

MODUL PRAKTIKUM
ANALISIS DAN DESAIN SISTEM BERORIENTASI OBJEK



Oleh :

Ayu Ratna Juwita, M.kom.

PROGRAM STUDI TEKNIK INFORMATIKA
FAKULTAS TEKNOLOGI DAN ILMU KOMPUTER
UNIVERSITAS BUANA PERJUANGAN KARAWANG

2019

DAFTAR ISI

MODUL 1




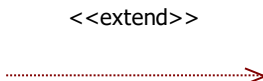
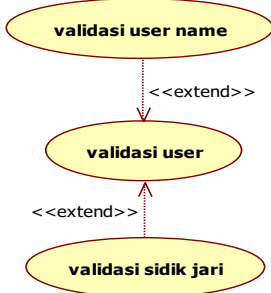

PENDAHULUAN

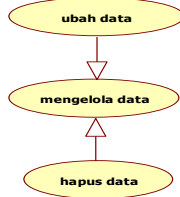

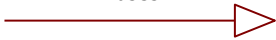
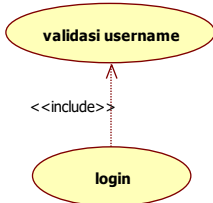
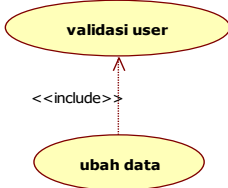
1.1. Pengenalan Unified Modeling Language (UML)

Unified Modeling Language (UML) adalah salah satu alat bantu dalam bahasa pemodelan yang digunakan untuk menspesifikasikan, memvisualisasikan, membangun, dan mendokumentasikan suatu sistem informasi.

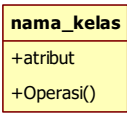




1. Diagram *Unified Modeling Language* (UML)

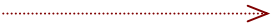

a. *Usecase Diagram*

Simbol	Deskripsi
<i>Use case</i> 	Fungsionalitas yang disediakan sistem sebagai unit-unit yang saling bertukar pesan antara unit atau aktor; biasanya dinyatakan dengan menggunakan kata kerja di awal frase nama <i>use case</i>
Aktor/ <i>actor</i> 	Orang, proses, atau sistem lain yang berinteraksi dengan aplikasi yang akan dibuat diluar aplikasi yang akan dibuat itu sendiri, jadi walaupun symbol dari aktor adalah gambar orang, tapi aktor belum tentu merupakan orang; biasanya dinyatakan menggunakan kata benda di frase nama aktor
Asosiasi/ <i>association</i> 	Komunikasi antara aktor dan <i>use case</i> yang berpartisipasi pada <i>use case</i> atau <i>use case</i> memiliki interaksi dengan aktor
Ekstensi/ <i>extend</i> 	<p>Relasi <i>use case</i> tambahan ke sebuah <i>use case</i> dimana <i>use case</i> yang ditambahkan dapat berdiri sendiri walau tanpa <i>use case</i> tambahan itu; mirip dengan prinsip inheritance pada pemrograman berorientasi objek; biasanya <i>use case</i> tambahan memiliki nama depan yang sama dengan <i>use case</i> yang ditambahkan, misalnya :</p> <div style="text-align: center;">  </div> <p>arah panah mengarah pada <i>use case</i> yang menjadi generalisasinya (umum)</p>
Generalisasi/ <i>Generalization</i> 	Hubungan generalisasi dan spesialisasi (umum-khusus) anantara dua buah <i>use case</i> dimana fungsi yang satu adalah fungsi yang lebih umum dari lainnya, misalnya :


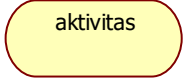
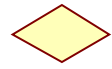


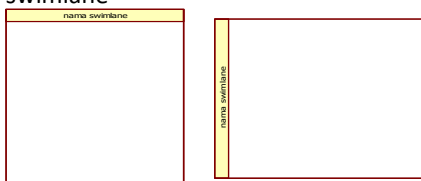
	 <p>Arah panah mengarah pada <i>use case</i> yang menjadi generalisasinya (umum)</p>
<p>Menggunakan/<i>include/uses</i></p> <p><code><<include>></code></p>  <p><code>uses</code></p> 	<p>Relasi <i>use case</i> tambahan ke sebuah <i>use case</i> dimana <i>use case</i> yang ditambahkan memerlukan <i>use case</i> ini untuk menjalankan fungsinya sebagai syarat dijalankan <i>use case</i> ini.</p> <p>Ada dua sudut pandang yang cukup besar mengenai include di <i>use case</i> :</p> <ul style="list-style-type: none"> include berarti <i>use case</i> yang ditambahkan akan selalu dipanggil saat <i>use case</i> tambahan dijalankan, misalkan pada kasus berikut :  <ul style="list-style-type: none"> include berarti <i>use case</i> yang ditambahkan akan selalu melakukan pengecekan apakah <i>use case</i> yang ditambahkan telah dijalankan sebelum <i>use case</i> tambahan dijalankan, misalkan pada kasus berikut : 

b. Class Diagram


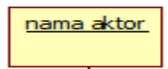

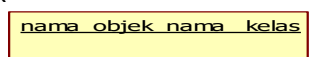

Simbol	Deskripsi
<p>Kelas</p> 	Kelas pada struktur sistem
<p>Antarmuka/<i>interface</i></p>  <p>nama_interface</p>	Sama dengan konsep <i>interface</i> dalam pemrograman berorientasi objek.
<p>Asosiasi/<i>association</i></p> 	Relasi antar kelas dengan makna umum, asosiasi biasanya juga disertai dengan <i>multiplicity</i> .
<p>Asosiasi berarah/<i>directed association</i></p> 	Relasi antar kelas dengan makna kelas yang satu digunakan oleh kelas yang lain, asosiasi biasanya juga disertai dengan <i>multiplicity</i>
<p>Generalisasi</p> 	Relasi antar kelas dengan makna generalisasi-spesialisasi (umum-khusus)

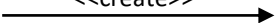




Menggunakan/ <i>include/uses</i> 	Relasi antar kelas dengan makna kebergantungan antar kelas
Agregasi / <i>aggregation</i> 	Relasi antar kelas dengan makna semua-bagian (<i>whole-part</i>)

c. Activity Diagram

Simbol	Deskripsi
Status awal 	Status awal aktivitas sistem, sebuah diagram aktivitas memiliki sebuah status awal
Aktivitas 	Aktivitas yang dilakukan sistem, aktivitas biasanya diawali dengan kata kerja
Percabangan / <i>decision</i> 	Asosiasi percabangan dimana jika ada pilihan aktivitas lebih dari satu
Penggabungan / <i>join</i> 	Asosiasi penggabungan dimana lebih dari satu aktivitas digabungkan menjadi satu
Status akhir 	Status akhir yang dilakukan sistem, sebuah diagram aktivitas memiliki sebuah status akhir
swimlane 	Memisahkan organisasi bisnis yang bertanggung jawab terhadap aktivitas yang terjadi

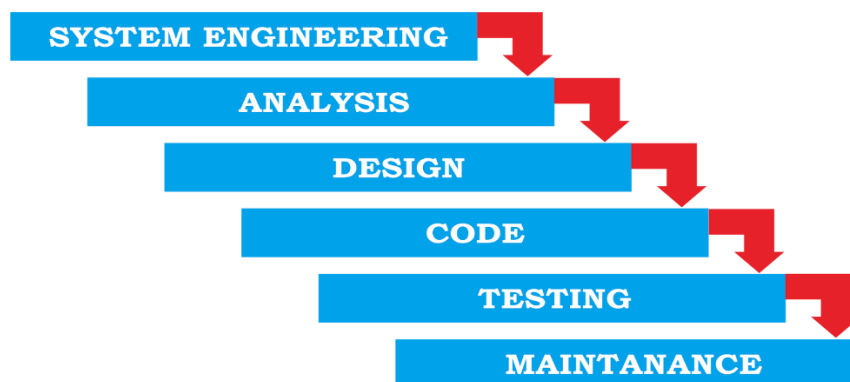
d. Sequence Diagram

Simbol	Deskripsi
Aktor  nama aktor atau  tanpa waktu aktif	Orang, proses, atau sistem lain yang berinteraksi dengan aplikasi yang akan dibuat di luar aplikasi yang akan dibuat itu sendiri, jadi walaupun symbol dari aktor adalah gambar orang, biasanya dinyatakan menggunakan kata benda di awal frase nama aktor
Garis hidup / <i>line life</i> 	Menyatakan kehidupan suatu objek
Objek 	Nama objek yang berinteraksi pesan
Waktu aktif 	Asosiasi penggabungan dimana lebih dari satu aktivitas digabungkan menjadi satu

Pesan tipe <i>create</i> <div style="text-align: center;"> <<create>>  </div>	Menyatukan suatu objek membuat objek yang lain, arah panah mengarah pada objek yang dibuat
Pesan tipe <i>call</i> 1: nama_metode () 	Menyatakan suatu objek memanggil operasi/metode yang ada pada objek lain atau dirinya sendiri
Pesan tipe <i>send</i> 1 : masukan 	Menyatakan bahwa suatu objek mengirimkan data/masukan/informasi ke objek lainnya, arah panah mengarah pada objek yang dikirim
Pesan tipe <i>return</i> 1 : keluaran 	Menyatakan bahwa suatu objek yang telah menjalankan suatu operasi atau metode menghasilkan suatu kembalian ke objek tertentu, arah panah mengarah pada objek yang menerima kembalian
Pesan tipe <i>destroy</i> 1 : keluaran 	Menyatakan bahwa suatu objek mengakhiri hidup objek yang lain, arah panah mengarah pada objek yang diakhiri, sebaiknya jika ada <i>create</i> maka ada <i>destroy</i>

1.2. Metodologi SDLC

Metode yang digunakan untuk pengembangan proyek ini menggunakan metode rekayasa dan menggunakan *Software Development Life Cycle* (SDLC) dalam pengembangan aplikasi sistem transaksi *laundry*, berikut tahapan dari metodologi SDLC :



Gambar. 1. 1 Tahapan *Software Development Life Cycle* (SDLC)

1. System Engineering

Pada tahapan ini sistem engineer adalah bertugas dalam pendefinisian permintaan (*requirement definition*) satu permasalahan kemudian menginvestigasikan masalah tersebut dengan menganalisa yang tepat kemudian mendesain solusi yang bertujuan untuk mendesain lebih detail.

