

## BAB II LANDASAN TEORI

### 2.1 Teori Tentang Topik Skripsi

#### 2.1.1 *Enterprise Resource Planning (ERP)*

*Enterprise Resource Planning* merupakan metode bagi industri dalam mengupayakan proses bisnis yang lebih efisien dengan membagi informasi di dalam dan antar bisnis proses dan menjalankan bisnis secara elektronik [3]. Penelitian ini bertujuan untuk menyempurnakan PT Rafa Indonesia Sejahtera agar memiliki sistem terintegrasi yang mudah diakses dengan *database* yang terhubung, pengelolaan data barang dagangan lebih efektif dan tidak memakan waktu lama. Implementasi ERP secara strategis dapat meningkatkan proses internal organisasi.

Meskipun ada banyak keuntungan menggunakan ERP, banyak sistem yang tidak dapat diimplementasikan. Kegagalan umumnya terjadi karena adanya ketidaksesuaian metode yang diterapkan dalam menganalisis suatu permasalahan yang terjadi di Perusahaan. Oleh karena itu, diperlukan penelitian untuk membantu memahami cara kerja penerapan ERP dan bagaimana penerapan tersebut dapat bermanfaat bagi proses bisnis Perusahaan. Dalam melakukan penelitian terhadap modul ini, diperlukan faktor terpenting untuk menjamin keberhasilan implementasi ERP yaitu *Key Success Factor*.

#### 2.1.2 *Key Success Factor*

*Key success factor* merupakan faktor utama bagi suatu perusahaan maupun dalam *project management* sebagai kunci utama dalam keberhasilan yang ditetapkan dari perusahaan itu sendiri [8]. Setiap perusahaan ataupun *project management* mempunyai jenis kunci kesuksesan yang berbeda, dengan adanya *Key success factor* akan memberikan batas di mana aplikasi yang telah ditetapkan akan berjalan

sesuai dengan yang diharapkan. *Key success factor* juga diartikan sebagai fondasi untuk mengawal setiap *project managemenet* dan untuk contoh analisa sebagai faktor keberhasilan implementasi ERP.

Faktor kunci keberhasilan sistem ERP meliputi dukungan dan komitmen manajemen, rekayasa ulang proses bisnis (BPR), kinerja ERP, pelatihan dan pendidikan, integrasi [8]. Sumber daya perusahaan perlu dikelola dengan baik. Hal inilah yang menyebabkan *software* ERP memerlukan modul agar dapat berfungsi sesuai keinginan. Modul ERP terdiri dari beberapa modul antara lain pembelian, penjualan, HRMS, keuangan, CRM, *e-commerce*, gudang dan inventaris [3]. Salah satu modul yang digunakan di PT Rafa Indonesia Sejahtera adalah modul pembelian. Sederhananya berisi sistem pencatatan pemasukan dan pengeluaran barang yang terintegrasi guna meminimalisir terjadinya kesalahan dalam persediaan barang di PT Rafa Indonesia Sejahtera.

### 2.1.3 Pembelian

Pembelian merupakan kegiatan utama dalam menjamin suatu kelancaran transaksi penjualan yang terjadi dalam suatu perusahaan. Adanya pembelian dapat memudahkan perusahaan untuk menyediakan sumber daya yang diperlukan organisasi secara efektif dan efisien. Pembelian merupakan akun yang digunakan untuk mencatat semua pembelian barang dagang dalam satu periode [9]. Pada suatu perusahaan dagang kegiatan pembelian meliputi pembelian aktiva produksi, pembelian barang dagangan serta pembelian barang dan jasa lain dalam rangka kegiatan usaha. Pembelian terdiri atas 2 (dua) jenis yaitu pembelian tunai dan kredit [9]. Pembelian tunai adalah pembelian yang dibayar dengan uang kas perusahaan, sedangkan pembelian kredit merupakan pembelian yang terjadi dengan adanya syarat tertentu.

