

## Desain Sistem : Use Case Diagram

**Lila Setiyani**

Program Studi Sistem Informasi, STMIK Rosma

Karawang, Indonesia

[lila.setiyani@dosen.rosma.ac.id](mailto:lila.setiyani@dosen.rosma.ac.id)

### ABSTRAK

Pemodelan merupakan bagian dari tahapan dari model proses pengembangan perangkat lunak. Use Case merupakan salah satu *tools* yang digunakan untuk membuat pemodelan interaksi user dengan sistem. Penelitian ini bertujuan untuk membuat pemodelan interaksi user dengan sistem pada sistem informasi penjualan dengan menggunakan use case. Metode untuk pemodelan use case menggunakan UML (Unified Modelling Language) yang merupakan standar pemodelan secara visual, perancangan dan pendokumentasian sebuah sistem yang menghasilkan sebuah *blueprint* dari aplikasi. Hasil dari penelitian ini adalah berupa *blueprint* dari aplikasi penjualan yang digambarkan dalam bentuk use case. Hasil penelitian ini dapat digunakan untuk menyusun pemodelan lanjutan yaitu pemodelan *activity diagram* serta dapat mendukung proses penyusunan *mockup* dari aplikasi penjualan.

**Kata kunci:** Pemodelan, UML, Unified Modelling Language, Use Case

### ABSTRACT

*Modeling is part of the stages of the software development process model. Use Case is one of the tools used to model user interaction with the system. This study aims to create a modeling of user interaction with the system on the sales information system by using a use case. The method for use case modeling uses UML (Unified Modeling Language) which is a standard for visual modeling, design and documentation of a system that produces a blueprint of the application. The results of this study are in the form of a blueprint of a sales application which is described in the form of a use case. The results of this study can be used to develop advanced modeling, namely modeling activity diagrams and can support the process of preparing mockups of sales applications.*

**Key words:** Modeling, UML, Unified Modeling Language, Use Case

## Pendahuluan

Model pengembangan perangkat lunak telah berkembang dengan pesat dari model *waterfall*(Pressman, 2010) sampai dengan *agile*(McDermid & Rook, 2019)(Rodríguez et al., 2019) yang telah memperkenalkan beberapa model yang lebih adaptif terhadap proyek pengembangan perangkat lunak seperti *scrum*(Egy Muhammad Rianof et al., 2020). Dalam setiap model pengembangan perangkat lunak, terdapat tahapan – tahapan yang memungkinkan terjadi pengulangan. Dalam konsep *software development life cycle*(SDLC)(Okesola et al., 2020) dikenal beberapa tahapan diantaranya *planning and requirement analysis, defining requirement, designing, development, testing, deployment and maintenance*. Setiap tahapan pada SDLC merupakan kerangka yang mendukung efektivitas dan efisiensi dalam proses pengembangan perangkat lunak. Dalam tahapan *designing*, dilakukan pemodelan interaksi antara *user* dengan sistem, sehingga dapat dipahami skenario dari sistem tersebut, yang disebut dengan *use case*. *Use case* adalah seperangkat skenario yang diikat bersama oleh *user* untuk mencapai tujuan(Fowler & Scott, 1993). *Use case* mendukung pengembang perangkat lunak memahami interaksi. Aktor dalam sebuah sistem melakukan banyak *use case*, dan *use case* mungkin bisa memiliki banyak aktor. Antara *use case* dengan aktor atau dengan *use case* terdapat beberapa *links* hubungan *include, extend, generalization* dan lain – lain.

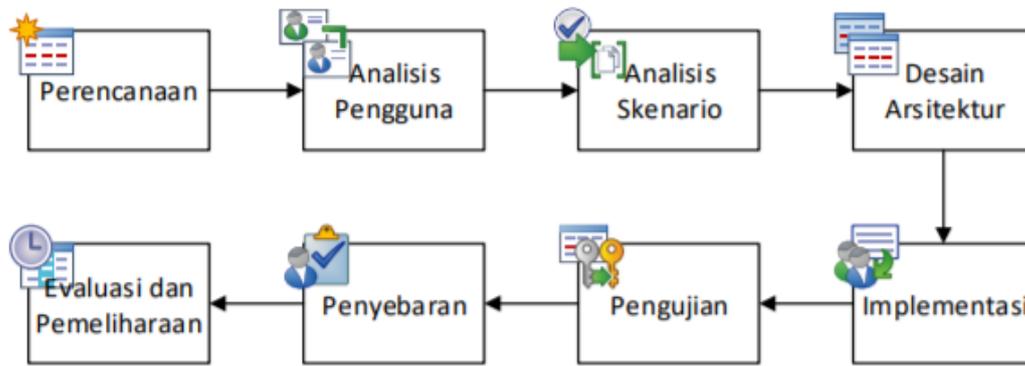
Pemodelan interaksi ini sangat penting dalam proses pengembangan perangkat lunak. Pentingnya aktivitas pemodelan ini, mendorong peneliti untuk dapat memberikan ulasan dalam tahapan pemodelan menggunakan pemodelan berbasis *Object Oriented Programming*(OOP). Ulasan yang tepat dalam menggambarkan proses pemodelan interaksi akan memberikan gambaran yang jelas kepada pengembang perangkat lunak dalam mengimplementasikan proyek pengembangan perangkat lunak. Oleh sebab itu, penelitian ini bertujuan untuk membangun sebuah pemodelan sistem yang berbasis OOP) dengan menggunakan *Unified Modelling Language*(UML). Beberapa peneliti telah melakukan pemodelan menggunakan *use case* diantaranya (Kurniawan, 2018), (Iskandar, 2012), (Wira et al., 2019). Dalam penelitian ini, peneliti akan melakukan pemodelan terhadap sebuah studi kasus yaitu transaksi penjualan. Pemodelan *use case* akan dilakukan, dengan menggambarkan interaksi *user* dengan sistem pada transaksi penjualan. Hasil penelitian ini dapat digunakan sebagai gambaran untuk menyusun pemodelan dari sebuah sistem yang dapat dikembangkan dengan pemodelan *activity diagram, class diagram* dan *mockup*, sehingga proses pengembangan perangkat lunak dapat lebih efektif dan efisien, karena kebutuhan pengguna dapat diterjemahkan dengan tepat(Migunani, 2007). Sunardi dan Fadli dalam penelitiannya melakukan implementasi model proses *agile scrum*, berdasarkan penelitiannya mereka menjelaskan bahwa dalam model proses tersebut, penting untuk

