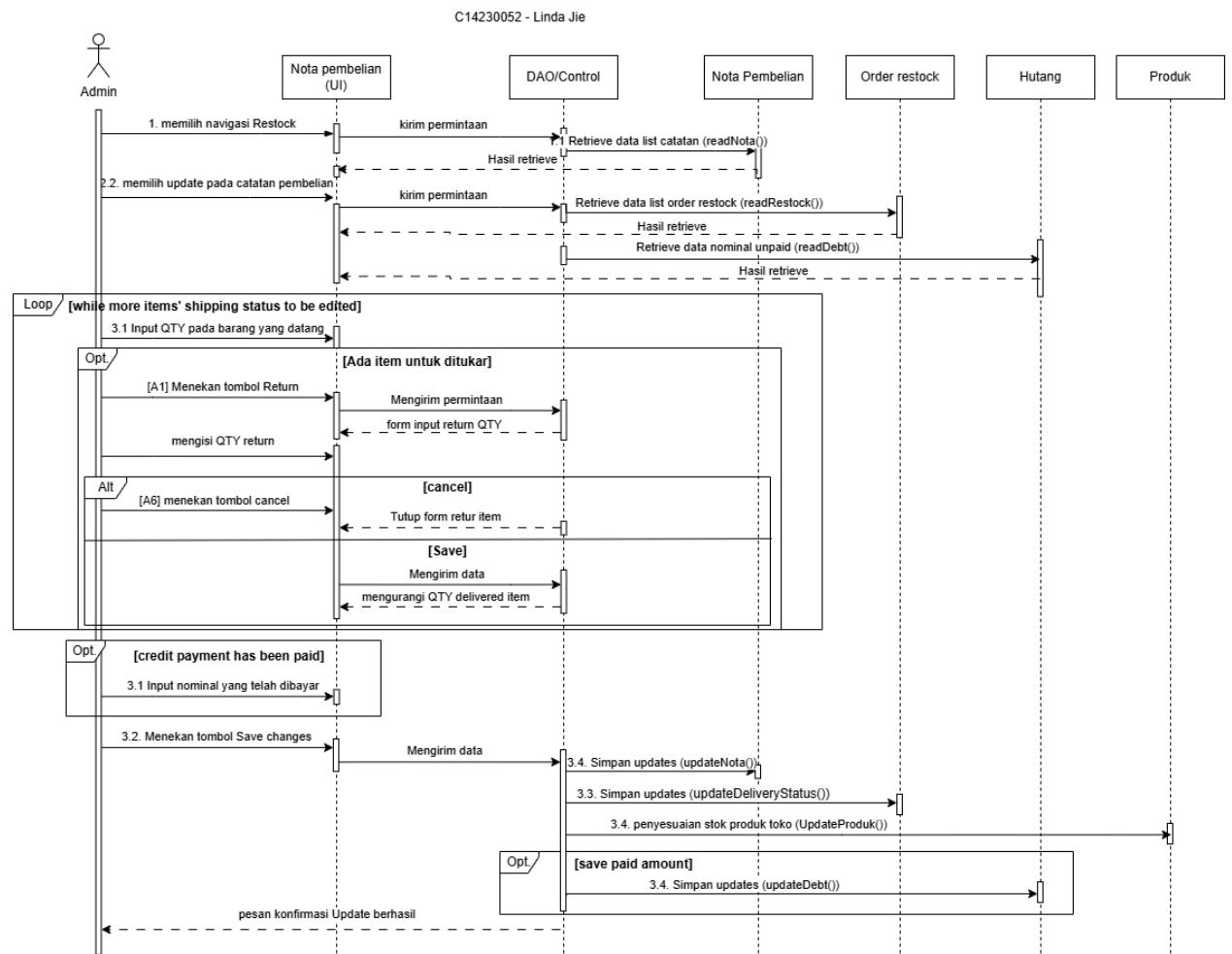
**SEQ\_TB.12 (Warning Tenggat Pembayaran Piutang)**