#### 2.1.4 Modul Pembelian

Suatu perusahaan tidak akan bisa beroperasi dengan baik tanpa adanya kegiatan pembelian. Dengan kata lain, kegiatan pembelian merupakan salah satu fungsi dasar dari sebuah perusahaan. Pengelolaan dari fungsi pembelian haruslah dilakukan secara sungguh-sungguh. Modul pembelian merupakan modul yang digunakan dalam rangka untuk manajemen persediaan barang dan membantu kebutuhan perusahaan dalam membeli barang dan jasa. Dalam modul pembelian biasanya memuat penjelasan dokumen dan fungsi yang digunakan dalam proses pembelian dan juga mengikuti perubahan inventaris selama proses pembelian. Setiap dokumen pembelian mempengaruhi jumlah persediaan. Dan pesanan pembelian mempengaruhi jumlah persediaan yang tersedia. Singkatnya, bahwa dalam modul pembelian memuat permintaan penawaran, *order* pembelian, data pemasok (*supplier*), produk masuk, dan tagihan pembelian [22].

Modul pembelian dalam ERP tidak hanya berfokus pada pembelian barang namun juga meliputi permintaan, meminta penawaran barang atau jasa yang dibutuhkan kepada pemasok sebagai penyedia barang dan jasa, menerima dan menyeleksi surat penawaran dari pemasok, dan menganalisa pemasok agar mendapatkan pemasok yang terbaik dalam pembelian barang. Modul pembelian meliputi pembuatan *purchase requisition*, *request for Quotation*, *maintain Quotation*, *vendor* atau *supplier analysis*, *Purchase order*, dan segala *report* mengenai *purchasing* [23].

##### 1) *Purchase Requisition*

*Purchase Requisition* adalah sarana untuk membuat permintaan pembelian, agar pihak *purchasing/buyer* dapat melakukan proses pengadaan barang yang diminta. *Purchase Requisition* merupakan awal dari sebuah proses pengadaan barang. Barang yang diminta adalah barang yang terdaftar pada *Item Master*. *Purchase*

*Requisition* berisi tentang jenis barang yang diminta, deskripsi atau detail barang yang diminta, dan tanggal barang tersebut dibutuhkan.

2) *Request for Quotation*

*Request for Quotation* (RfQ) adalah dokumen yang berisi permintaan penawaran secara resmi kepada *supplier*. RfQ atau yang lebih dikenal dengan surat permintaan penawaran berisi tentang detail barang yang diminta, jumlahnya, alamat pengiriman serta tanggal batas pengiriman *Quotation*. RfQ bertujuan untuk mendapatkan *Quotation order* atau surat penawaran dari *supplier* secara resmi.

3) *Quotation Order*

*Quotation Order* adalah surat penawaran barang yang diberikan oleh *supplier*, berisi tentang detail barang, harga barang, diskon, biaya pengiriman, dan pengeluaran untuk pajak. *Quotation* yang didapatkan dari *supplier* akan dianalisa oleh bagian *purchasing* dan menentukan apakah akan menerima atau menolak penawaran tersebut. *Quotation order* adalah patokan untuk membuat *purchase order*.

4) *Supplier Analysis*

*Supplier Analysis* adalah sebuah analisa untuk menentukan rekomendasi *supplier* untuk pembelian barang berdasarkan dari transaksi pembelian dan penerimaan barang yang telah dilakukan sebelumnya. Analisa ini bertujuan agar pihak *purchasing* tidak salah dalam memilih *supplier* dan mendapatkan barang dengan kualitas, dan harga yang terbaik.

5) *Purchase Order*

*Purchase Order* atau yang biasa lebih dikenal dengan PO merupakan salah satu jenis surat bisnis yang sering ditemui. Saat

*purchasing* melakukan *order* atau pemesanan kepada *supplier*, *purchasing* wajib untuk dikirimkan. Surat ini akan digunakan *supplier* untuk mengetahui detail barang-barang apa saja yang dipesan. Ini berguna untuk mengatasi kesalahan dalam memproduksi pesanan. Selain itu, surat *purchase order* juga akan digunakan *supplier* (dan juga *purchasing*) sebagai *file* dan bagi bagian keuangan, surat *purchase order* ini akan dijadikan bukti transaksi laporan keuangan. Di dalam *purchase order* harus membuat dengan jelas jenis yang dipesan, spesifikasi/permintaan khusus bila ada, tanggal pengiriman, serta yang tidak kalah pentingnya adalah harus mencantumkan nomer PO.

## 2.2 Teori tentang *Framework*

### 2.2.1 *System Development Life Cycle (SDLC)*

SDLC merupakan metode rekayasa perangkat lunak dalam mengimplementasi sistem. Proses SDLC dimulai dari proses pembuatan dan perubahan sistem yang dikembangkan dalam sebuah sistem. Tahapan metode SDLC diantaranya *planning*, *analysis*, *design*, *implementation*, *testing* dan *maintenance*. SDLC merupakan konsep untuk pengembangan lunak.