melakukan manajemen waktu dalam tim pengembang agar pekerjaan dapat terjadwal tepat waktu (Sunardi & Fadli, 2018).

### **1. Pengembangan perangkat lunak dan model proses pengembangan perangkat lunak**

Pengembangan perangkat lunak merupakan sebuah proses membangun perangkat lunak baru untuk mengganti perangkat lunak yang lama atau memperbaiki perangkat lunak yang ada (Budi et al., 2016). Keberhasilan dalam proses pengembangan perangkat lunak bergantung dari manajemen proyek pengembangan perangkat lunak. Agar pengembangan perangkat lunak dapat sukses, maka diperlukan model proses pengembangan perangkat lunak yang tepat, disesuaikan dengan kebutuhan pengguna.

Dalam proses pengembangan perangkat lunak dikenal beberapa model diantaranya *System Development Life Cycle (SDLC)*, *Linear Sequential Model* atau *Waterfall*, *Parallel Model*, *Iterative Model*, *Prototyping Model*, *RAD (Rapid Application Development) Model*, *Spiral Model*, *V-Shaped Model* dan *Agile Development*. Menurut (Budi et al., 2016), pertimbangan pemilihan metodologi yang didasarkan pada faktor – faktor kriteria penilaian yang terdiri atas kejelasan persyaratan pengguna, keakraban dengan teknologi, kompleksitas sistem, keandalan sistem, dan *time management*. Dalam penelitian (Raharja & Sn, 2012) membandingkan model pengembangan *spiral* dan *cleanroom* yang merupakan model pengembangan *evolusioner*, Raharja menjelaskan bahwa model tersebut sangat diperhitungkan dan banyak digunakan pada proyek skala besar dan kritis. Migunani membandingkan model proses *waterfall* dan *spiral*, yang mendapati bahwa model *waterfall* menggambarkan sebuah model proses statis dengan tahapan yang berlapis menggunakan sebuah *milestone* sebagai transisi pada setiap tahap perancangan, sedangkan model *spiral* menerapkan model proses secara sirkular tanpa adanya *milestone*, namun menurutnya model *spiral* lebih memiliki kelebihan yaitu terkait kebutuhan pengembangan secara berkelanjutan dan adanya keterlibatan pemakai dalam pembangunan perangkat lunak, sehingga perangkat lunak akan berkembang dengan versi dan fitur – fitur yang lebih baik. Selain itu dalam pengembangan perangkat lunak berbasis mobile, Firamon dalam penelitiannya membahas tiga model yaitu *wireless development*, *mobile application development lifecycle model (MADLC)* dan *mobile development (Mobile-D)* (Syakti, 2019). Menurut Ependi et al. (2017) dalam (Syakti, 2019) model proses pengembangan perangkat lunak *wireless development* dapat digambarkan sebagai berikut



Gambar 1. Model proses pengembangan perangkat lunak *wireless development* Ependi et.al(2017)

Menurut Vithani dan Kumar (2014) dalam (Syakti, 2019) menggambarkan model pengembangan MADLC sebagai berikut :



Gambar 2. Model pengembangan *mobile application development lifecycle model*(MADLC) Vithani dan Kumar (2014)

Sedangkan model pengembangan dengan menggunakan model *mobile development (Mobile – D)* menurut Abrahamsson et. al (2004) dalam (Syakti, 2019) digambarkan sebagai berikut :



Gambar 3. Model *mobile development (Mobile – D)* Abrahamsson et. al (2004)

Melihat pada model proses pengembangan perangkat lunak yang telah di ulas oleh para peneliti sebelumnya, dapat disimpulkan bahwa setiap model memiliki aktivitas bertujuan untuk memberikan kemudahan dalam implementasi pengembangan perangkat lunak. Oleh sebab itu, penting bagi para pengembang perangkat lunak untuk dapat memilih model yang tepat dalam proses pengembangan perangkat lunak, sehingga prosesnya dapat menjadi lebih efektif dan efisien.