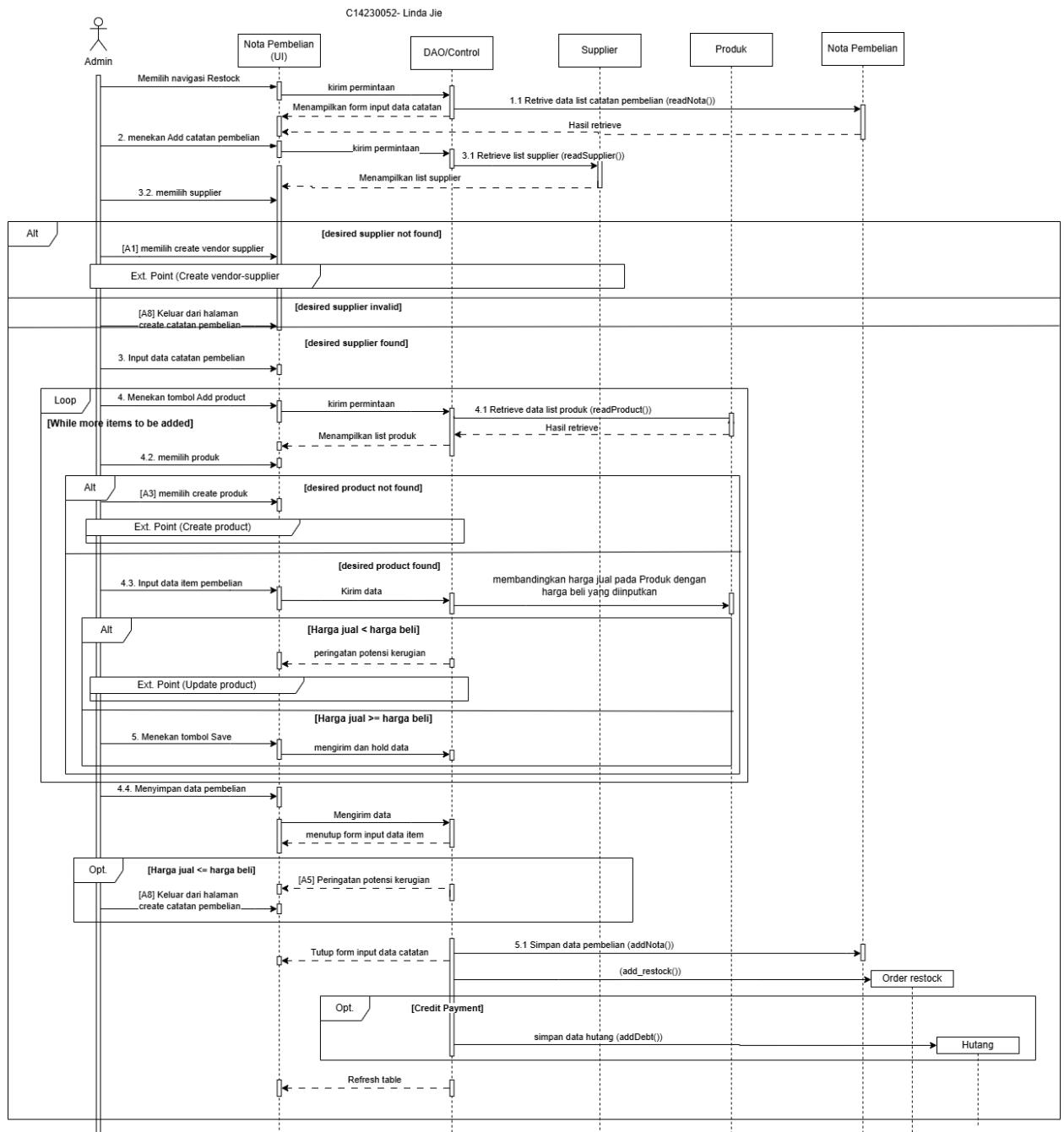
Linda Jie / c14230052

### SEQ\_TB.09 (CRUD Nota Catatan Pembelian)

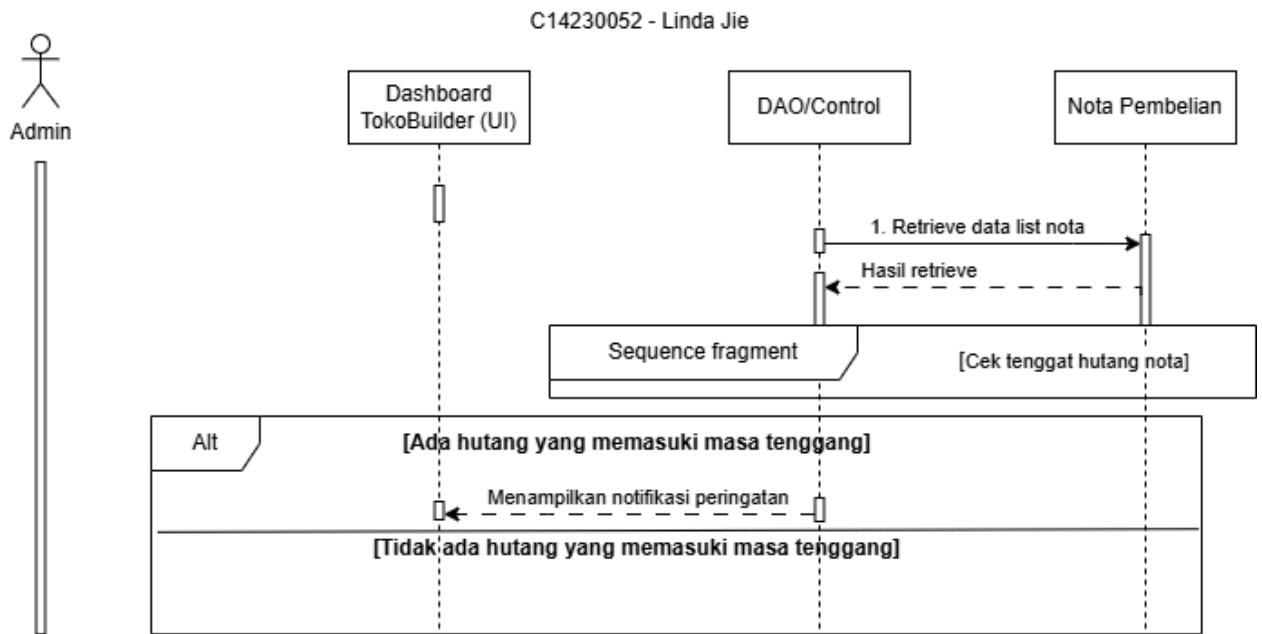
(Update)



(Create)