2. Analysis and definition

Sebelum melakukan design sistem kali ini kami melakukan analisa kebutuhan sistem pada Sistem Transaksi *Laundry*, sistem yang dibutuhkan pada sebagai berikut:

a. Pengguna *Owner*

- 1) *Owner* dapat mengelola Data *Login*
- 2) *Owner* dapat mengelola Data Jasa *Laundry*
- 3) *Owner* dapat mengelola Data Pelanggan *Laundry*
- 4) *Owner* dapat mengelola Data Transaksi Masuk
- 5) *Owner* dapat mengelola Data Transaksi Keluar
- 6) *Owner* dapat mengelola Data Laporan

b. Petugas Penerima dan Pengembalian *Laundry*

Penerima dan Pengembalian *Laundry* hanya sebagai konsumen saja dimana akan memperoleh nota untuk pengambilan barang

3. *Desain System*

Tahapan selanjutnya dari Model SDLC adalah tahapan System Design. Desain yang akan dibangun menggunakan UML (*Unified Modeling Language*) untuk mendesain sistem.

UML tersebut antara lain :

1. Use Case Diagram
2. Class Diagram
3. Activity Diagram
4. Sequence Diagram

4. *Code*

Dalam tahapan ini membuat code program dengan menggunakan bahasa pemrograman java untuk mengimplementasikan desain sUML pada sistem *Laundry*.

5. *Testing dan Implementasi*

Pada tahapan ini akan melakukan Implementasi dan *Testing*, dalam perancangan sistem yang telah dibuat sebelumnya dengan menggunakan aplikasi desktop dengan bahasa pemrograman java. untuk merancang program dan untuk database dibangun dengan menggunakan PhpMyadmin. Pada *System Testing* dalam tahapan ini akan melakukan Integrasi terhadap setiap *form* yang ada di Sistem Transaksi *Laundry*.

6. *Maintenance*

Pada tahapan Maintenance, dalam tahapan ini pengembang melakukan pengoperasian program dan maintenance program.

1.3. Introduction

Dokumen *Software Requirements Specification (SRS)* ini memberikan gambaran tentang tujuan, ruang lingkup, definisi, singkatan-singkatan, referensi dan gambaran secara keseluruhan dari perangkat lunak.

1.4. Purpose

Tujuan dari dokumen ini adalah untuk mengumpulkan, menganalisis dan memberikan wawasan mendalam dari Sistem Transaksi *Laundry*, lengkap dengan mendefinisikan pernyataan masalah secara rinci. Secara garis besar, dokumen ini menyajikan :

- a. Deskripsi tentang lingkungan produk yang akan digunakan
- b. Deskripsi tentang kemampuan sistem.
- c. Persyaratan spesifikasi sistem yang digunakan untuk mengoperasikan produk sistem.

SRS ini memungkinkan dalam pemahaman tentang apa yang diharapkan dari sistem yang baru dan yang akan dibangun. Pemahaman yang jelas mengenai sistem dan fungsionalitas akan memungkinkan dikembangkannya produk yang tepat bagi pengguna. SRS ini dapat digunakan sebagai dasar dalam pengembangan proyek, dimana Sistem Transaksi Laundry dapat dirancang, dibangun dan diuji.

Dokumen ini ditujukan pada pengguna dan pihak pengembang. Pembaca diasumsikan memiliki pengetahuan tentang Sistem Transaksi Laundry serta pengetahuan dan pemahaman tentang *Unified Modeling Language (UML)* diagram.

1.5. Scope

Produk yang akan dibuat dalam dokumen ini merupakan bagian dari ruang lingkup kebutuhan pembangunan perangkat lunak yang berupa aplikasi yang digunakan untuk pengelolaan Sistem Transaksi Laundry, yang diantaranya aplikasi tersebut mampu menangani mengelola data pelanggan, data jasa, data cucian masuk, data cucian keluar. Serta mencetak laporan

1.6. Definition, Acronyms, and Abbreviation

SRS (<i>Software Requirement Specification</i>)	SRS adalah dokumen yang menjelaskan tentang kebutuhan fungsional maupun non-fungsional sistem perangkat lunak yang akan dikembangkan.
UML (<i>Unified Modeling Language</i>)	Bahasa spesifikasi standar untuk mendokumentasikan, menspesifikasikan, dan membangun sistem perangkat lunak.

MODUL 2

OVERALL DESCRIPTION

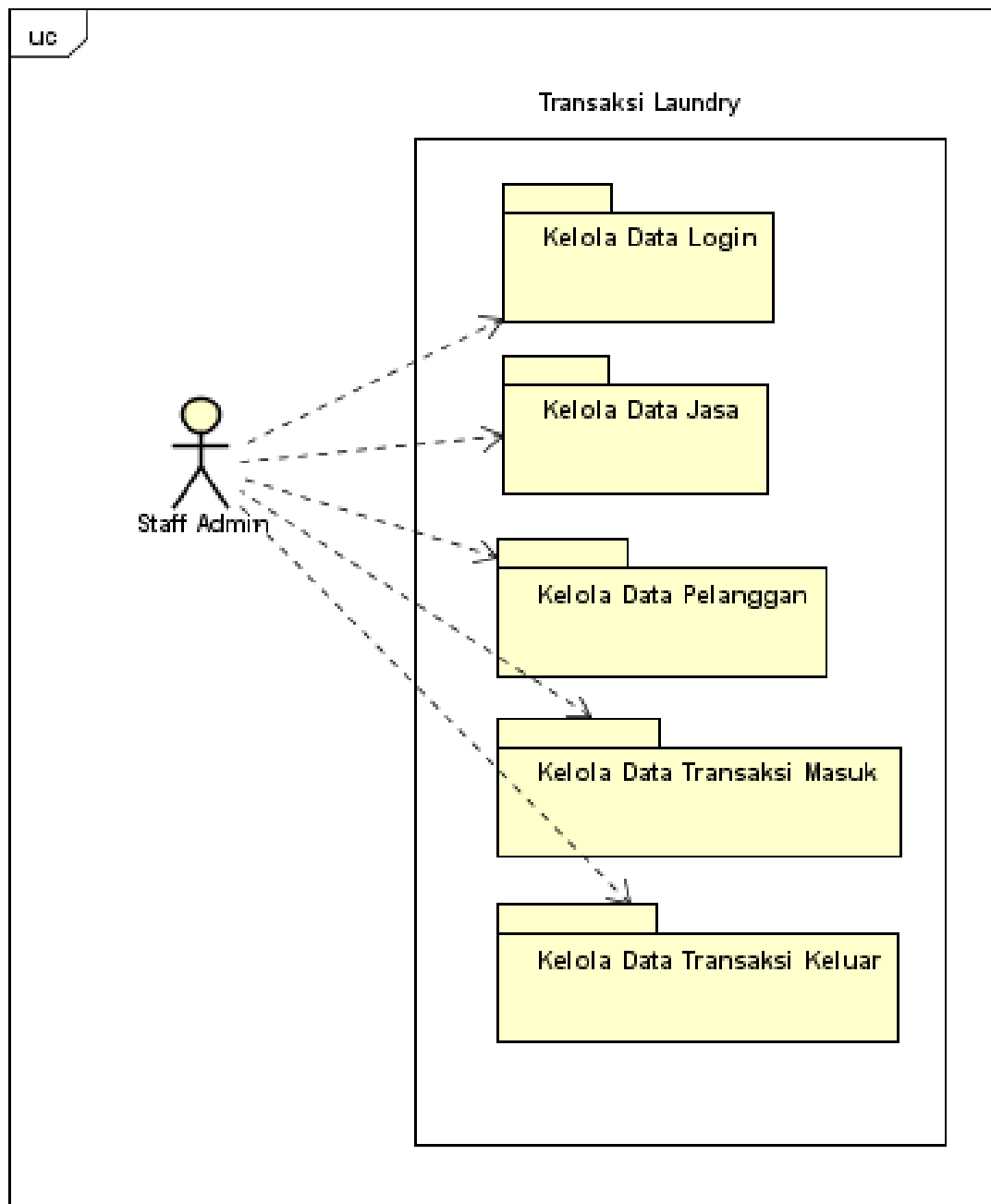
2.1. Product Perspective

Perangkat lunak sistem transaksi *laundry* ini merupakan perangkat lunak yang digunakan untuk mempermudah proses transaksi *laundry*. Aplikasi transaksi *laundry* berkaitan dengan beberapa entitas luar, yaitu pelanggan dan admin atau pemilik. Sistem transaksi *laundry* merupakan suatu aplikasi yang mengolah data pelanggan dan data jasa saat melakukan transaksi dengan memasukkan kode pelanggan dan kode jasa.

Pelanggan mendapatkan informasi mengenai total kuantiti dan total harga serta waktu selesai pengerjaan *laundry* dan pelanggan akan mendapatkan *discount* harga dari total *point* yang dikumpulkan, setelah admin mengisi *form* pelanggan dan mengisi *form* transaksi untuk mentransaksikan pakaian *laundry*.

2.2. Product Functions

Dengan mengunjungi tempat *laundry*, pelanggan akan ditanyakan informasi data diri untuk diinputkan ke dalam *form* data pelanggan oleh *admin/pemilik laundry*. *Admin* akan mentransaksikan pakaian ke *form* cucian masuk dengan mengisi kode pelanggan, kode jasa sesuai keinginan pelanggan dimana jasa tersebut sudah ditentukan oleh pemilik *laundry*, waktu pengerjaan serta total kg pakaian serta pelanggan akan mendapatkan poin setiap transaksi pakaian. Transaksi dikatakan sukses jika admin berhasil memasukkan data transaksi. Setelah berhasil mengisi data transaksi cucian masuk, *admin* juga akan mengisi *form* cucian keluar berdasarkan *no order* cucian masuk .



No.	Aktor	Deskripsi
1	Staff Administrasi / pemilik	Seseorang yang bertugas untuk mengelola data pelanggan, jasa, cucian masuk dan cucian keluar

2.3. User Characteristic

Dalam Rancang Bangun Sistem Transaksi *Laundry*, karakteristik masing-masing *user* sebagai berikut :

1. Staff Administrasi

Staff administrasi mempunyai hak untuk nota yang akan diterbitkan, mengelola data pelanggan, mengelola data jasa, mengelola data cucian masuk, mengelola data cucian keluar dan membuat laporan.

2. Pelanggan

Pelanggan hanya memiliki tidak memiliki hak akses namu hanya melakukan pemesanan *laundry* pakaian saja.