## 2. Usecase

Dalam pemograman berbasis OOP, untuk mendeskripsikan sistem dapat menggunakan diagram UML. Diagram tersebut terdiri dari 13 jenis diagram yaitu *activity*, *class*,

*communication, component, composite structure, deployment, interaction overview, object, package, squence, state machine, timing dan use case. Use case* merupakan deskripsi fungsi dari sebuah sistem dari perspektif atau sudut pandang para pengguna sistem. Use case mendefinisikan apa yang akan diproses oleh sistem dan komponen – komponennya. Use case bekerja dengan menggunakan *scenario* yang merupakan deskripsi dari urutan atau langkah – langkah yang menjelaskan apa yang dilakukan oleh *user* terhadap sistem maupun sebaliknya. Use case mengidentifikasi fungsionalitas yang dipunya sistem, interaksi *user* dengan sistem dan keterhubungan antara *user* dengan fungsionalitas sistem (Arifin & Hs, 2017). DSRM terdiri dari enam tahapan yaitu *problem identification and motivitation, objective of the solution, design and development, demonstration, evaluation dan communication*.

### Materi dan Metode

Penelitian ini termasuk pada penelitian *Design Science Research Methodology* (DSRM) yang merupakan metodologi yang berorientasi pada desain sistem informasi. DSRM memberikan kerangka prosedur yang digunakan untuk memberikan kemudahan penelitian dibidang teknologi informasi, dimana membantu dalam proses pemahaman serta mengulas untuk mengenali dan mengevaluasi hasil penelitian (Nabyla & Sigitta, 2019). Pada penelitian ini DSRM diadopsi, sehingga prosedur pada penelitian ini adalah sebagai berikut :



Gambar 4. Prosedur penelitian

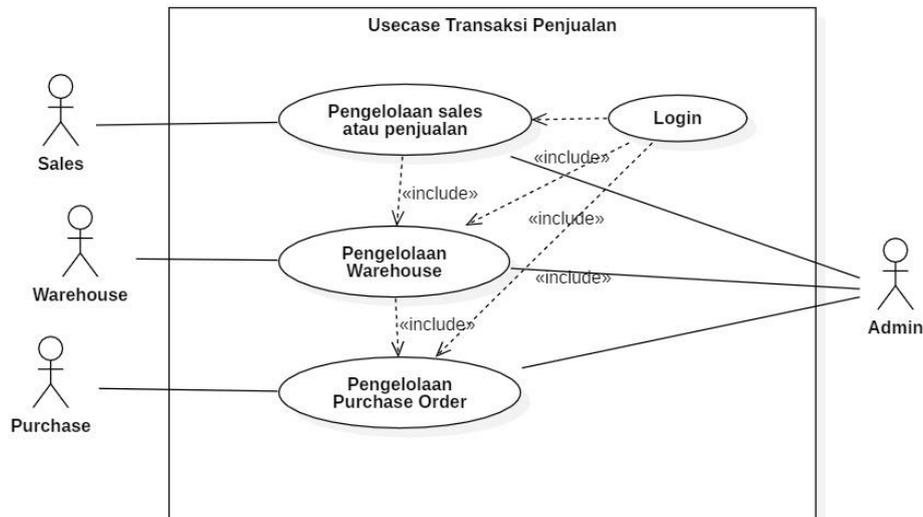
### Hasil dan Pembahasan

Berdasarkan hasil identifikasi kebutuhan pengguna ditetapkan bahwa terdapat empat aktor yang mempunyai akses dalam menggunakan sistem dalam usecase diagram yang berinteraksi dengan sistem, aktor-aktor tersebut dapat dilihat pada tabel 1 dibawah ini.

Tabel 1. Identifikasi Aktor

Aktor	Deskripsi
Sales	Orang yang bertugas membuat penawaran barang kepada customer, mengelola PO dari Customer, mengelola invoice dan faktur
Warehouse	Orang yang mengelola barang masuk, barang keluar, surat jalan dan retur dari customer dan supplier
Purchase	Orang yang mengelola Purchase Order
Admin	Super admin yang dapat mengakses seluruh modul sistem

Pemodelan interaksi user dengan sistem terdapat empat usecase, yaitu login, pengelolaan sales atau penjualan, pengelolaan warehouse dan pengelolaan purchase order. Usecase transaksi penjualan dapat dilihat pada gambar 5 dibawah ini.