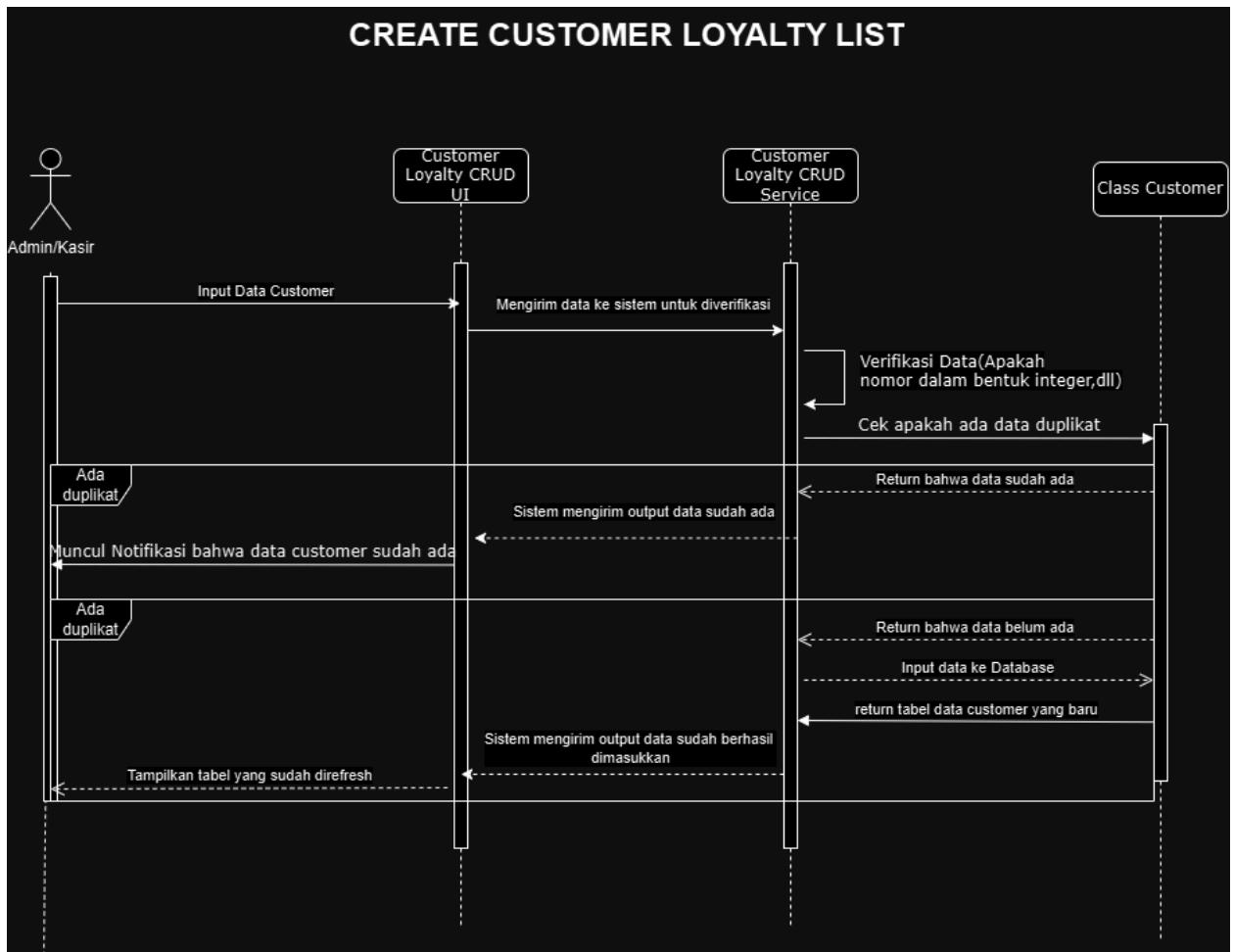


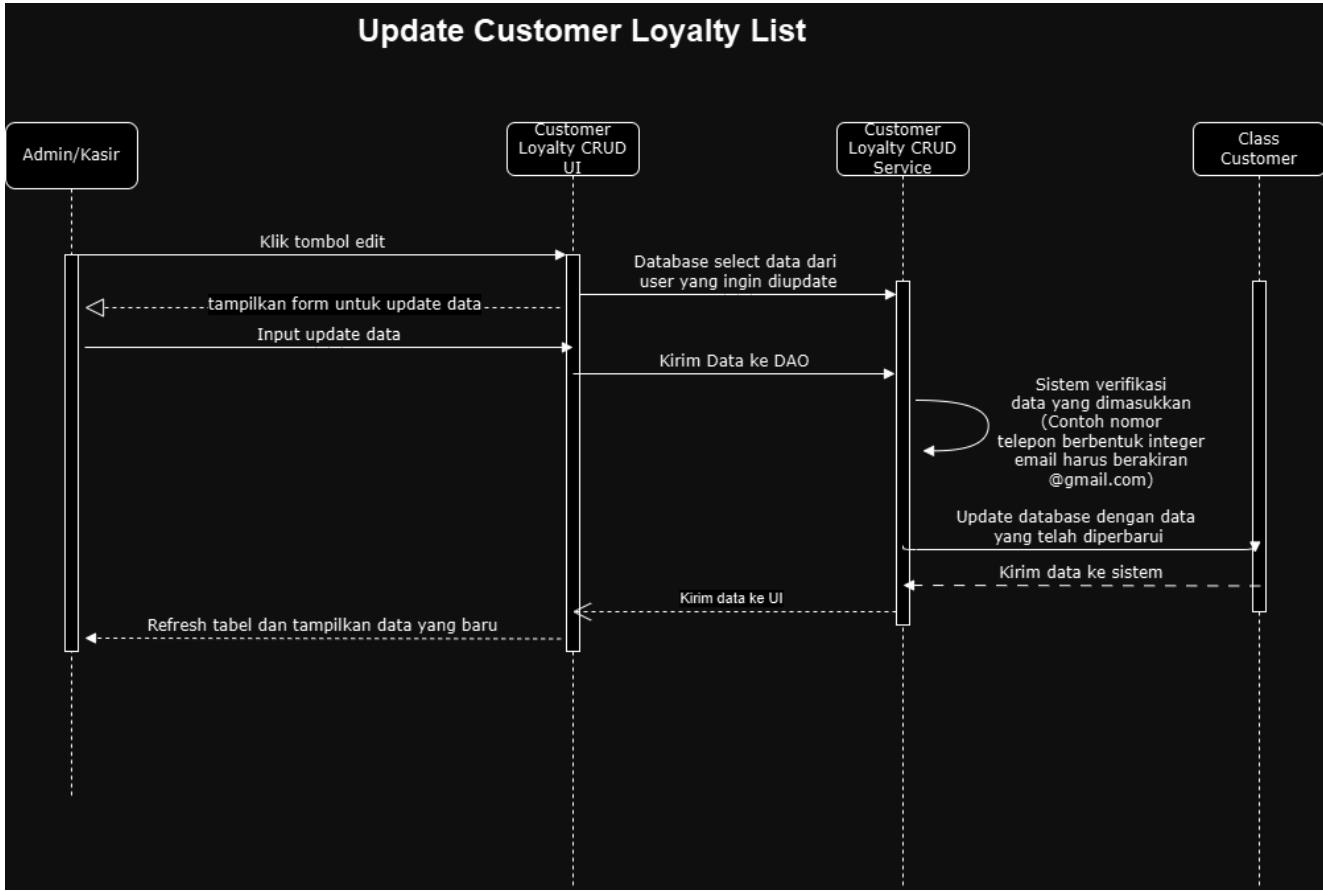
### SEQ\_TB.13 (Warning Tenggat Pembayaran Hutang)