Metode SDLC untuk membantu kerangka kerja dalam membuat sistem informasi dan pengembangan perangkat lunak. Metode SDLC memiliki banyak jenis dan beberapa yang banyak digunakan merupakan *life cycle using prototyping*, *traditional system life cycle* dan *object-oriented system life cycle*. Kerangka kerja SDLC banyak digunakan Perusahaan IT. Sehingga dapat mempercepat proses pengembangan dan meminimalkan risiko proyek terkait waktu dan biaya yang diperlukan. SDLC memiliki tahapan metode sebagai berikut

[10]:

- 1) *Planning*

Tahapan *planning* merupakan tahap untuk melakukan perencanaan bersama tim dalam mengidentifikasi dan menentukan *scope* atau ruang lingkup untuk dilakukan dalam proses. Dalam tahap ini mengumpulkan semua informasi dari perusahaan yang dibutuhkan dalam proses pengembangan *software* dari pemangku kepentingan. Selanjutnya melakukan struktur tim, *time frame*, *budget*, *security*, dan berbagai faktor yang penting untuk pengembangan *software*.

2) *Analysis*

Tahapan *analysis* adalah proses lanjutan dari tahapan *planning* yang ditujukan untuk melakukan analisis terhadap kebutuhan fungsional sistem. Sehingga dalam tahap ini dapat mengetahui masalah bisnis dan target dicapai untuk pengembangan *software*. Dengan tujuan untuk mendapatkan hasil akhir yang sesuai dengan ekspektasi perusahaan.

3) *Design*

Tahapan selanjutnya melakukan pembuatan desain yang diinginkan bagi perusahaan. Terdapat beberapa aspek desain yang akan ditentukan yaitu *architecture*, *user interface*, *platform*, *security*. *Design* tersebut digunakan untuk kepentingan perusahaan dan mendapatkan *feedback* sehingga dapat membantu tim dalam proses pembuatan.

4) *Development*

Fase *development* merupakan pengembangan *software*. Tahap ini dimulai dengan cara tim pengembangan memulai membangun sistem dengan membuat kodingan dengan bahasa pemrograman yang dipilih. Dan selanjutnya menentukan *database admin* yang membuat menyimpan data.

5) *Testing*

Tahapan SDLC *testing* untuk melakukan pengujian sistem dan menilai *software* yang telah dibuat dan melakukan pengujian dapat berkerja sesuai dengan fungsional yang telah diharapkan.

#### 6) *Implementation*

Tahapan terakhir SDLC merupakan hasil dari semua yang telah dibuat dan melanjutkan ke tahap implementasi. Tujuan dari tahapan ini adalah untuk membantu *user* untuk menggunakan yang telah dikembangkan.

Konsep SDLC memiliki beberapa model perangkat lunak lainnya seperti *waterfall iterative*, *spiral*, *prototype* dan *rapid application development* (RAD). Didalam penelitian ini menggunakan SDLC metode *Rapid Application Development* (RAD).

#### 2.2.2 *Rapid Application Development* (RAD)

RAD merupakan model dalam tahapan pengembangan perangkat lunak dan fokusnya pada siklus waktu pengembangan yang jangka pendek sehingga metode sekuensial lineral dengan waktu perkembangan sekitar 60 (enam puluh) hingga 90 (sembilan puluh) hari. Model ini melakukan adaptasi terhadap kecepatan dalam waktu pengembangan yang cepat dan melakukan pendekatan dengan kontruksi berbasi komponen. Motode RAD dipilih karena memiliki kelebihan yaitu mampu menghasilkan sistem yang sesuai serta pengujian dilakukan berdasarkan fungsi.