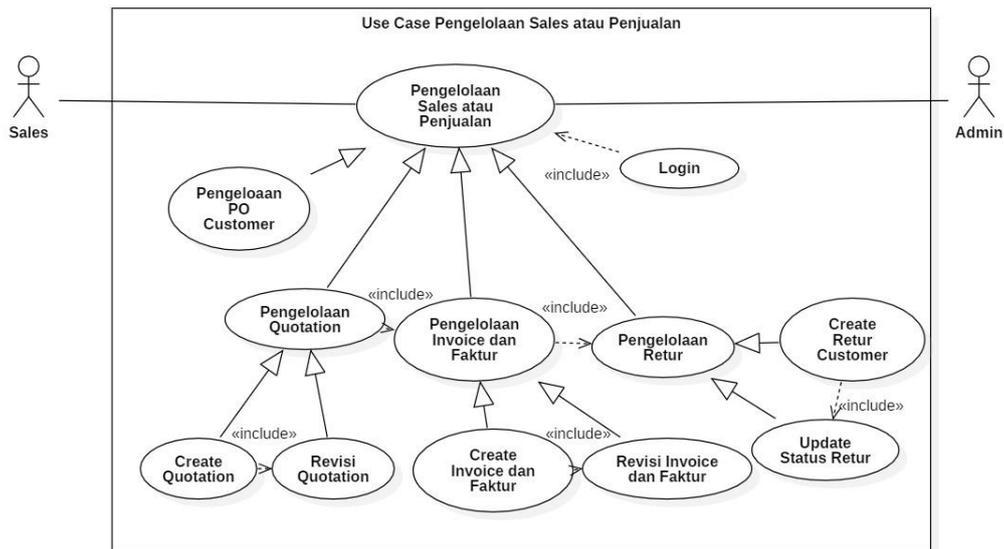
Gambar 5. Usecase Transaksi Penjualan

Tabel 2. Definisi Use Case Transaksi Penjualan

Use Case	Deskripsi
<b>Login</b>	Merupakan proses untuk melakukan login staf
<b>Pengelolaan sales atau penjualan</b>	Merupakan proses pengelolaan penjualan yang meliputi membuat quotation, merevisi quotation, membuat invoice dan faktur pajak, mengelola PO dari customer, dan pengelolaan retur barang dari customer
<b>Pengelolaan warehouse</b>	Merupakan proses pengelolaan transaksi yang ada di warehouse meliputi, pencatatan barang masuk, pencatatan barang keluar, pengelolaan surat jalan
<b>Pengelolaan purchase order</b>	Merupakan proses yang mengelola purchase barang ke supplier yang meliputi pengelolaan PO dari supplier, pengelolaan penerimaan surat jalan, pengelolaan pembayaran PO

### 1. Usecase Pengelolaan Sales dan Penjualan

Pemodelan usecase pengelolaan sales dan penjual dapat dilihat pada gambar dibawah ini.



Gambar 6. Usecase Pengelolaan Sales atau Penjualan

Berdasarkan gambar 6 diatas, definisi dan skenario usecase pengelolaan barang dapat dilihat pada tabel 3 dan tabel 4 dibawah ini.

Tabel 3. Definisi Use Case Pengelolaan Sales atau Penjualan

Use Case	Deskripsi
<b>Login</b>	Merupakan proses untuk melakukan login staf
<b>Pengelolaan PO Customer</b>	Merupakan proses yang mengelola penerimaan PO dari customer
<b>Pengelolaan quotation</b>	Merupakan proses yang mengelola pembuatan quotation dan revisi quotation
<b>Pengelolaan invoice dan faktur</b>	Merupakan proses yang mengelola pembuatan invoice dan faktur serta revisi nya
<b>Pengelolaan retur</b>	Merupana proses yang mengelola pembuatan retur customer serta update status retur

Tabel 4. Skenario Usecase Pengelolaan Sales atau Penjualan

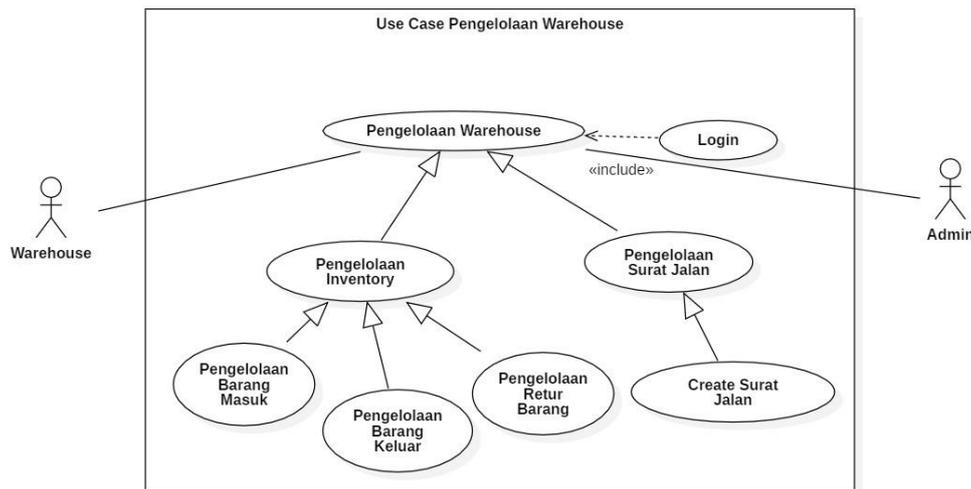
Aksi Aktor	Reaksi Sistem
<b>Skenario Use Case Create Quotation</b>	
<b>Skenario Normal</b>	
	1. Memeriksa status login
2. Memasukkan data penawaran yang berupa data customer dan data barangnya	3. Memeriksa valid tidaknya data masukkan
	4. Menyimpan data penawaran ke basis data
	5. Menampilkan pesan sukses disimpan
<b>Skenario Alternatif</b>	
6. Memasukkan data penawaran sesuai dengan kolom yang ada	7. Memeriksa valid tidaknya data masukksn
	8. Mengeluarkan pesan bahwa data yang dimasukkan tidak valid