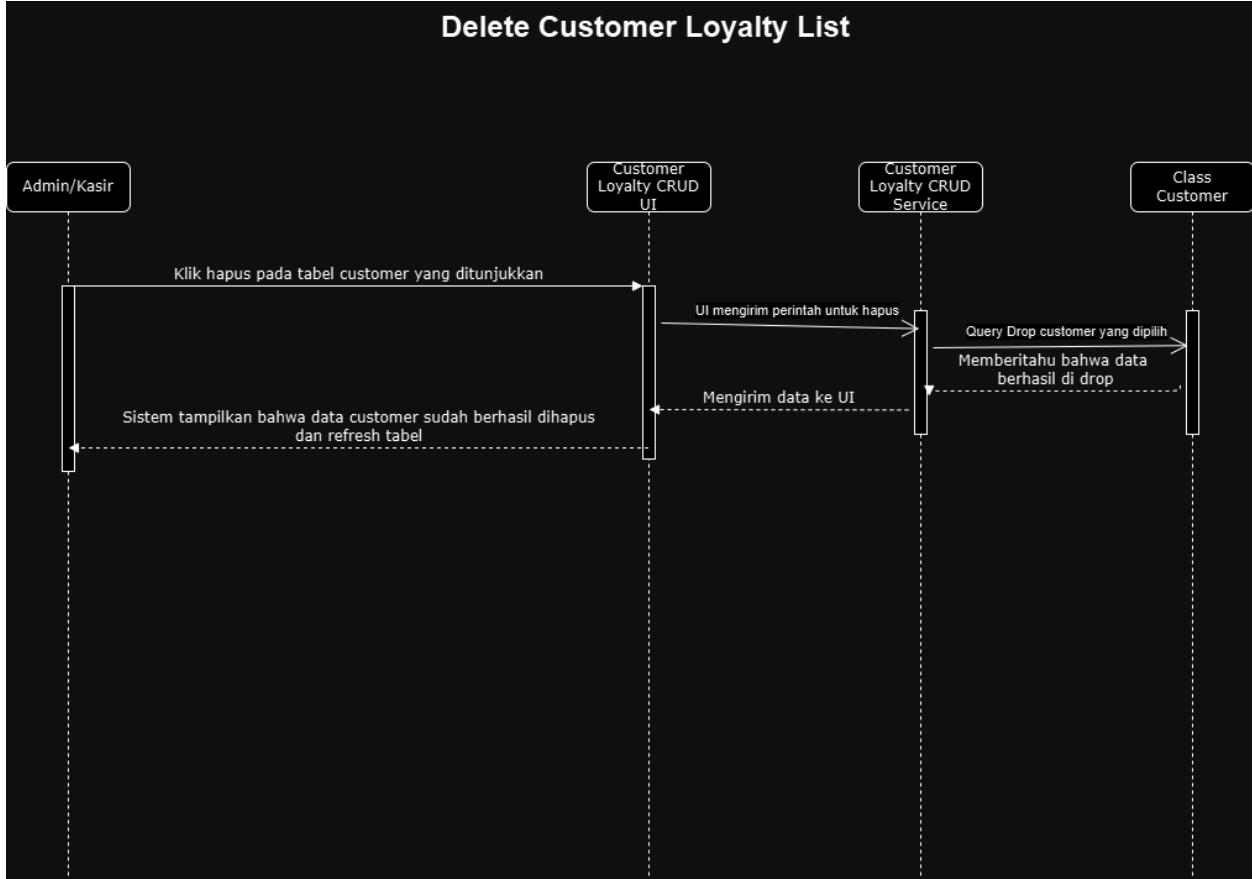
Evelin Felicia Buntaran / c14230100

SEQ\_TB.01 (CRUD Customer Loyalty List)