2.4. Constraints

Batasan proyek sistem Transaksi Laundry dalam dokumen SRS ini adalah sebagai berikut:

1. Sistem Pemesanan ini mengakomodir pelanggan untuk tidak menggunakan layanan Transaksi Laundry, namun hanya pelanggan yang akan melakukan pemesanan.
2. Sistem Pemesanan ini mengakomodir pemilik atau staff administrasi atau staff pelayanan untuk mengelola .

2.5. Assumptionts and Depedencies

Asumsi dan ketergantungan yang digunakan dalam proyek ini adalah:

1. Semua manajemen terkait transaksi *laundry* hanya dapat dibuat dan dikelola oleh staff administrasi atau pemilik *laundry* dengan persetujuan bersama.
2. Tidak ada *trainning* program bagi *user* (*user* dianggap sudah mengerti dan dapat mengoperasikan program dengan baik).
3. Perangkat keras yang dibutuhkan untuk operasional program telah disediakan oleh pihak pemilik *laundry*.
4. Segala lisensi *software* ditanggung oleh pihak *laundry*.
5. Sistem Operasi yang digunakan minimal Windows 7.

2.6. Specific Requirements

Berikut adalah kebutuhan perangkat lunak untuk perancangan system dan petugas penguji dalam melakukan verifikasi Sehingga diperlukannya suatu pengolahan data-data yang diproses secara komputerisasi guna mendapatkan informasi-informasi yang berguna.

Bagian ini berisi semua persyaratan fungsional dan kualitas dari produk. Hal ini memberikan penjelasan secara rinci tentang sistem dan semua fitur-fiturnya.

2.7. Functionality

No.	Fungsi	Deskripsi
1	Login	Proses login ke sistem transaksi laundry.
2	Entri Data Jasa	Proses untuk menambahkan data jasa.
3	Update Data Jasa	Proses untuk mengedit data jasa.
4	Delete Data Jasa	Proses untuk menghapus data jasa
5	Batalkan Data Jasa	Proses untuk membatalkan proses data jasa.
6	Entri Data Pelanggan	Proses untuk menambahkan data pelanggan.
7	Update Data Pelanggan	Proses untuk mengedit data pelanggan.
8	Delete Data Pelanggan	Proses untuk menghapus data pelanggan
9	Batalkan Data Pelanggan	Proses untuk membatalkan proses data pelanggan.
10	Entri Data Cucian Masuk	Proses untuk menambahkan data cucian masuk.
11	Update Data Cucian Masuk	Proses untuk mengedit data cucian masuk.
12	Delete Data Cucian Masuk	Proses untuk menghapus data cucian masuk
13	Batalkan Data Cucian Masuk	Proses untuk membatalkan proses data cucian masuk.
14	Cek Poin	Proses untuk pengecekan poin
15	Cetak Nota Cucian Masuk	Proses untuk mencetak nota cucian masuk
16	Cetak Laporan Data Cucian Masuk	Proses untuk mencetak laporan cucian masuk
17	Entri Data Cucian Keluar	Proses untuk menambahkan data cucian keluar.
18	Update Data Cucian Keluar	Proses untuk mengedit data cucian keluar.
19	Delete Data Cucian Keluar	Proses untuk menghapus data cucian keluar
20	Batalkan Data Cucian Keluar	Proses untuk membatalkan proses data cucian keluar.
21	Cek Discount	Proses untuk pengecekan discount potongan harga
22	Cetak Nota Cucian Keluar	Proses untuk mencetak nota cucian keluar.
23	Cetak Laporan Data Cucian Keluar	Proses untuk mencetak laporan cucian keluar.

2.8. Hardware Interfaces

Hardware Interface yang dibutuhkan untuk membantu kelengkapan dari pembangunan sistem yang sedang dirancang pada umumnya hanya berupa komputer.

Karena aplikasi ini tidak harus berjalan melalui internet, sehingga tidak perlu menggunakan internet. Dan koneksi dari komputer ke server database dikelola oleh sistem operasi yang mendukung phpmyadmin atau localhost dengan bantuan XAMPP..

2.9. Performance Requirements

Pada dasarnya kinerja dalam penggunaan produk akan tergantung pada komponen hardware dan koneksi localhost yang digunakan oleh client/user. Karena produk yang dihasilkan merupakan aplikasi desktop. Produk ini memerlukan waktu pada saat memuat halaman awal tergantung pada kecepatan komputer pada saat dijalankan.

2.10. Software System Quality Attributes

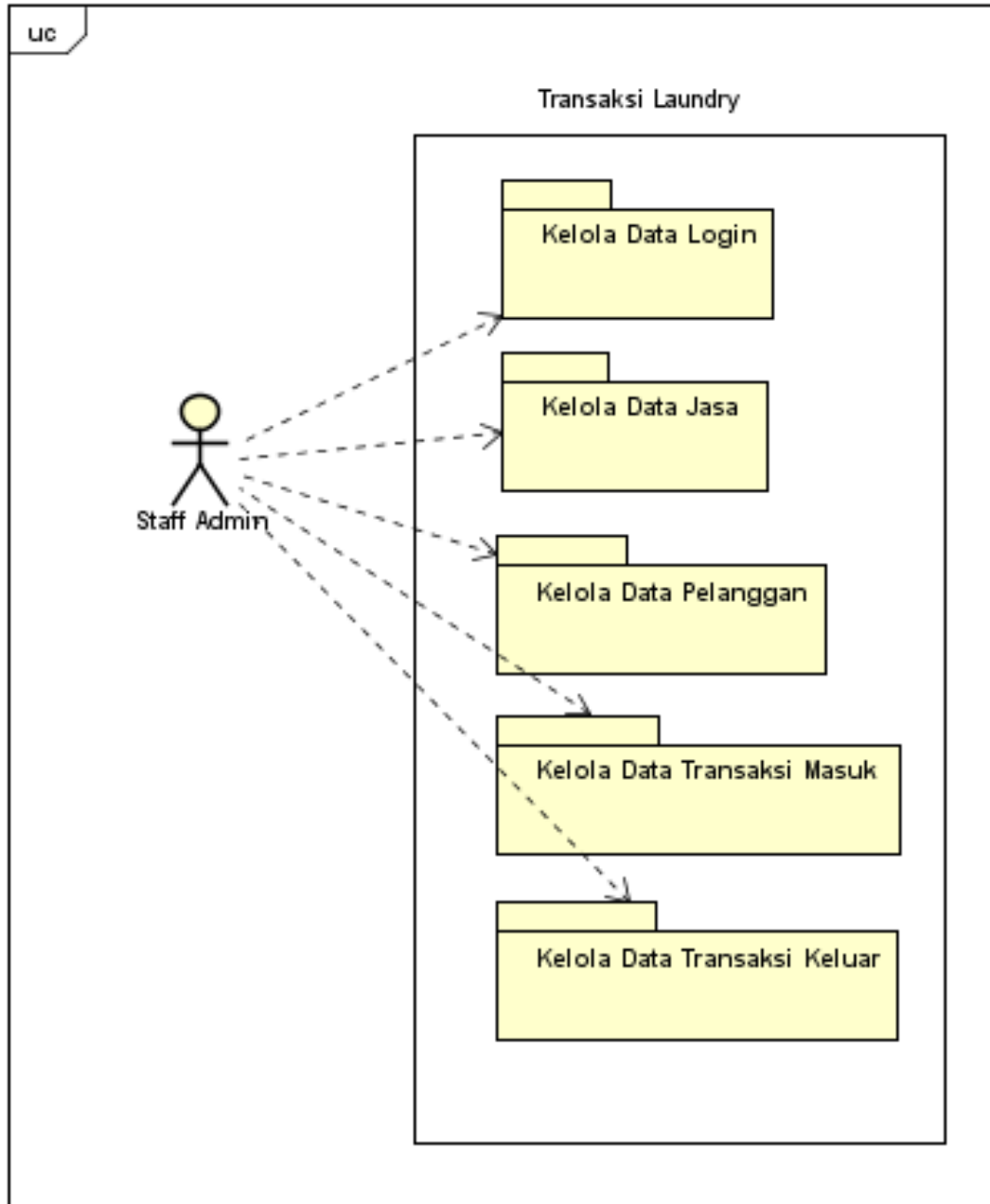
Persyaratan di bagian ini menentukan reliability, availability, security dan maintainability yang diperlukan software system.

- Usability : Untuk mengakses sistem ini, user dapat menggunakan aplikasi dekstop Sistem Transaksi Laundry. Sistem dapat di akses selama 7 X 24 jam, kecuali saat maintenance/ perbaikan sistem
- Performance : Kinerja dalam penggunaan sistem akan tergantung pada komponen perangkat keras dan processor yang digunakan oleh client/user.
- Supportability : Adanya dukungan secara teknis oleh petugas operasional, dalam kaitan melakukan panduan atas adanya permasalahan dalam proses penggunaan Sistem.
- Portability : Sistem ini berjalan pada platform atau sistem operasi apa saja yang mendukung database phpmyadmin / localhost.
- Legalitas, Copyright dan Other Notices : Hak cipta perangkat lunak sistem transaksi laundry menjadi milik pengembang proyek dan pihak jasa laundry. Masing masing pihak tidak dapat mendistribusikan perangkat lunak kepada pihak lain tanpa adanya kesepakatan bersama.