Aksi Aktor	Reaksi Sistem
9. Memperbaiki data masukkan yang tidak valid	10. Memeriksa valid tidaknya data masukkan
	11. Menyimpan data ke basis data
	12. Menampilkan pesan sukses disimpan
<b>Skenario Use Case Revisi Quotation</b>	
<b>Skenario Normal</b>	
	1. Memeriksa status login
2. Memilih no quotation	3. Menampilkan no quotation yang dipilih
4. Mervisi data	5. Menyimpan data penawaran ke basis data
	6. Menampilkan pesan sukses disimpan
<b>Skenario Alternatif</b>	
7. Memilih no quotation	8. Menampikan no quotation
9. Mervisi data	10. Memeriksa valid tidaknya data masuksn
	11. Mengeliarkan pesan bahwa data yang dimasukkan tidak valid
12. Memperbaiki data masukkan yang tidak valid	13. Memeriksa valid tidaknya data masukkan
	14. Menyimpan data ke basis data
	15. Menampilkan pesan sukses disimpan
<b>Skenario Use Case Create Retur Customer</b>	
<b>Skenario Normal</b>	
	1. Memeriksa status login
2. Memasukkan data retur yang berupa data customer dan data barangnya	3. Memeriksa valid tidaknya data masukkan
	4. Menyimpan data tagihan ke basis data
	5. Menampilkan pesan sukses disimpan
<b>Skenario Alternatif</b>	
6. Memasukkan data tagihan sesuai dengan kolom yang ada	7. Memeriksa valid tidaknya data masuksn
	8. Mengeliarkan pesan bahwa data yang dimasukkan tidak valid
9. Memperbaiki data masukkan yang tidak valid	10. Memeriksa valid tidaknya data masukkan
	11. Menyimpan data ke basis data
	12. Menampilkan pesan sukses disimpan
<b>Skenario Use Case Update Retur Customer</b>	
<b>Skenario Normal</b>	
	1. Memeriksa status login
2. Memilih no retur	3. Menampilkan no retur yang dipilih
4. Update data	5. Menyimpan data rertur ke basis data
	6. Menampilkan pesan sukses disimpan
<b>Skenario Alternatif</b>	
7. Memilih no retur	8. Menampikan no retur
9. Update data	10. Memeriksa valid tidaknya data masukkan
	11. Mengeliarkan pesan bahwa data yang dimasukkan tidak valid
12. Memperbaiki data masukkan yang tidak valid	13. Memeriksa valid tidaknya data masukkan
	14. Menyimpan data ke basis data

Aksi Aktor	Reaksi Sistem
	15. Menampilkan pesan sukses disimpan
<b>Skenario Use Case Create Invoice dan Faktur</b>	
<b>Skenario Normal</b>	
	1. Memeriksa status login
2. Memasukkan data tagihan yang berupa data customer dan data barangnya	3. Memeriksa valid tidaknya data masukkan
	4. Menyimpan data tagihan ke basis data
	5. Menampilkan pesan sukses disimpan
<b>Skenario Alternatif</b>	
6. Memasukkan data tagihan sesuai dengan kolom yang ada	7. Memeriksa valid tidaknya data masukksn
	8. Mengeliarkan pesan bahwa data yang dimasukkan tidak valid
9. Memperbaiki data masukkan yang tida valid	10. Memeriksa valid tidaknya data masukkan
	11. Menyimpan data ke basis data
	12. Menampilkan pesan sukses disimpan
<b>Skenario Use Case Revisi Invoice dan Faktur</b>	
<b>Skenario Normal</b>	
	1. Memeriksa status login
2. Memilih no Invoice	3. Menampilkan no invoice yang dipilih
4. Mervisi data	5. Menyimpan data invoice ke basis data
	6. Menampilkan pesan sukses disimpan
<b>Skenario Alternatif</b>	
7. Memilih no invoice	8. Menampikan no invoice
9. Mervisi data	10. Memeriksa valid tidaknya data masukksn
	11. Mengeliarkan pesan bahwa data yang dimasukkan tidak valid
12. Memperbaiki data masukkan yang tida valid	13. Memeriksa valid tidaknya data masukkan
	14. Menyimpan data ke basis data
	15. Menampilkan pesan sukses disimpan
<b>Skenario Use Case Penerimaan PO Customer</b>	
<b>Skenario Normal</b>	
	1. Memeriksa status login
2. Memasukkan data PO yang berupa data customer dan data barangnya	3. Memeriksa valid tidaknya data masukkan
	4. Menyimpan data PO ke basis data
	5. Menampilkan pesan sukses disimpan
<b>Skenario Alternatif</b>	
6. Memasukkan data PO sesuai dengan kolom yang ada	7. Memeriksa valid tidaknya data masukkan
	8. Mengeliarkan pesan bahwa data yang dimasukkan tidak valid
9. Memperbaiki data masukkan yang tida valid	10. Memeriksa valid tidaknya data masukkan
	11. Menyimpan data ke basis data
	12. Menampilkan pesan sukses disimpan

## 2. Usecase Pengelolaan Warehouse

Pemodelan usecase pengelolaan warehouse dapat dilihat pada gambar 7 dibawah ini.