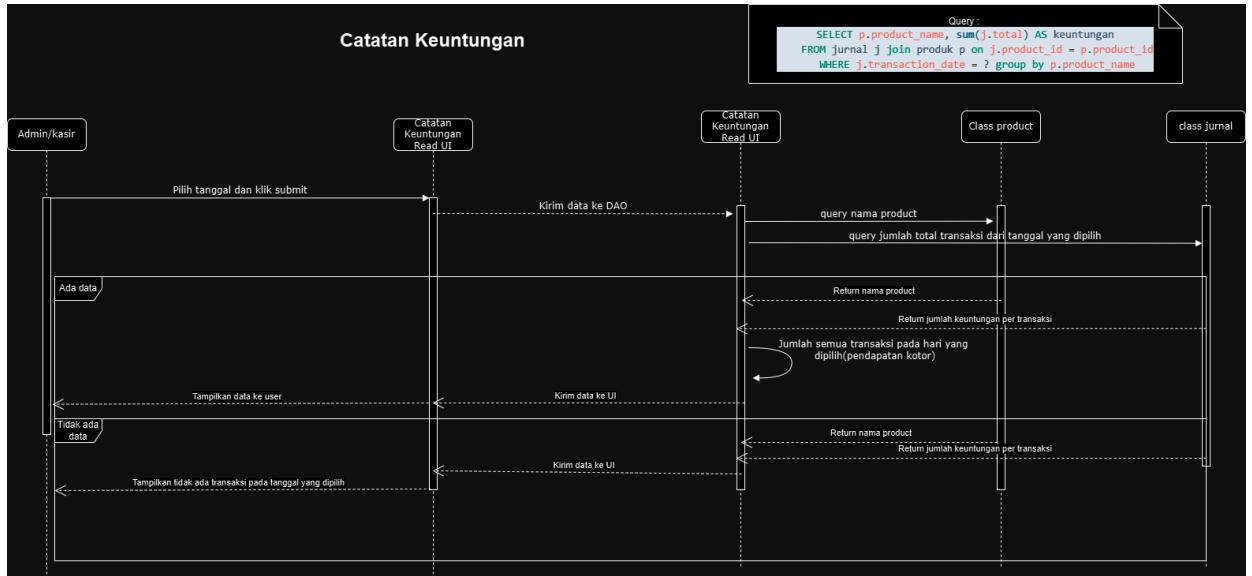




## Delete Customer Loyalty List

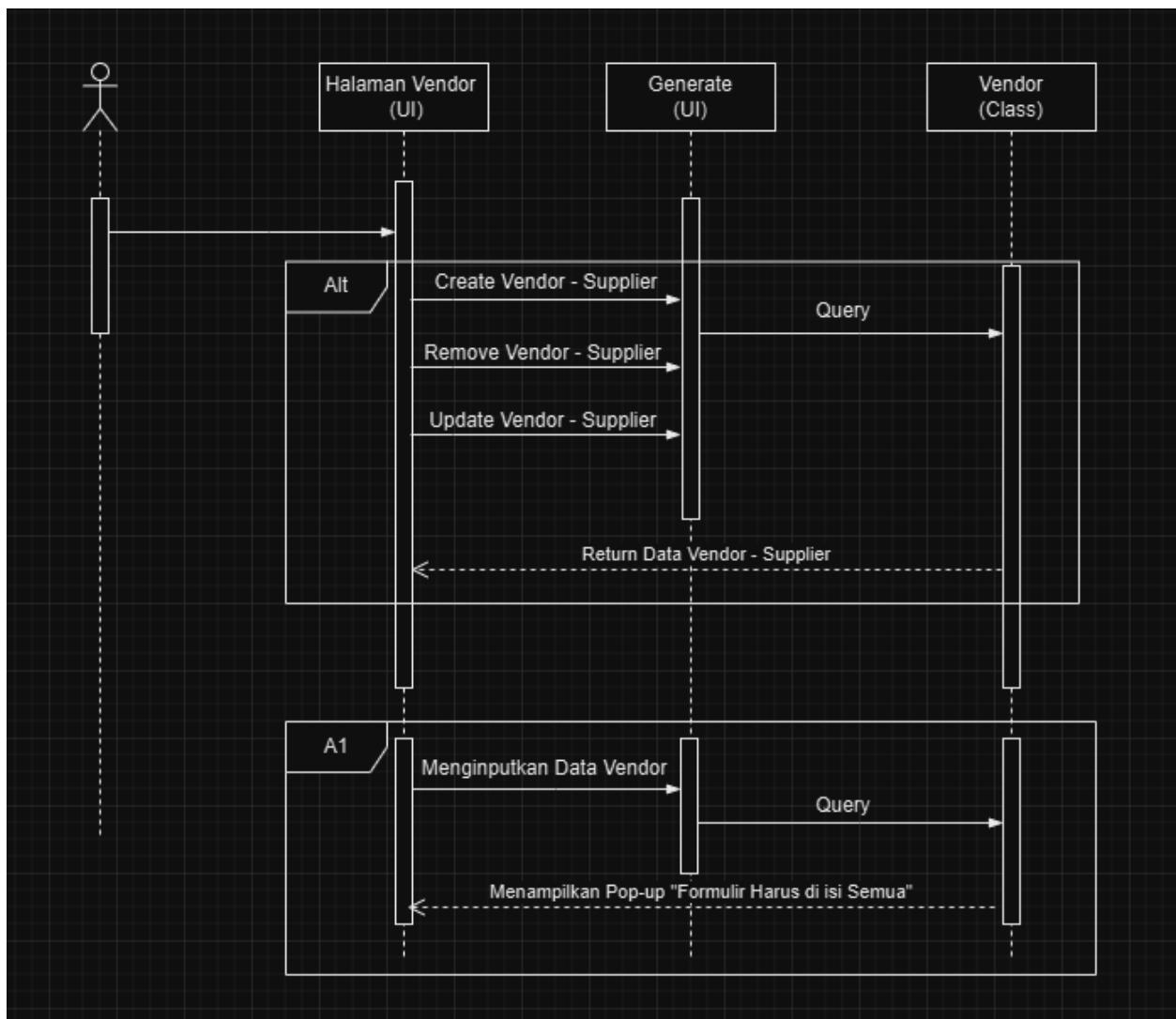


## SEQ\_TB.07 (CUD