Metode RAD merupakan suatu model pengembangan sistem dengan pendekatan berorientasi objek yang terdiri dari pengembangan dan perangkat-perangkat lunak. Adapun tahapan metode RAD sebagai berikut [2]:

##### 1) Perencanaan (*Requirement Planning*)

Tahap ini dilakukan untuk melakukan identifikasi tujuan aplikasi, kebutuhan informasi yang didapatkan dari proses pengumpulan data antara pengguna dengan analisis

##### 2) Proses Desain (*RAD Design Workshop*)

Tahap ini yaitu tahapan setelah objek data dikumpulkan dan didefinisikan, yangmana dari hasil analisis kebutuhan akan digunakan untuk mendefinisikan proses bisnis dalam membangun sebuah sistem. Pada tahap ini terbagi menjadi 2 (dua) tahapan, yaitu:

a) Desain Sistem (*Design System*)

Tahap ini dilakukan dengan membuat desain sistem sesuai dengan perencanaan. Tahap desain sistem dapat dilakukan secara berulang sampai kebutuhan sistem sesuai dengan keinginan pengguna.

b) Membangun Sistem (*Build The System*)

Selanjutnya setelah desain sistem sudah dipastikan telah sesuai dengan kebutuhan pengguna, maka akan dilakukan pengkodean dalam bahasa pemrograman yang dilakukan oleh *programmer* sesuai dengan kebutuhan pengguna, dan komputer dapat mengenali dari suatu fungsi tertentu.

3) Implementasi (*Implementation*)

Tahap ini dilakukan dengan cara mengimplementasi terhadap sistem yang telah dibuat oleh *programmer* dan telah disetujui oleh pengguna. Pada tahap implementasi, pengguna akan memberikan *feedback* terhadap sistem yang telah jadi dan digunakan pengguna.


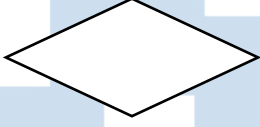

### 2.2.3 Entity Relationship Diagram (ERD)

ERD adalah diagram yang menggambarkan tipe dari entitas di dalam sistem yang akan dijelaskan dalam data dengan atribut data dan menghubungkan atau relasi entitas satu dengan yang lain. Terdapat beberapa simbol simbol dalam ERD ditampilkan pada Tabel 2.1 [12]

**Tabel 2. 1 ERD**

Simbol	Nama	Fungsi
--------	------	--------



	Entitas	Simbol ini ditujukan menggambarkan obyek untuk mewakili data dan memiliki pembeda data satu dengan data yang lain ( <i>unique</i> ).
	Relasi	Simbol ini bertujuan menguraikan hubungan entitas yang satu dengan entitas yang lain.
	Atribut	Simbol ini digunakan untuk mendefinisikan sifat-sifat entitas berupa nama kolom pada baris data.

## 2.3 Teori Tentang *Tools*

### 2.3.1 Laravel

Laravel merupakan sebuah *framework* bahasa pemrograman PHP yang digunakan untuk sebuah pengembangan aplikasi web [13]. *Taylor Otwell* merupakan pencipta *framework Laravel*. Aplikasi yang dikembangkan memiliki pola MVC untuk menghubungkan sebuah *request* dari *user* dan *controller*.

MVC merupakan sebuah konsep arsitektur yang terdiri dari tiga komponen utama yaitu *model*, *view* dan *controller*. Dan dalam setiap komponen memiliki tanggung jawab yang berbeda-beda dalam proses pengembangan aplikasi. *Model* berfungsi untuk bertanggung jawab dalam mengelola data dan menyimpan informasi dalam sistem. *View* berfungsi untuk menampilkan data ke pengguna. *Controller* merupakan bagian untuk mengontrol alur kerja sistem memiliki hubungan antara *model* dan *view*.

### 2.3.2 SQL

*Structured Query language* (SQL) merupakan bahasa pemrograman untuk menyimpan data dan memproses informasi dalam basis data relasional [14]. SQL biasanya sebagai perintah yang terdapat pada sistem Manajemen Bisnis Dara (SMBD). SMBD didukung oleh *MySQL Server, MySQL, Interbase* dan *Oracle*.

### 2.3.3 *Blackbox Testing*

Pengujian *black box* merupakan pengujian terhadap setiap spesifikasi fungsional perangkat lunak. Saat melakukan pengujian, kita dapat menentukan kumpulan *input* dan menguji fungsionalitas perangkat lunak [15]. Metode pengujian *black box* terdiri dari beberapa metode partisi ekuivalen, analisis nilai batas, pengujian transisi keadaan dan pengujian tabel keputusan *Black box Testing* memungkinkan pengembang perangkat lunak memperoleh sekumpulan kondisi masukan dari fungsional program. Dalam pengujian *blackbox testing* digunakan untuk menemukan kesalahan seperti:

- 1) Mengetahui struktur data yang salah;
- 2) terdapat fungsional sistem yang tidak berjalan;
- 3) proses insialisasi dan terminasi tidak sesuai ;
- 4) adanya batasan dari data; dan
- 5) terjadinya *interface* yang salah.