Gambar 7. Usecase Pengelolaan Warehouse

Berdasarkan gambar 7 diatas, definisi dan skenario usecase pengelolaan barang dapat dilihat pada tabel 5 dan tabel 6 dibawah ini.

Tabel 5. Definisi Use Case Pengelolaan Warehouse

Use Case	Deskripsi
<b>Login</b>	Merupakan proses untuk melakukan login staf
<b>Pengelolaan Inventory</b>	Merupakan proses yang mengelola barang masuk, barang keluar dan retur barang
<b>Pengelolaan Surat Jalan</b>	Merupakan proses yang mengelola pembuata surat jalan

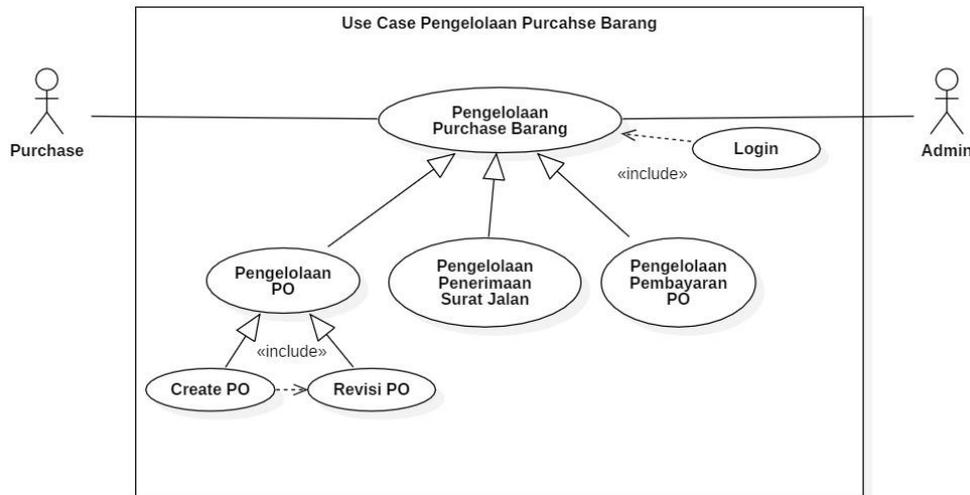
Tabel 6. Skenario Usecase Pengelolaan Warehouse

Aksi Aktor	Reaksi Sistem
<b>Skenario Use Case Pengelolaan Barang Masuk</b>	
<b>Skenario Normal</b>	
	1. Memeriksa status login
2. Memasukkan data barang masuk	3. Memeriksa valid tidaknya data masukkan
	4. Menyimpan data tagihan ke basis data
	5. Menampilkan pesan sukses disimpan
<b>Skenario Alternatif</b>	
6. Memasukkan data barang masuk sesuai dengan kolom yang ada	7. Memeriksa valid tidaknya data masukkan
	8. Mengeliarkan pesan bahwa data yang dimasukkan tidak valid
9. Memperbaiki data masukkan yang tidak valid	10. Memeriksa valid tidaknya data masukkan
	11. Menyimpan data ke basis data
	12. Menampilkan pesan sukses disimpan
<b>Skenario Use Case Pengelolaan Barang Keluar</b>	
<b>Skenario Normal</b>	
	1. Memeriksa status login

Aksi Aktor	Reaksi Sistem
2. Memasukkan data barang keluar	3. Memeriksa valid tidaknya data masukkan
	4. Menyimpan data barang ke basis data
	5. Menampilkan pesan sukses disimpan
<b>Skenario Alternatif</b>	
6. Memasukkan data barang keluar sesuai dengan kolom yang ada	7. Memeriksa valid tidaknya data masukkan
	8. Mengeluarkan pesan bahwa data yang dimasukkan tidak valid
9. Memperbaiki data masukkan yang tidak valid	10. Memeriksa valid tidaknya data masukkan
	11. Menyimpan data ke basis data
	12. Menampilkan pesan sukses disimpan
<b>Skenario Use Case Pengelolaan Retur Barang</b>	
<b>Skenario Normal</b>	
	1. Memeriksa status login
2. Memasukkan data retur barang	3. Memeriksa valid tidaknya data masukkan
	4. Menyimpan data retur ke basis data
	5. Menampilkan pesan sukses disimpan
<b>Skenario Alternatif</b>	
6. Memasukkan data retur sesuai dengan kolom yang ada	7. Memeriksa valid tidaknya data masukksn
	8. Mengeliarkan pesan bahwa data yang dimasukkan tidak valid
9. Memperbaiki data masukkan yang tidak valid	10. Memeriksa valid tidaknya data masukkan
	11. Menyimpan data ke basis data
	12. Menampilkan pesan sukses disimpan
<b>Skenario Use Case Create Surat Jalan</b>	
<b>Skenario Normal</b>	
	1. Memeriksa status login
2. Memasukkan data surat jalan	3. Memeriksa valid tidaknya data masukkan
	4. Menyimpan data retur ke basis data
	5. Menampilkan pesan sukses disimpan
<b>Skenario Alternatif</b>	
6. Memasukkan data surat jalan sesuai dengan kolom yang ada	7. Memeriksa valid tidaknya data masukksn
	8. Mengeluarkan pesan bahwa data yang dimasukkan tidak valid
9. Memperbaiki data masukkan yang tidak valid	10. Memeriksa valid tidaknya data masukkan
	11. Menyimpan data ke basis data
	12. Menampilkan pesan sukses disimpan