MODUL 3

USE CASE DIAGRAM

3.1. UC Transaksi Laundry

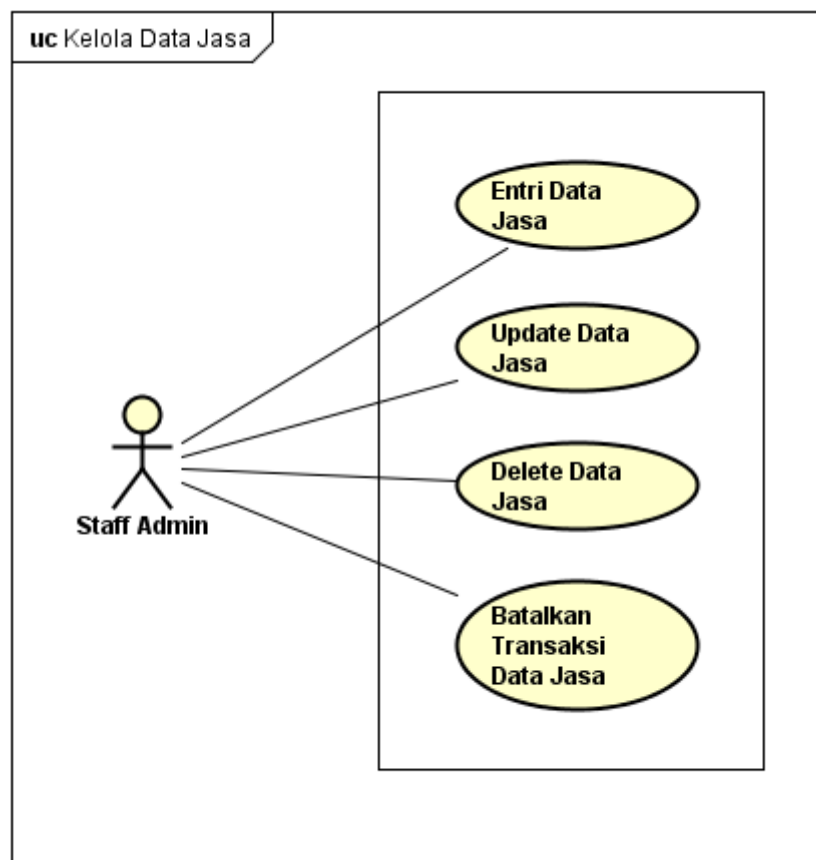


3.2. UC Data Login

Use Case Name	:	Login
Triger Event	:	User memilih menu login pada halaman utama
Aktor	:	Admin, Pelanggan
Pre-conditon	:	Aktor telah membuka halaman login

Post-condition	:	Aktor berada di halaman home sesuai hak akses	
Normal Course		Aktor	Sistem
		1. Mengisi data sesuai kolom yang tersedia	
			2. Validasi kelengkapan data pada form login
			3. Sistem akan menampilkan pesan berhasil login
			4. Menampilkan halaman home sesuai hak aksesnya
Alternative Course	:	Aktor	Sistem
			3a. Muncul pesan karena data yang diisi tidak sesuai dengan form
		3b. User dapat memperbaiki data yang salah dan melanjutkan login	
Exception	:	User dapat membatalkan login dengan klik button batal	

3.3. UC Data Jasa



Use Case Name	:	Entry Data Jasa	
Triger Event	:	Aktor memilih menu data jasa	
Aktor	:	Admin	
Pre-conditon	:	<ul style="list-style-type: none"> - Aktor telah login - Aktor telah membuka halaman data jasa 	
Post-condition	:	Aktor telah menambah data jasa dan data telah tersimpan di database	
Normal Course		Aktor	Sistem
		1. Memilih menu data jasa	
			2. Menampilkan form data jasa
		3. Mengisi form jasa	
		4. Memilih tombol simpan	
			5. Melakukan validasi data
			6. Jika data berhasil divalidasi maka tampil pesan data berhasil ditambahkan
		Aktor	Sistem
			8a. Jika data gagal divalidasi maka tampil pesan data gagal ditambahkan
Alternative Course	:	Aktor dapat membatalkan tambah data jasa dengan mengklik tombol exit pilih menu keluar	
Exception	:		

Use Case Name	:	Update Data Jasa	
Aktor	:	Admin	
Triger Event	:	Aktor memilih menu data jasa	
Pre-conditon	:	<ul style="list-style-type: none"> - Aktor telah login - Aktor telah membuka halaman data jasa 	
Post-condition	:	Aktor telah menambah data jasa dan data telah tersimpan di database	
Normal Course		Aktor	Sistem
		1. Memilih menu data jasa	
			2. Menampilkan data jasa
		3. Memilih nama jasa yang akan diupdate pada tabel jasa	
			4. Menampilkan data jasa sesuai pencarian
		5. Mengubah form jasa yang akan di update	
		6. Klik tombol update	

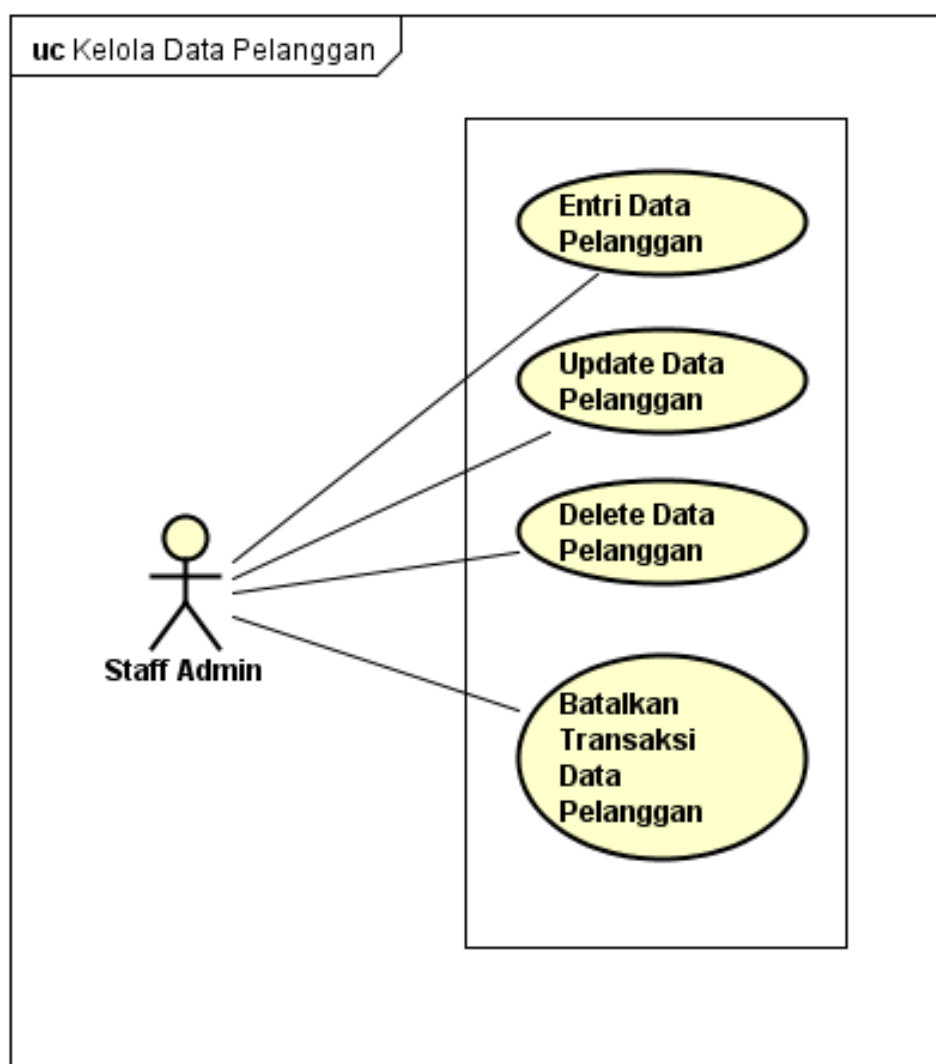
			7. Validasi data
			8. Jika validasi berhasil maka tampil pesan data berhasil diupdate
Alternative Course	:	Aktor	Sistem
			8a. Jika validasi gagal maka tampil pesan data gagal diupdate
Exception	:	Aktor dapat membatalkan update data jasa dengan klik tombol exit	

Use Case Name	:	Delete Data Jasa	
Aktor	:	Admin	
Triger Event	:	Aktor memilih menu data jasa	
Pre-conditon	:	- Aktor telah melakukan login - Aktor telah membuka halaman data jasa	
Post-condition	:	Aktor telah menghapus data jasa yang ada pada sistem transaksi laundry	
Normal Course		Aktor	Sistem
		1. Memilih menu data jasa	
			2. Menampilkan halaman data jasa
		3. Mengisi id jasa pada kolom id jasa	
		4. Klik tombol hapus	
			5. Validasi data
			6. Jika validasi berhasil maka tampil pesan data berhasil dihapus
Alternative Course	:	Aktor	Sistem
			6a. Jika validasi gagal maka tampil pesan data gagal dihapus
Exception	:	Aktor dapat membatalkan delete data jasa dengan klik tombol exit atau pilih menu keluar	

Use Case Name	:	Batalkan Data Jasa	
Aktor	:	Admin	
Triger Event	:	Aktor memilih menu data jasa	
Pre-conditon	:	- Aktor telah melakukan login - Aktor telah membuka halaman data jasa	
Post-condition	:	Aktor telah membatalkan entri data jasa	
Normal Course		Aktor	Sistem

		1. Memilih menu data jasa	
			2. Menampilkan halaman data jasa
		3. Klik tombol batal	
			4. Menampilkan Form Data jasa kosong pada tiap kolom
Alternative Course	:	Aktor	Sistem
Exception	:	Aktor dapat membatalkan data jasa dengan klik tombol exit atau menutup browser	

3.4. UC Data Pelanggan



Use Case Name	:	Entry Data Pelanggan
Triger Event	:	Aktor memilih menu data pelanggan
Aktor	:	Admin
Pre-conditon	:	- Aktor telah login

		- Aktor telah membuka halaman data pelanggan	
Post-condition	:	Aktor telah menambah data pelanggan dan data telah tersimpan di database	
Normal Course		Aktor	Sistem
		1. Memilih menu data pelanggan	
			2. Menampilkan form data pelanggan
		3. Mengisi form pelanggan	
		4. Memilih tombol simpan	
			5. Melakukan validasi data
			6. Jika data berhasil divalidasi maka tampil pesan data berhasil ditambahkan
		Aktor	Sistem
			8a. Jika data gagal divalidasi maka tampil pesan data gagal ditambahkan
Alternative Course	:	Aktor dapat membatalkan tambah data pelanggan dengan mengklik tombol exit pilih menu keluar	
Exception	:		
Use Case Name	:	Update Data Pelanggan	
Aktor	:	Admin	
Triger Event	:	Aktor memilih menu data pelanggan	
Pre-conditon	:	- Aktor telah login - Aktor telah membuka halaman data pelanggan	
Post-condition	:	Aktor telah menambah data pelanggan dan data telah tersimpan di database	
Normal Course		Aktor	Sistem
		1. Memilih menu data pelanggan	
			2. Menampilkan data pelanggan
		3. Memilih nama pelanggan yang akan diupdate pada tabel jasa	
			4. Menampilkan data pelanggan sesuai pencarian
		5. Mengubah form pelanggan yang akan di update	
		6. Klik tombol update	
			7. Validasi data
			8. Jika validasi berhasil maka tampil pesan data berhasil diupdate