Catatan Pendapatan Keuntungan)

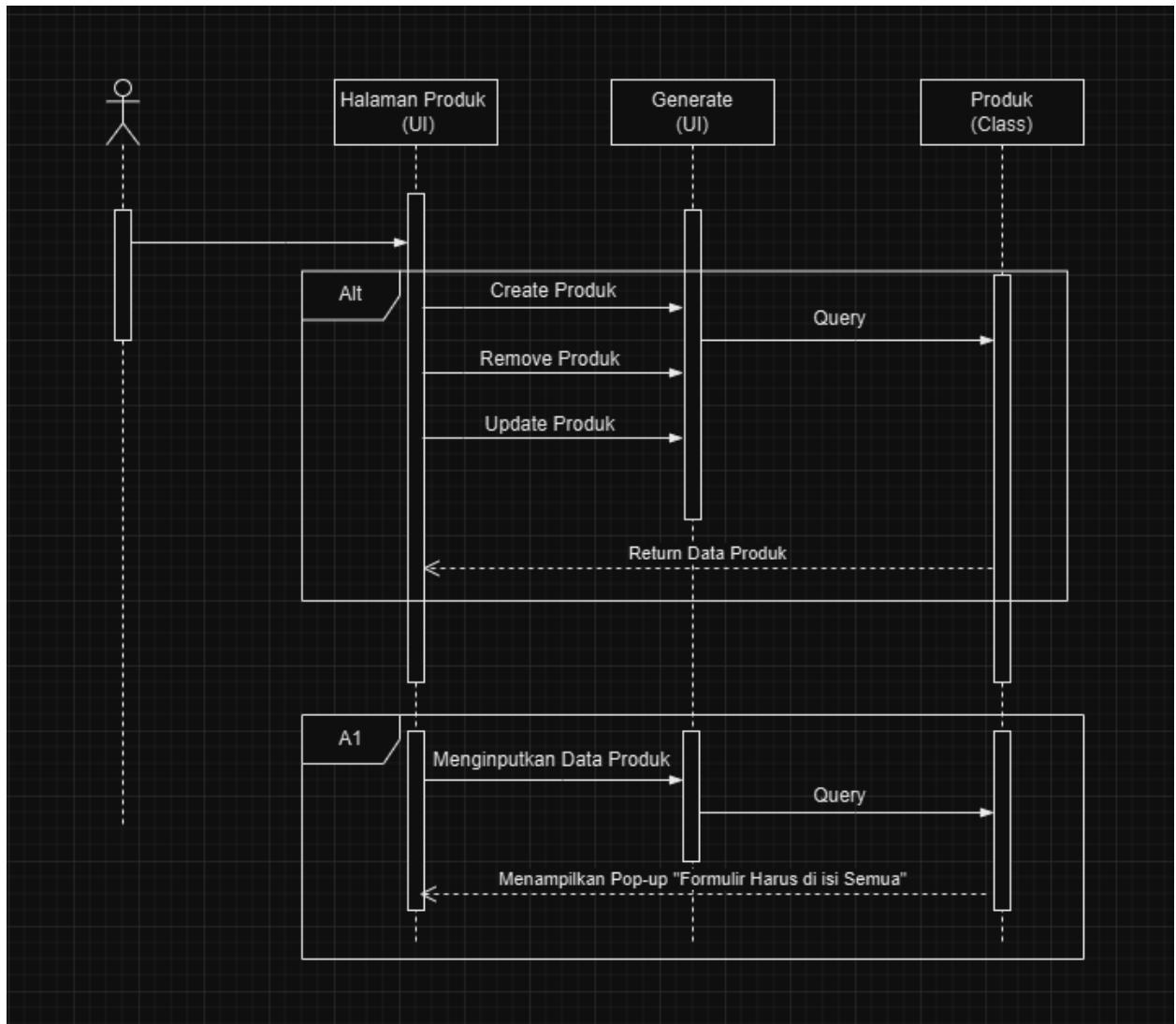


Marcel Hans Sasongko / c14230099

SEQ\_TB.10 (CRUD Vendor / Supplier)