### 3. Usecase Pengelolaan Purchase Barang

Pemodelan usecase pengelolaan purchase barang dapat dilihat pada gambar dibawah ini.



Gambar 8. Usecase Pengelolaan Purchase Barang

Berdasarkan gambar 8 diatas, definisi dan skenario usecase pengelolaan barang dapat dilihat pada tabel 7 dan tabel 8 dibawah ini.

Tabel 7. Definisi Use Case Pengelolaan Barang

Use Case	Deskripsi
<b>Login</b>	Merupakan proses untuk melakukan login staf
<b>Pengelolaan PO</b>	Merupakan proses mengelola proses create PO dan Revisi PO
<b>Pengelolaan penerimaan surat jalan</b>	Merupakan proses yang mengelola dokumentasi penerimaan surat jalan
<b>Pengelolaan pembayaran PO</b>	Merupakan proses yang mengelola dokumentasi dan pembayaran PO ke supplier

Tabel 8. Skenario Usecase Pengelolaan Purchase Barang

Aksi Aktor	Reaksi Sistem
<b>Skenario Use Case Create PO</b>	
<b>Skenario Normal</b>	
1. Memasukkan data PO	1. Memeriksa status login 3. Memeriksa valid tidaknya data masukkan 4. Menyimpan data tagihan ke basis data 5. Menampilkan pesan sukses disimpan
<b>Skenario Alternatif</b>	
6. Memasukkan data PO sesuai dengan kolom yang ada	7. Memeriksa valid tidaknya data masukkan 8. Mengeluarkan pesan bahwa data yang dimasukkan tidak valid
9. Memperbaiki data masukkan yang tidak valid	10. Memeriksa valid tidaknya data masukkan 11. Menyimpan data ke basis data 12. Menampilkan pesan sukses disimpan
<b>Skenario Use Case Revisi PO</b>	

Aksi Aktor	Reaksi Sistem
<b>Skenario Normal</b>	
	1. Memeriksa status login
2. Memilih no PO	3. Menampilkan no PO yang dipilih
4. Mervisi data	5. Menyimpan data PO ke basis data
	6. Menampilkan pesan sukses disimpan
<b>Skenario Alternatif</b>	
7. Memilih no PO	8. Menampikan no PO
9. Mervisi data	10. Memeriksa valid tidaknya data masukkan
	11. Mengeliarkan pesan bahwa data yang dimasukkan tidak valid
12. Memperbaiki data masukkan yang tidak valid	13. Memeriksa valid tidaknya data masukkan
	14. Menyimpan data ke basis data
	15. Menampilkan pesan sukses disimpan
<b>Skenario Use Case Pengelolaan Penerimaan Surat Jalan</b>	
<b>Skenario Normal</b>	
	1. Memeriksa status login
2. Memasukkan data surat jalan	3. Memeriksa valid tidaknya data masukkan
	4. Menyimpan data surat jalan ke basis data
	5. Menampilkan pesan sukses disimpan
<b>Skenario Alternatif</b>	
6. Memasukkan data surat jalan sesuai dengan kolom yang ada	7. Memeriksa valid tidaknya data masuksn
	8. Mengeliarkan pesan bahwa data yang dimasukkan tidak valid
9. Memperbaiki data masukkan yang tida valid	10. Memeriksa valid tidaknya data masukkan
	11. Menyimpan data ke basis data
	12. Menampilkan pesan sukses disimpan
<b>Skenario Use Case Pengelolaan Pembayaran PO</b>	
<b>Skenario Normal</b>	
	1. Memeriksa status login
2. Memasukkan data pembayaran PO	3. Memeriksa valid tidaknya data masukkan
	4. Menyimpan data pembayaran PO ke basis data
	5. Menampilkan pesan sukses disimpan
<b>Skenario Alternatif</b>	
6. Memasukkan data pembayaran PO sesuai dengan kolom yang ada	7. Memeriksa valid tidaknya data masuksn
	8. Mengeliarkan pesan bahwa data yang dimasukkan tidak valid
9. Memperbaiki data masukkan yang tidak valid	10. Memeriksa valid tidaknya data masukkan
	11. Menyimpan data ke basis data
	12. Menampilkan pesan sukses disimpan

## Kesimpulan

Berdasarkan penelitian yang dilakukan, penelitian ini membahas mengenai use case diagram dan use case scenario: use case diagram sistem transaksi penjualan, pengelolaan sales atau penjualan, pengelolaan warehouse, dan usecase pengelolaan purchase barang yang dimodelkan menggunakan UML. Dari pemodelan yang dihasilkan, terdapat empat pengguna (aktor) yang terlibat dalam sistem ini yang terdiri dari Sales, Warehouse, Purchase, dan Admin. Pemodelan use case ini diharapkan dapat memberikan rekomendasi alur sebuah sistem yang telah dibuat dalam penelitian ini, karena use case membantu untuk memahami kebutuhan sistem dan berkomunikasi dengan pengguna akhir, sehingga sistem yang akan dibangun memiliki fungsi yang sesuai dengan kebutuhan pengguna.