Alternative Course	:	Aktor	Sistem
			8a. Jika validasi gagal maka tampil pesan data gagal diupdate
Exception	:	Aktor dapat membatalkan update data pelanggan dengan klik tombol exit	

Use Case Name	:	Delete Data Pelanggan	
Aktor	:	Admin	
Triger Event	:	Aktor memilih menu data pelanggan	
Pre-conditon	:	<ul style="list-style-type: none"> - Aktor telah melakukan login - Aktor telah membuka halaman data pelanggan 	
Post-condition	:	Aktor telah menghapus data pelanggan yang ada pada sistem transaksi laundry	
Normal Course		Aktor	Sistem
		5. Memilih menu data pelanggan	
			6. Menampilkan halaman data pelanggan
		7. Mengisi id jasa pada kolom id pelanggan	
		4. Klik tombol hapus	
			5. Validasi data
			6. Jika validasi berhasil maka tampil pesan data berhasil dihapus
Alternative Course	:	Aktor	Sistem
			6a. Jika validasi gagal maka tampil pesan data gagal dihapus
Exception	:	Aktor dapat membatalkan delete data pelanggan dengan klik tombol exit atau pilih menu keluar	

Use Case Name	:	Batalkan Data Pelanggan	
Aktor	:	Admin	
Triger Event	:	Aktor memilih menu data pelanggan	
Pre-conditon	:	<ul style="list-style-type: none"> - Aktor telah melakukan login - Aktor telah membuka halaman data pelanggan 	
Post-condition	:	Aktor telah membatalkan entri data pelanggan	
Normal Course		Aktor	Sistem

		1. Memilih menu data pelanggan	
			2. Menampilkan halaman data pelanggan
		3. Klik tombol batal	
			4. Menampilkan Form data pelanggan kosong pada tiap kolom
Alternative Course	:	Aktor	Sistem
Exception	:	Aktor dapat membatalkan data pelanggan dengan klik tombol exit atau menutup browser	

3.5. Latihan 1

Buat Diagram UC Data Cucion Masuk Use Case Deskripsi dibawah ini...

Use Case Name	:	Entry Data Cucion Masuk	
Triger Event	:	Aktor memilih menu cucian masuk	
Aktor	:	Admin	
Pre-conditon	:	<ul style="list-style-type: none"> - Aktor telah login - Aktor telah membuka halaman data cucian masuk 	
Post-condition	:	Aktor telah menambah data cucian masuk dan data telah tersimpan di database	
Normal Course		Aktor	Sistem
		1. Memilih menu data cucian masuk	
			2. Menampilkan form data cucian masuk
		3. Mengisi form cucian masuk	
		4. Memilih tombol simpan	
			5. Melakukan validasi data
			6. Jika data berhasil divalidasi maka tampil pesan data berhasil ditambahkan
		Aktor	Sistem
			8a. Jika data gagal divalidasi maka tampil pesan data gagal ditambahkan
Alternative Course	:	Aktor dapat membatalkan tambah data cucian masuk dengan mengklik tombol exit pilih menu keluar	

Exception	:		
------------------	---	--	--

Use Case Name	:	Update Data Cucian Masuk	
Aktor	:	Admin	
Triger Event	:	Aktor memilih menu data cucian masuk	
Pre-conditon	:	<ul style="list-style-type: none"> - Aktor telah login - Aktor telah membuka halaman data cucian masuk 	
Post-condition	:	Aktor telah menambah data cucian masuk dan data telah tersimpan di database	
Normal Course		Aktor	Sistem
		1. Memilih menu data cucian masuk	
			2. Menampilkan data cucian masuk
		3. Memilih nama cucian masuk yang akan diupdate pada tabel cucian masuk	
			4. Menampilkan data cucian masuk sesuai pencarian
		5. Mengubah form cucian masuk yang akan di update	
		6. Klik tombol update	
			7. Validasi data
			8. Jika validasi berhasil maka tampil pesan data berhasil diupdate
Alternative Course	:	Aktor	Sistem
			8a. Jika validasi gagal maka tampil pesan data gagal diupdate
Exception	:	Aktor dapat membatalkan update data cucian masuk dengan klik tombol exit atau pilih menu keluar	

Use Case Name	:	Delete Data Cucian Masuk	
Aktor	:	Admin	
Triger Event	:	Aktor memilih menu data cucian masuk	
Pre-conditon	:	<ul style="list-style-type: none"> - Aktor telah melakukan login - Aktor telah membuka halaman cucian masuk 	
Post-condition	:	Aktor telah menghapus data cucian masuk yang ada pada sistem transaksi laundry	
Normal Course		Aktor	Sistem

		1. Memilih menu data cucian masuk	
			2. Menampilkan halaman data cucian masuk
		3. Mengisi id cucian masuk pada kolom id cucian masuk	
		4. Klik tombol hapus	
			5. Validasi data
			6. Jika validasi berhasil maka tampil pesan data berhasil dihapus
Alternative Course	:	Aktor	Sistem
			6a. Jika validasi gagal maka tampil pesan data gagal dihapus
Exception	:	Aktor dapat membatalkan delete data cucian masuk dengan klik tombol exit atau pilih menu keluar	

Use Case Name	:	Batalan Data Cucian Masuk	
Aktor	:	Admin	
Triger Event	:	Aktor memilih menu data cucian masuk	
Pre-conditon	:	- Aktor telah melakukan login - Aktor telah membuka halaman data cucian masuk	
Post-condition	:	Aktor telah membatalkan entri data cucian masuk	
Normal Course		Aktor	Sistem
		1. Memilih menu data cucian masuk	
			2. Menampilkan halaman data cucian masuk
		3. Klik tombol batal	
			4. Menampilkan Form Data cucian masuk kosong pada tiap kolom
Alternative Course	:	Aktor	Sistem
Exception	:	Aktor dapat membatalkan data cucian masuk dengan klik tombol exit atau pilih menu keluar	

Use Case Name	:	Cek Point
Aktor	:	Admin
Triger Event	:	Aktor memilih menu data cucian masuk
Pre-conditon	:	- Aktor telah melakukan login

		- Aktor telah membuka halaman data cucian masuk
Post-condition	:	Aktor telah mengetahui poin yang di dapat saat proses entri data cucian masuk
Normal Course		Aktor
		1. Memilih menu data cucian masuk
		2. Menampilkan halaman data cucian masuk
		3. Klik tombol cek point
		4. Menampilkan data point pada Form Data cucian masuk
Alternative Course	:	Aktor
		Sistem
Exception	:	Aktor dapat membatalkan cek point data cucian masuk dengan klik tombol exit atau pilih menu keluar

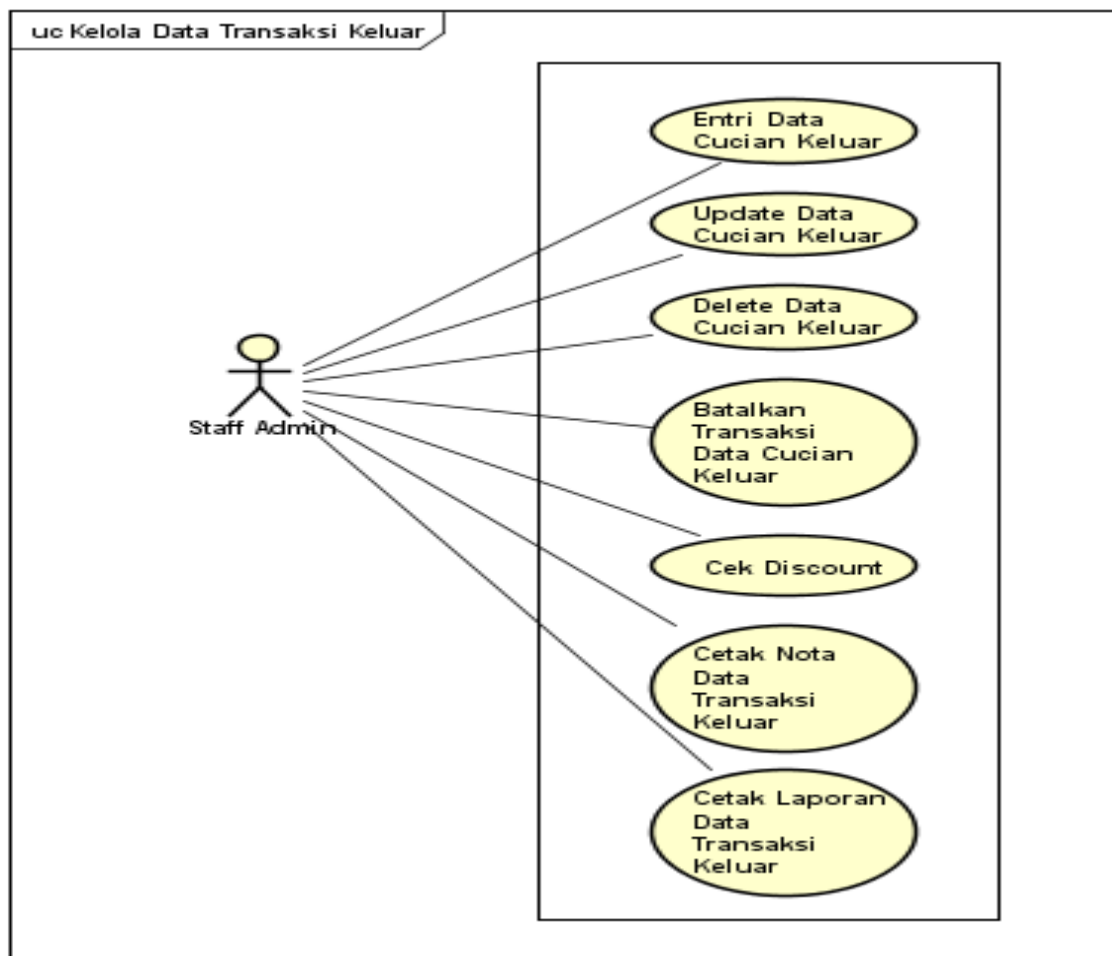
Use Case Name	:	Cetak Nota Cucian Masuk
Triger Event	:	Aktor memilih untuk mencetak nota cucian masuk
Aktor	:	Admin
Pre-conditon	:	- Aktor telah Login - Aktor telah membuka halaman data cucian masuk
Post-condition	:	Aktor dapat mencetak nota cucian masuk
Normal Course		Aktor
		1. Memilih menu data cucian masuk
		2. Menampilkan halaman data cucian masuk
		3. Memilih tombol cetak nota
		4. Menampilkan data transaksi berupa form cetakan
Alternative Course	:	Aktor
		Sistem
Exception	:	User dapat membatalkan cancel cetak nota cucian masuk dengan klik tombol batalkan atau pilih menu keluar

Use Case Name	:	Cetak Laporan Cucian Masuk
Triger Event	:	Aktor memilih untuk mencetak laporan cucian masuk
Aktor	:	Admin
Pre-conditon	:	- Aktor telah Login - Aktor telah membuka halaman cetak laporan cucian masuk

Post-condition	:	Aktor dapat mencetak laporan transaksi	
Normal Course		Aktor	Sistem
		1. Memilih menu cetak laporan	
			2. Menampilkan halaman cetak laporan
		3. Memilih tombol cetak	
			4. Menampilkan data transaksi berupa form cetakan
Alternative Course	:	Aktor	Sistem
Exception	:	User dapat membatalkan cancel cetak laporan dengan klik tombol batalkan atau pilih menu keluar	

3.6. Latihan 2

Buat Usecase Deskripsi Dari Diagram UC Kelola Data Transaksi Keluar.