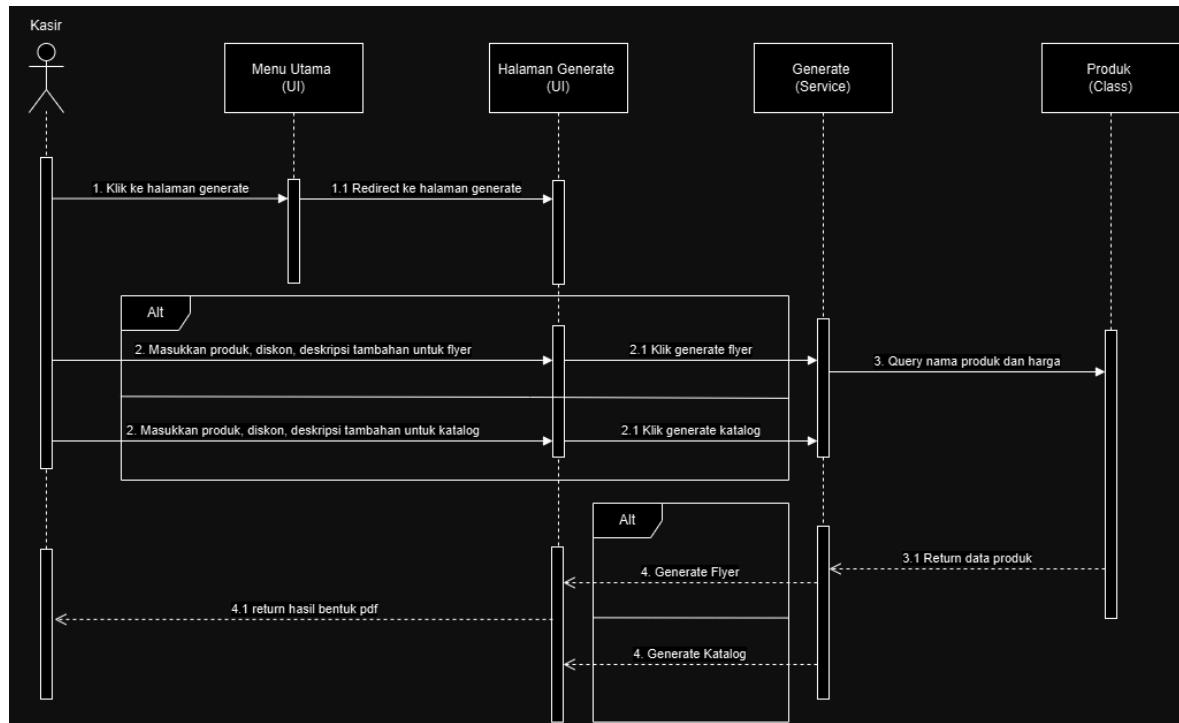


### SEQ\_TB.11 (CRUD Produk yang disuplai)

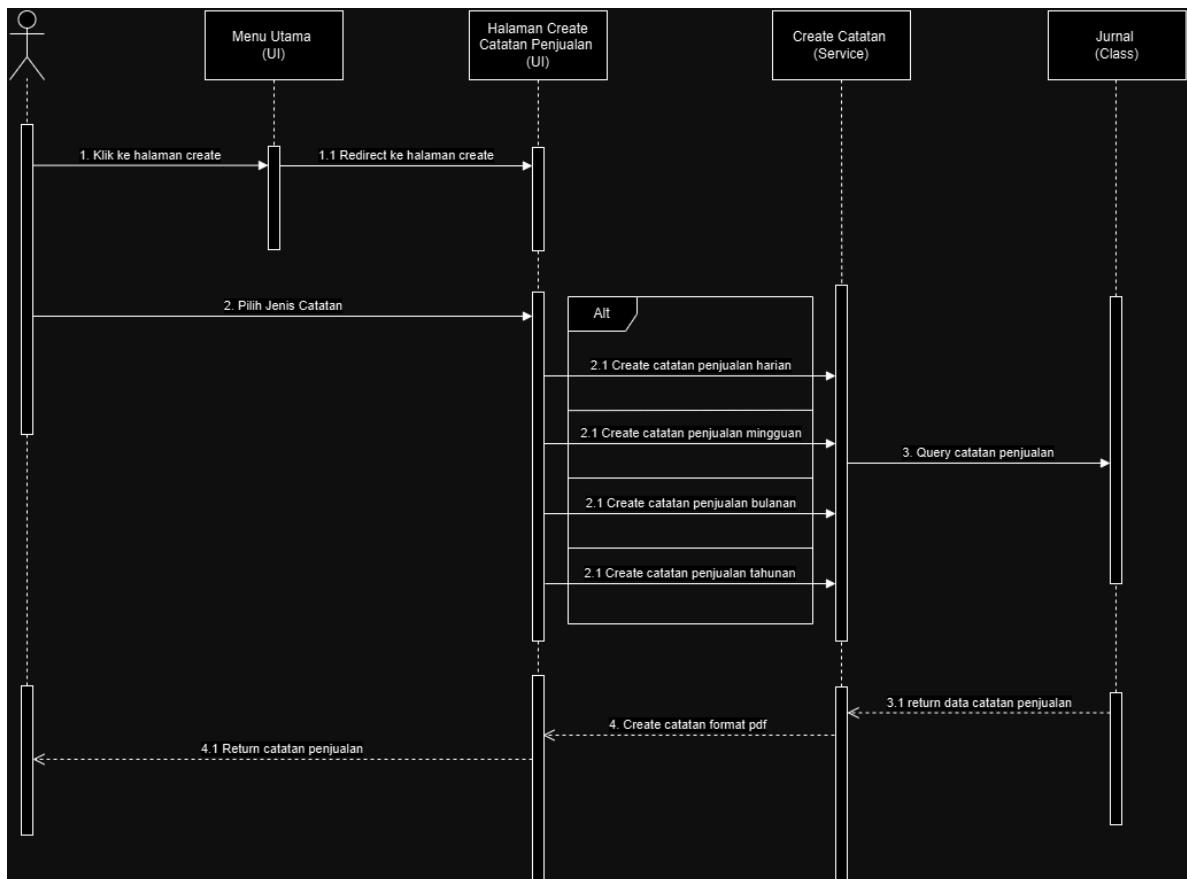


**Joseph Evan Tanujaya / c14230096**

**SEQ\_TB.02 (Generate Flyer dan Katalog)**

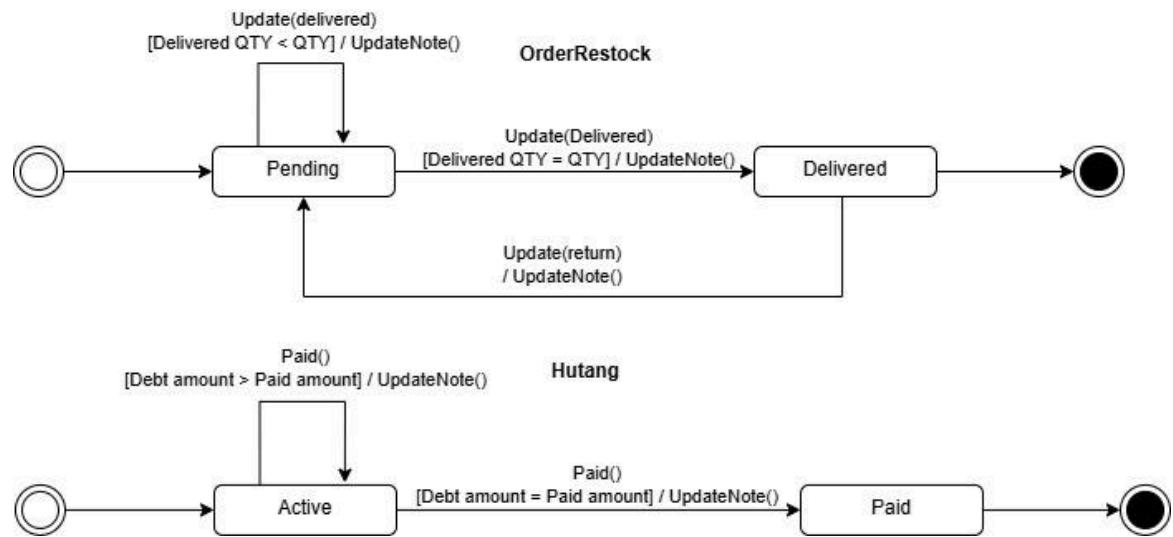


#### SEQ\_TB.04 (Create Catatan Penjualan)



e. State Diagram

S\_TB.09 & S\_TB.13



#### 4. Pemetaan Akses

Tabel akses setiap *field* dari dokumen sumber untuk masing-masing aktor. Disusun dengan menggunakan tabel untuk setiap Dokumen yang didefinisikan di Bagian 3. Menggunakan kolom Field, Function (Create, Read, Update, Delete), dan juga daftar aktor.

Contoh :

a. *DOC\_CPL*

Resource DOC_CPL			
Field	Function	KAPRODI	
id_CPL			
kode_CPL			
deskripsi_CPL_Ind	create	V	
	read	V	
	update	V	
deskripsi_CPL_Ing	create	V	
	read	V	
	update	V	
id_kurikulum	create	V	
	read	V	
	update	V	

#### 5. Design Antar-muka

Gambaran antar-muka menggunakan sketsa dasar(*wireframe*), untuk menggambarkan kebutuhan antarmuka dan hubungan nya dengan masing-masing aktor dalam sistem.

Gunakan format pengkodean untuk ID dari desain antar-muka (misal menggunakan rumus T\_<Nama Modul / Aplikasi>.<2 digit numbering>, contoh: T\_OBE.01).