### 2.3.4 *User Acceptance Test (UAT)*

*User Acceptance Test* digunakan untuk menentukan sistem yang dikembangkan telah memenuhi kebutuhan pengguna atau belum maksimal [15]. Sehingga pengujian ini dilakukan setelah melakukan *Unit Testing, Integration Testing, dan System Testing* selesai dan menggunakan metode *Black Box Testing*.

*Acceptance testing* biasanya berusaha menunjukkan bahwa sistem telah memenuhi persyaratan-persyaratan tertentu [15]. Pada pengembangan *software* dan *hardware* komersial, *acceptance test* biasanya disebut juga "*alpha tests*" (yang dilakukan oleh pengguna *in-house*) dan "*beta tests*" (yang dilakukan oleh pengguna yang sedang menggunakan atau akan menggunakan sistem tersebut). *Alpha* dan *beta test* biasanya juga menunjukkan bahwa produk telah siap untuk dijual atau dipasarkan. *Acceptance testing* mencakup data, *environment* dan skenario yang sama atau hampir sama pada saat *live* yang biasanya berfokus pada skenario penggunaan produk tertentu.

Tes *alfa* dan *beta* juga biasanya menunjukkan bahwa produk tersebut siap untuk dijual atau dipasarkan [15]. Pengujian pengguna biasanya dilakukan oleh pelanggan atau pengguna akhir dan tidak hanya berfokus pada mengidentifikasi masalah kecil seperti kesalahan ketik, cacat tampilan, atau kerusakan perangkat lunak, tetapi juga mencakup masalah lainnya. Hasil pengujian ini memberikan keyakinan kepada pelanggan mengenai seberapa siap sistem untuk produksi. Dalam pengembangan perangkat lunak, UAT disebut juga pengujian beta, pengujian aplikasi, dan pengujian pengguna akhir, yaitu *fase* pengembangan perangkat lunak ketika perangkat lunak diuji di dunia nyata.

### **2.3.5 Unified Modeling Language (UML)**

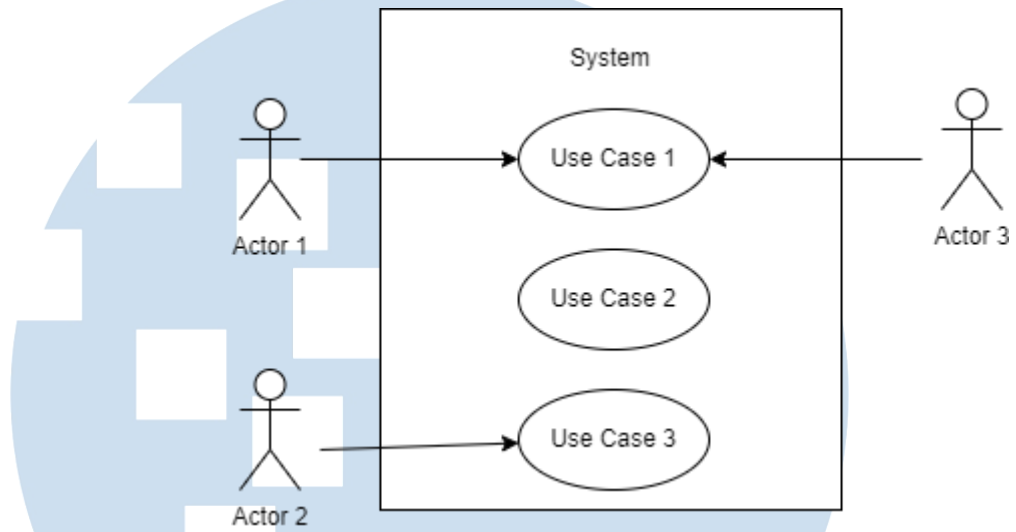
*Unified Modeling Language* (UML) merupakan suatu bahasa yang ditujukan untuk menentukan, memvisualisasikan, membangun, dan mendokumentasikan suatu sistem informasi. UML sendiri dikembangkan sebagai suatu alat untuk analisis dan desain berorientasi objek oleh Grady Booch, Kim Rumbaugh, dan Ivar Jacobson [16]. Jenis diagram UML yang digunakan dalam penelitian ini, yaitu:

- 1) Diagram Model *Use Case*

Model diagram *use case* menggambarkan interaksi antara sistem dengan sistem eksternal dan pengguna. Jika menggunakan kasus yang berbeda, kita akan dapat menggunakan sistem yang berbeda dan jumlah orang yang berbeda serta berinteraksi satu sama lain [16]. Hal-hal yang berkaitan dengan diagram *use case*, yaitu:

- a) *Use Case* merupakan urutan langkah-langkah yang secara tindakan saling terkait (skenario), baik terotomatis maupun secara manual, untuk tujuan melengkapi satu tugas bisnis tunggal.
- b) *Actor* (Pelaku) adalah segala sesuatu yang perlu berinteraksi dengan sistem dengan tujuan untuk pertukaran informasi. Aktor tidak harus manusia, dapat berupa perusahaan, alat eksternal yang lain seperti sensor panas, *mainframe* di sistem lama dimana diperoleh data atau perangkat lunak memperoleh data seperti *keypad* di mesin ATM.
- c) *Relationship* (Hubungan), terdiri menjadi 5 (lima) yaitu :
  - (1) *Use Case Association Relationship*, adalah relasi antara *actor* dengan *use case*. Asosiasi digambarkan dengan garis yang menghubungkan antara *actor* dengan *use case*.
  - (2) *Use Case Extends Relationship*, adalah sebuah *use case* terdiri dari langkah-langkah yang diambil dari sebuah *use case* yang lebih kompleks untuk mempermudah *case* aslinya sehingga memperluas fungsionalitasnya.
  - (3) *Use Case Uses Relationship*, *use case* yang dapat mengurangi redundansi antara dua atau lebih *use case* dengan mengkombinasikan langkah-langkah serupa yang ditemukan dalam *case-case* tersebut.
  - (4) *Use Case Depends On Relationship*, adalah relasi *use case* yang menentukan *use case* lain mana yang harus dilakukan sebelum *use case* yang bersangkutan.
  - (5) *Use Case inheritance Relationship*, adalah relasi *use case* dimana perilaku serupa dari 2 (dua) *actor* yang memulai *use case* yang

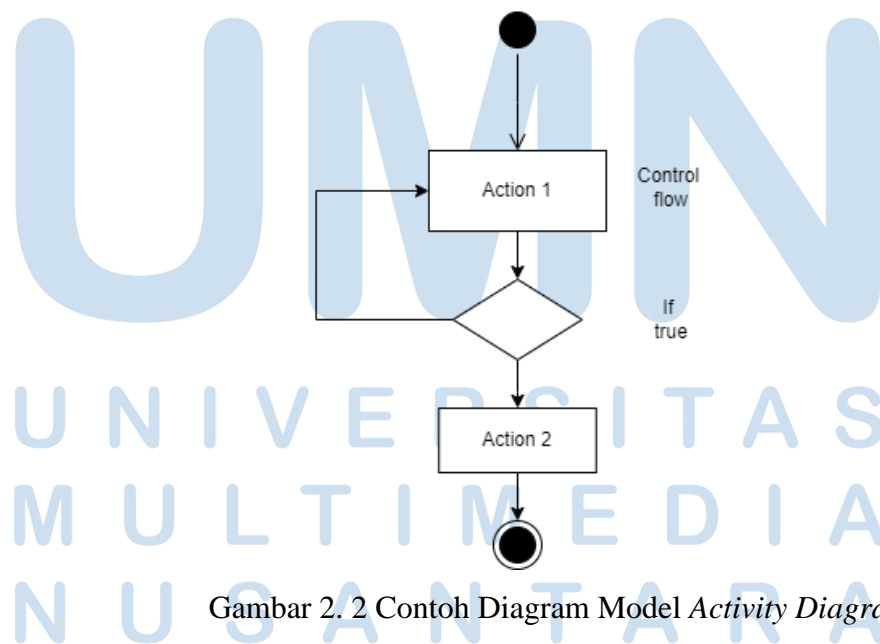
sama diekstrapolasi dan ditugaskan kepada 1 (satu) *abstract actor* untuk mengurangi redundansi.