Use Case Name	:	Entry Data Cucian Keluar	
Triger Event	:		
Aktor	:		
Pre-conditon	:	-	
Post-condition	:		
Normal Course		Aktor	Sistem
		Aktor	Sistem
Alternative Course	:		
Exception	:		

3.7. Latihan 3

KASUS

Pada perpustakaan Universitas Buana Perjuangan Karawang memberikan fasilitas peminjaman buku kepada mahasiswa, dosen, dan karyawan.. namun sebelum melakukan peminjaman buku terlebih dahulu harus menjadi anggota perpustakaan. Agar mempermudah petugas perpustakaan untuk mengelola data pustaka, data anggota perpustakaan maka dibuatkan aplikasi perpustakaan untuk mempermudah pengelolaan data-data perpustakaan.

Dimana seorang petugas bisa mengetahui jumlah buku yang sedang dipinjam dan daftar buku apa saja yang sedang dipinjam oleh anggota. Setiap anggota hanya bisa meminjam buku maximal 3, dan hanya diberi waktu pinjaman selama 1 minggu, apabila anggota melebihi dari jangka waktu peminjaman maka anggota harus membayar denda.

Setiap Anggota perpustakaan mempunyai kartu anggota setelah mendaftarkan diri menjadi anggota perpustakaan. Lalu petugas perpustakaan yang mengelola data anggota tersebut.

Aktor yang dijalankan :

1. **Petugas Perpustakaan** : orang yang bertugas untuk mengelola data pustaka, data anggota perpustakaan. Dimana petugas bisa menambahkan daftar pustaka (buku), mengedit pustaka, dan menghapus data pustaka, serta petugas bisa menambahkan data anggota perpustakaan.
2. **Anggota Perpustakaan** : hanya mendaftarkan diri sebagai anggota perpustakaan.

MODUL 4

CLASS DIAGRAM

4.1. Class Sistem Loundry

1. Class Login

Nama Field	Tipe Data	Panjang Data	keterangan
username	Varchar	50	Nama user
password	Int	20`	Kata Sandi

2. Class Jasa

Nama Field	Tipe Data	Panjang Data	keterangan
kode_jasa	Int	20	Primari Key
nama_jasa	Varchar	100	Nama Jasa
satuan	Varchar	100	Satuan Jasa
harga_jasa	Int	50	Harga jasa

3. Class Pelanggan

Nama Field	Tipe Data	Panjang Data	keterangan
kode_pelanggan	Int	20	Primari Key
nama_pelanggan	Varchar	100	Nama Pelanggan
alamat	Varchar	100	Alamat
no_telp	Int	50	Nomor telpn

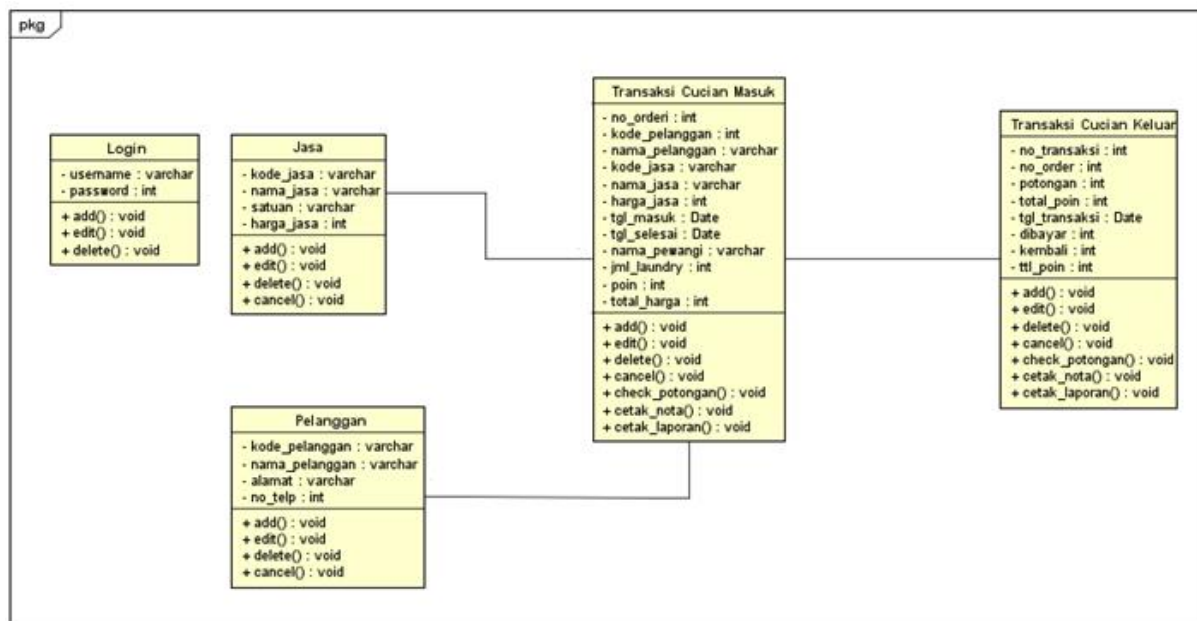
4. Class transaksi_cucian_masuk

Nama Field	Tipe Data	Panjang Data	keterangan
no_order	int	20	Primari Key
kode_pelanggan	Int	20	Foreigen key
nama_pelanggan	Varchar	100	Nama Pelanggan
kode_jasa	Int	20	Foreigen key
nama_jasa	Varchar	100	Nama Jasa
harga_jasa	Int	50	Harga jasa
tgl_masuk	Date	-	Tanggal
tgl_keluar	Date	-	Tanggal
nama_pewangi	Varchar	100	Pengharum Pakaian
jml_laundry	Int	50	Jumlah Laundry
poin	Int	20	Point Loundry
total_harga	Int	100	Total Harga Loundry

5. Class transaksi_cucian_keluar

Nama Field	Tipe Data	Panjang Data	keterangan
no_transaksi	Int	20	Primari Key
no_order	Int	20	Foreigen key
potongan	Int	50	Potongan harga
total_poin	Int	50	Jumlah Point
tgl_transaksi	Date	-	Tanggal Pengambilan
dibayar	Int	50	Pembayaran
kembali	Int	50	Kembalian
tgl_poin	Int	50	Jumlah point yang sudah digunakan