## 6. Acceptance Criteria

Bagian ini merupakan bagian dari SRS nantinya yang akan dijadikan rujukan oleh tim *Quality Assurance* (QA), dalam melakukan pengujian secara detail berdasarkan kasus-kasus uji yang akan disiapkan.

Daftar persyaratan dasar produk aplikasi dapat dinyatakan berjalan dengan baik. Gunakan format untuk kode uji, misal menggunakan rumus kode sebagai berikut: UJI\_<Nama Modul/App>.<Penomoran 2 digit>, contoh: UJI\_OBE.01.

Contoh :

Kode Uji	ID Use Case	Role	Menu/Sub Menu/Fitur	Deskripsi	Hasil Yang Diharapkan
UJI_OBE.01	UC_OBE.01	KAPRODI	Tambah CPL	Pembuatan CPL dari prodi untuk kurikulum yang aktif.	CPL berhasil tersimpan dalam sistem
UJI_OBE.02	UC_OBE.02	KAPRODI	Ubah CPL	Perubahan terhadap deskripsi CPL yang telah dibuat sebelumnya.	Perubahan CPL berhasil tersimpan dalam sistem
UJI_OBE.03	UC_OBE.03	KAPRODI	Hapus CPL	Penghapusan terhadap CPL yang telah dibuat sebelumnya.	Penghapusan CPL berhasil dilakukan. Jika CPL sudah dihubungkan dengan indikator CPL dan MK maka tidak dapat dihapus.
UJI_OBE.04	UC_OBE.04	KAPRODI	Lihat CPL	Aktor melihat data CPL dari suatu kurikulum tertentu, default: kurikulum aktif	Tampilan daftar CPL