Gambar 2. 1 Contoh Diagram Model *Use Case Diagram*

## 2) Diagram aktivitas

Secara grafis, diagram aktivitas ditujukan untuk menggambarkan rangkaian aliran aktivitas baik proses bisnis atau *use case* [16]. *Activity* diagram secara grafis digunakan untuk menggambarkan rangkaian aliran aktivitas baik proses bisnis atau *use case*.

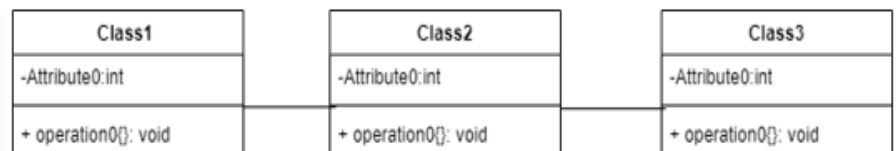


Gambar 2. 2 Contoh Diagram Model *Activity Diagram*

### 3) Diagram kelas

Diagram kelas menggambarkan struktur objek sistem. Diagram ini menunjukkan kelas objek yang membentuk sistem dan hubungan antara kelas objek tersebut. Diagram kelas merupakan penggambaran struktur dan deskripsi kelas, paket, dan objek satu sama lain. Diagram kelas juga menunjukkan properti suatu kelas, operasinya, dan batasan dalam hubungan ini. Diagram kelas adalah gambaran grafis dari struktur objek statis suatu sistem, yang menunjukkan kelas-kelas objek yang membentuk sistem serta hubungan antara kelas-kelas objek tersebut. Diagram kelas memuat 3 (tiga) jenis hubungan, yaitu [16]:

- a) *Association*, suatu hubungan antara bagian dari dua kelas yang terjadi jika salah satu bagian dari kelas mengetahui kelas yang lain dengan melakukan suatu kegiatan.
- b) *Aggregation*, hubungan *association* dimana salah satu kelasnya merupakan bagian dari suatu kumpulan dan memiliki titik pusat yang mencakup keseluruhan bagian.
- c) *Generalization*, hubungan turunan dengan mengasumsikan satu kelas merupakan suatu kelas super dari kelas yang lain.



Gambar 2. 3 Contoh Model *Class Diagram*

## 2.4 Penelitian Terdahulu

Tabel 2. 2 Penelitian Terdahulu

No	Nama Peneliti (Tahun)	Topik	Hasil Penelitian Terdahulu
1	Nurman Hidayat dalam Jurnal	Penerapan Metode <i>Rapid</i>	Aplikasi ini dibuat dengan menggunakan <i>CodeIgniter 3</i> untuk <i>frameworknya</i> , PHP

	Sistem Informasi STMIK Antar Bangsa, vol. 10, no. 1, pp. 8-10, 2021.	<i>Application Development</i> (RAD) dalam Rancang Bangun Sistem Informasi Rapor <i>Online</i> (SIRALINE)	untuk bahasa pemrogramannya, MySQL untuk <i>databasenya</i> dan <i>Visual Studio Code</i> sebagai alat bantu untuk membantu pembuatan sistem informasinya. Kesimpulannya, sistem yang dirancang sangat bermanfaat dalam proses pengolahan raport hingga mencetak hasil laporan yang tersampaikan dengan baik dan cepat [6].
2	Bilal Abdul Wahid dalam Jurnal Esensi Infokom, vol. 3, no. 1, pp. 34, 2019.	Penerapan Metode <i>Rapid Application Development</i> (RAD) Terhadap Penjualan <i>Fashion Distro</i> Secara <i>Online</i>	Penelitian ini bertujuan untuk menghasilkan sistem penjualan <i>online Fashion Distro</i> dengan mengintegrasikan ke dalam <i>database</i> dan membangun hubungan dengan pengguna sebagai pembeli secara lebih personal. Model yang digunakan untuk memodelkan sistem menggunakan <i>Unified Modeling Language</i> (UML). Dampak dari hasil pengujian terlihat pada proses digitalisasi penjualan produk dengan verifikasi data yang lebih akurat pada sistem navigasi, validasi data isi keranjang belanja, pelacakan proses data pesanan, dan konfirmasi pembayaran [7].
3	Yahya Dwi Wijaya, dalam jurnal Sistem Informasi dan Teknologi, vol. 3, no. 2, 2020.	Penerapan Metode <i>Rapid Application Development</i> (RAD) Dalam Pengembangan	Penelitian ini menjelaskan bahwa pengembangan sistem informasi pada Toko Berkah bertujuan untuk menyediakan sistem informasi pada Toko Berkah untuk digunakan dalam pengolahan data toko dan sebagai bentuk