4.2. Class Diagram Sistem Laundry



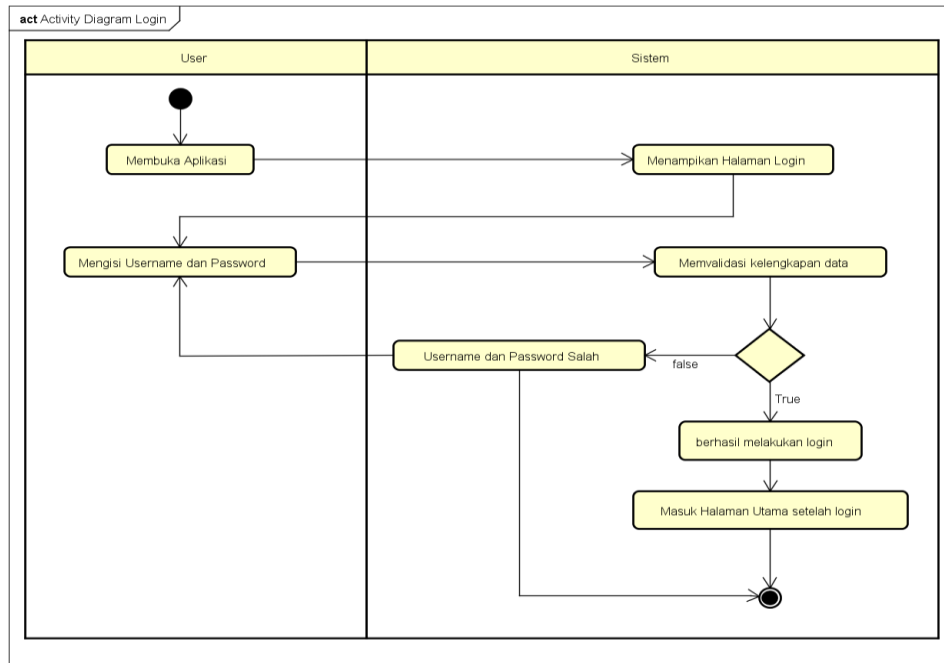
4.3. Latihan 4

1. Buat tabel class dari sistem perpustakaan
2. Buat class diagram dari sistem perpustakaan

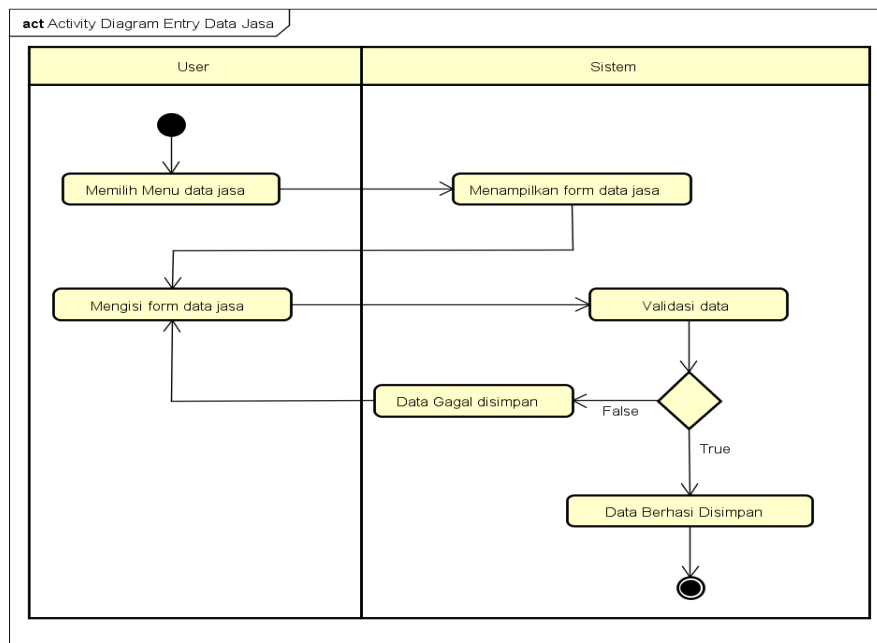
MODUL 5

ACTIVITY DIAGRAM

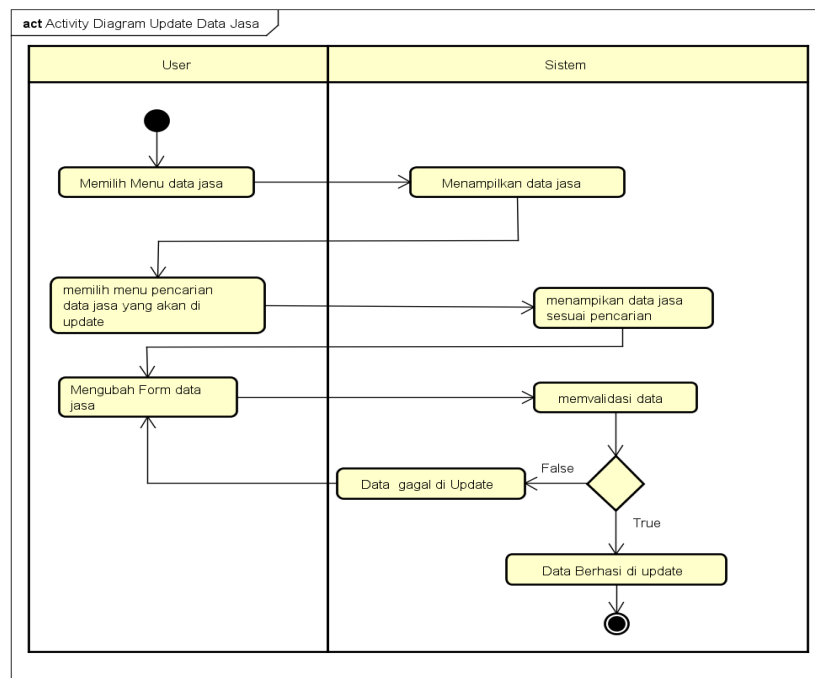
5.1. Activity Diagram Login



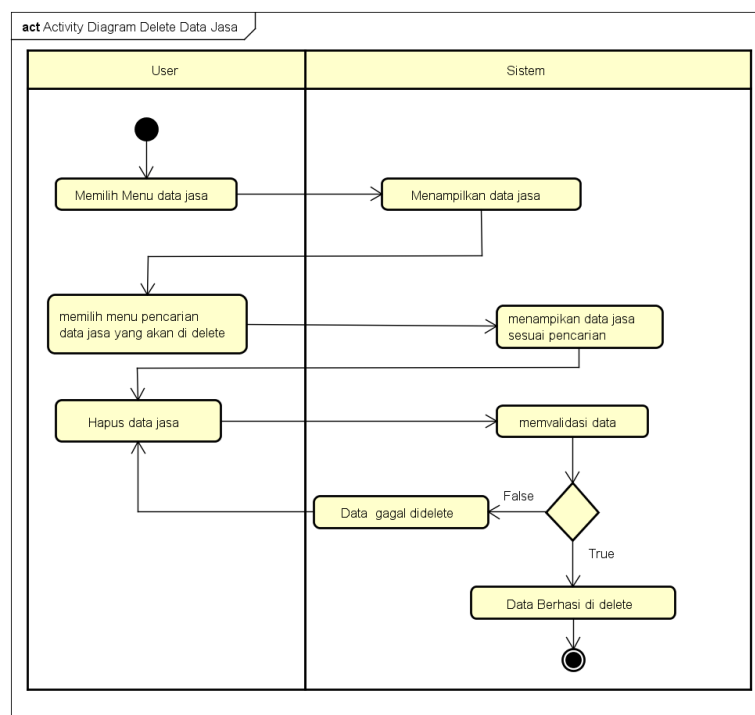
5.2. Activity Diagram Entry Data Jasa



5.3. Activity Diagram Update Data Jasa



5.4. Activity Diagram Delete Data Jasa



5.5. Latihan 5

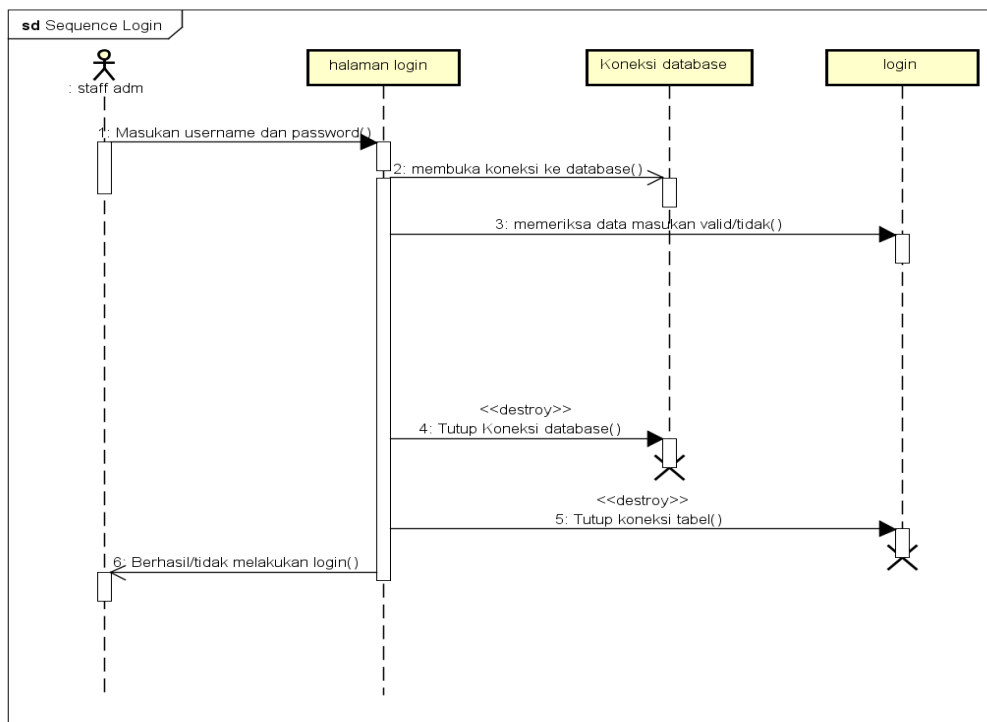
Buat activity Diagram dari Package diagram Kelola data Pelanggan, Kelola Data Transaksi Keluar, Kelola DataTransaksi Masuk, sesuai dengan UC. Deskripsi yang dibuat sebelumnya.

MODUL 6

SEQUENCE DIAGRAM

Diagram sekuen adalah menggambarkan kelakuan objek pada usecase dengan mendeskripsikan waktu hidup objek dan pesan yang dikirimkan atau diterima antar objek. Oleh karena itu untuk menggambarkan diagram sekuen maka harus diketahui objek-objek yang terlibat dalam sebuah usecase diagram beserta metode-metode yang dimiliki kelas yang diinstansiasi menjadi objek itu. membuat diagram skuen juga dibutuhkan untuk melihat skenario/deskripsi usecase pada usecase diagram yang sudah dibuat.

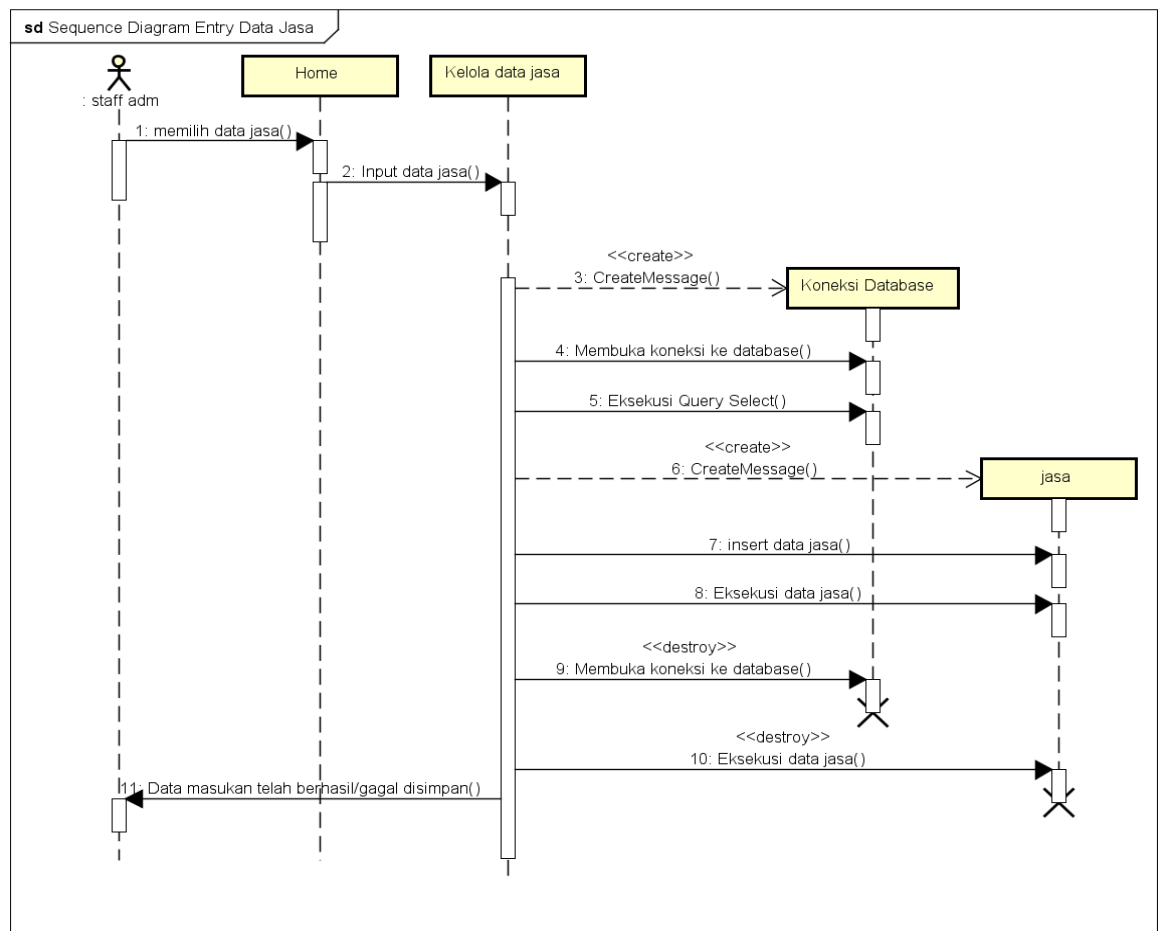
6.1. Sequence Diagram Login



Ket :

Admin akan memasukan username dan password saat melakukan login untuk masuk kehalaman utama, kemudian system akan melakukan koneksi database ke tabel login untuk memeriksa data masukan, apabila data masukan sesuai/berhasil maka akan masuk kehalaman utama/home jika tidak berhasil maka lakukan kembali proses login.