## Daftar Pustaka

- Arifin, M., & Hs, R. H. H. (2017). PERANCANGAN SISTEM INFORMASI PUSAT KARIR SEBAGAI UPAYA MENINGKATKAN RELEVANSI ANTARA LULUSAN DENGAN DUNIA KERJA MENGGUNAKAN UML Muhammad. *IC-Tech*, *XII*(2), 42–49.
- Budi, D. S., Siswa, T. A. Y., & Abijono, H. (2016). Analisis Pemilihan Penerapan Proyek Metodologi Pengembangan Rekayasa Perangkat Lunak. *TEKNIKA*, *5*(1), 24–31.
- Egy Muhammad Rianof, Bambang P. Adhi, & Z.E. Ferdi F. Putra. (2020). Pengembangan Aplikasi M-Commerce Pada Toko Optik Menggunakan Android Studio. *PINTER : Jurnal Pendidikan Teknik Informatika Dan Komputer*, *4*(2), 15–18.  
<https://doi.org/10.21009/pinter.4.2.3>
- Fowler, M., & Scott, K. (1993). UML Distilled Second Edition A Brief Guide to the Standart Object Modeling Language. In *Assidon Wesley* (Vol. 31).
- Iskandar, K. (2012). Menggunakan Diagram Use Case Dan Rich Picture. *ComTech*, *3*(1), 654–662.
- Kurniawan, T. A. (2018). Pemodelan Use Case (UML): Evaluasi Terhadap beberapa Kesalahan dalam Praktik. *Jurnal Teknologi Informasi Dan Ilmu Komputer*, *5*(1), 77.  
<https://doi.org/10.25126/jtiik.201851610>
- McDermid, J., & Rook, P. (2019). Software development process models. In *Software Engineer's Reference Book* (Second Edi). Elsevier Inc. <https://doi.org/10.1016/b978-0-7506-0813-8.50022-8>
- Migunani. (2007). Microsoft Solution Framework sebagai Model Proses Pengembangan Perangkat Lunak Berbasis Milestone , Tinjauan pada Fase Envisioning dan Planning. *Jurnal Teknologi Informasi DINAMIK*, *XII*(2).
- Nabyla, F., & Sigitta, R. C. (2019). Desain Aplikasi Sistem Pendaftaran Online Menggunakan Smartphone Untuk Meningkatkan Mutu Pelayanan Pada Rumah Sakit. *JOINS (Journal*

- of Information System), 4(2), 168–177. <https://doi.org/10.33633/joins.v4i2.3078>
- Okesola, O. J., Adebiyi, A. A., Owoade, A. A., Adeaga, O., Adeyemi, O., & Odun-Ayo, I. (2020). Software Requirement in Iterative SDLC Model. In *Advances in Intelligent Systems and Computing: Vol. 1224 AISC* (Issue August). Springer International Publishing. [https://doi.org/10.1007/978-3-030-51965-0\\_2](https://doi.org/10.1007/978-3-030-51965-0_2)
- Pressman, R. S. (2010). *No Title Software Engineering : a practitioner's approach* (6th ed.). McGraw-Hill.
- Raharja, I. M. S., & Sn, A. (2012). Perbandingan Proses Pengembangan Perangkat Lunak Model Spiral Dan Cleanroom. *Seminar Nasional Informatika 2015 (SemnasIF 2015), 2012(semnasIF)*, 103–109.
- Rodríguez, P., Mäntylä, M., Oivo, M., Lwakatare, L. E., Seppänen, P., & Kuvaja, P. (2019). Advances in Using Agile and Lean Processes for Software Development. *Advances in Computers*, 113, 135–224. <https://doi.org/10.1016/bs.adcom.2018.03.014>
- Sunardi, S., & Fadli, S. (2018). IDENTIFIKASI MASALAH PENERAPAN METODE AGILE (SCRUM) PADA PENGEMBANGAN PERANGKAT LUNAK DI PERGURUAN TINGGI (Studi Kasus Universitas Nahdlatul Ulama Nusa Tenggara Barat). *Jurnal Manajemen Informatika Dan Sistem Informasi*, 1(2), 14. <https://doi.org/10.36595/misi.v1i2.37>
- Syakti, F. (2019). METODE PENGEMBANGAN PERANGKAT LUNAK BERBASIS MOBILE : A REVIEW. *Jurnal Bina Komputer*, 1(2), 82–89.
- Wira, D., Putra, T., & Andriani, R. (2019). *Unified Modelling Language ( UML ) dalam Perancangan Sistem Informasi Permohonan Pembayaran Restitusi SPPD*. 7(1).