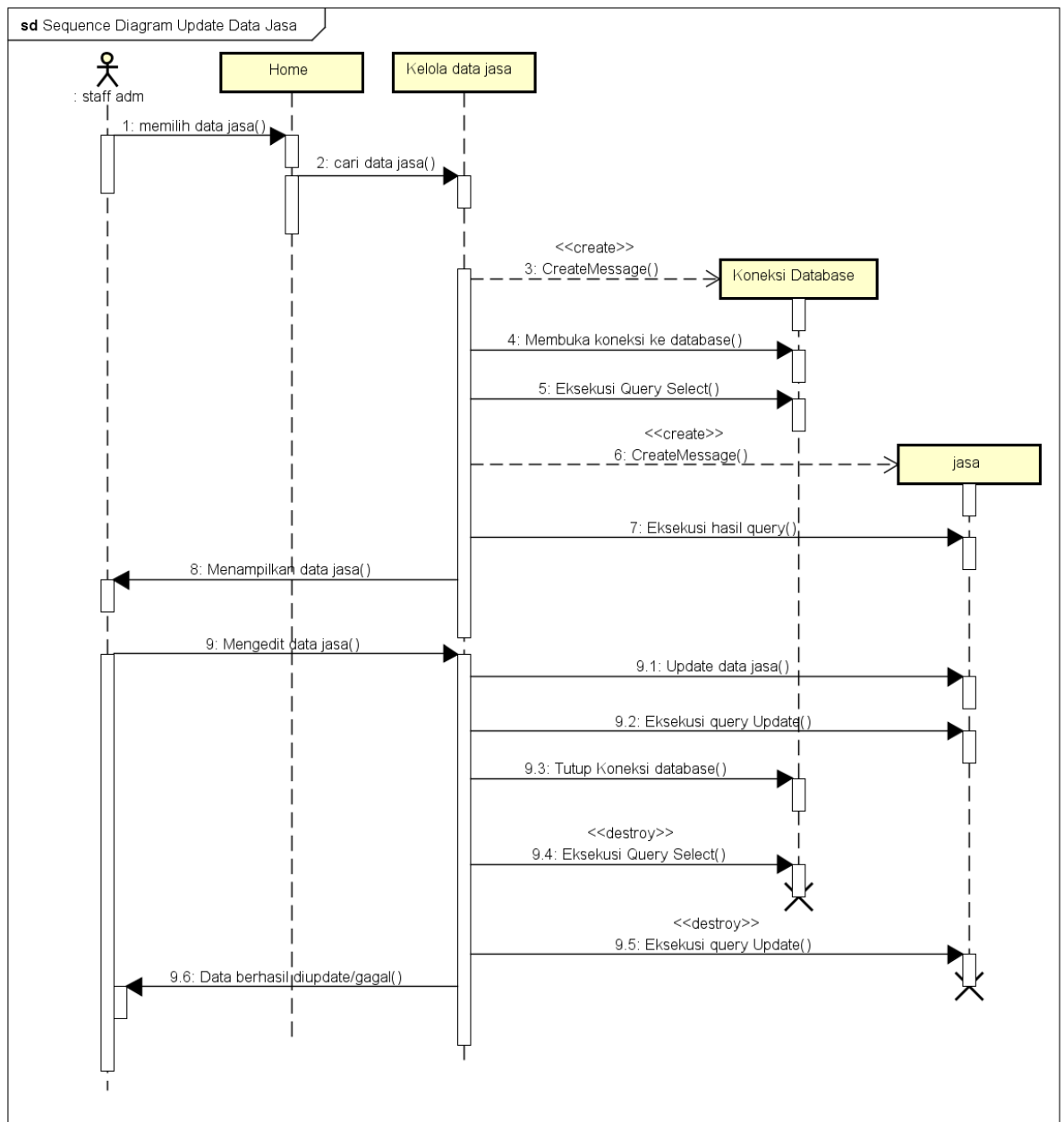
6.2. Squence Diagram Entry Data Jasa



Ket :

Staff admin memasukan data jasa ke dalam system, kemudian memilih menu tambah data jasa, selanjutnya data akan tersimpan dalam database tabel jasa. Apabila sudah selesai memasukan data barang maka terdapat pesan data barang tersimpan/gagal tersimpan.

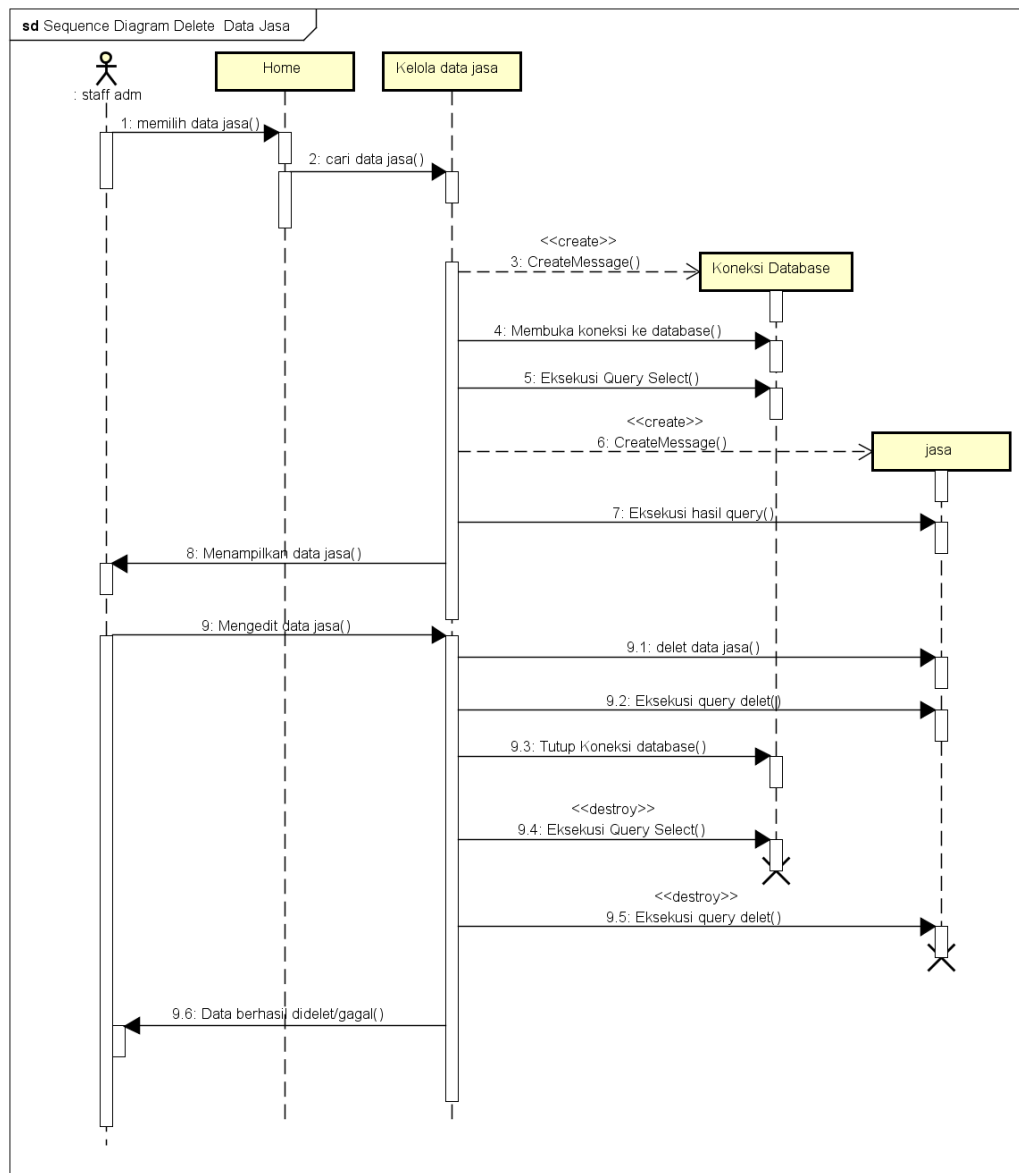
6.3. Sequence Diagram Update Data Jasa



Ket :

Staff admin melakukan edit data jasa didalam system, kemudian memilih menu cari data jasa, selanjutnya menampilkan data jasa yang dicari dan memilih data jasa yang akan diedit. Apabila sudah selesai mengedit data jasa maka terdapat pesan data jasa berhasil/tidak terupdate.

6.4. Squence Diagram Delete Data Jasa



Ket :

Staff admin melakukan delete data jasa didalam system, kemudian memilih menu cari data jasa, selanjutnya menampilkan data jasa yang dicari dan memilih data jasa yang akan di delete. Apabila sudah selesai delete data jasa maka terdapat pesan data jasa berhasil/tidak terdelete.

6.5. Latihan 6

Buat Squence Diagram dari Package diagram Kelola data Pelanggan, Kelola Data Transaksi Keluar, Kelola Data Transaksi Masuk.