		Sistem Informasi Data Toko	keamanan data agar tidak dapat diakses oleh sembarang orang [16].
4	Dicky Hariyanto dalam Jurnal Jupiter, vol. 13, no. 1, pp. 111, 2021.	Implementasi Metode <i>Rapid Application Development</i> Pada Sistem Informasi Perpustakaan	Penelitian tersebut menghasilkan kesimpulan berupa penerapan teknologi informasi dan komunikasi di dalam perpustakaan sangat membantu mengingat untuk meminimalisir terjadinya masalah akibat pencatatan data yang masih konvensional. Selain itu, metode RAD dipilih karena memiliki kelebihan dalam menghasilkan sebuah sistem yang cepat dan berkualitas [17].
5	Anak Agung Gde Mahendra Yana dalam Jurnal Ilmiah Teknologi dan Komputer, vol. 3, no. 3, pp. 57, 2022.	Penerapan <i>Enterprise Resource Planning (ERP)</i> Menggunakan Odoo 14 (Studi Kasus: Usaha Garmen Pada Club Ink Bali).	Penerapan <i>Enterprise Resource Planning (ERP)</i> pada kasus ini meliputi modul <i>sales</i> , modul <i>purchase</i> , modul <i>point of sale</i> , modul <i>inventory</i> , modul <i>manufacturing</i> , modul <i>invoice</i> , dan modul <i>website</i> . Penerapan modul tersebut sangat baik dikarenakan Club Ink Bali belum memiliki sistem informasi yang terintegrasi [18].
6	Fitri Purwaningtias dalam Jurnal Komputer, vol 6, no 3, pp. 276, 2019.	Pemodelan <i>Enterprise Resource Planning</i> Pada Z-Tech Komputer.	Hasil implementasi menunjukkan bahwa pemodelan UML dan implementasi menggunakan <i>Odoo</i> dapat berjalan sesuai dengan kebutuhan. Hal ini dikarenakan <i>Odoo</i> memiliki beragam modul yang saling terintegrasi, baik modul manajemen pelanggan, akuntansi, penjualan, dan bahkan pembelian [22].



--	--	--	--

Pada tabel 2.2 merupakan penelitian terdahulu yang dibuat sebagai dasar pembuatan penelitian. Berikut merupakan penjelasan dari tabel tersebut:

1. Penelitian pertama dilakukan untuk mencari tahu tentang penerapan metode *rapid application development* pada rancang bangun sistem. Penelitian ini menghasilkan informasi aplikasi yang dengan *framework*, PHP untuk bahasa pemrograman, *MySQL* untuk *database* dan *visual studio* sebagai *tools*. Sehingga membangun sistem yang dirancang dapat tersampaikan dengan baik dan cepat.
2. Penelitian kedua dilakukan untuk mencari manfaat *database* dan sistem yang terintegrasi dengan baik sehingga mendapatkan hasil sistem yang baik. Dalam penerapan *rapid application development* dapat menggunakan *unified modeling language*.
3. Penelitian ketiga dengan topik Penerapan Metode *Rapid Application Development* (RAD) Dalam Pengembangan Sistem Informasi Data Toko. Penelitian ini menunjukkan bahwa pengembangan sistem informasi pada Toko Berkah bertujuan untuk menghadirkan sistem informasi kepada Toko Berkah untuk digunakan sebagai pengolahan data toko dan bentuk pengamanan data agar tidak diakses oleh sembarang orang.
4. Penelitian keempat dengan topik Implementasi Metode *Rapid Application Development* Pada Sistem Informasi Perpustakaan. Penelitian tersebut menghasilkan kesimpulan berupa penerapan teknologi informasi dan komunikasi di dalam perpustakaan sangat membantu mengingat untuk meminimalisir terjadinya masalah akibat pencatatan data yang masih konvensional. Selain itu, metode RAD dipilih karena memiliki kelebihan dalam menghasilkan sebuah sistem yang cepat dan berkualitas.

5. Penelitian kelima dengan topik Penerapan *Enterprise Resource Planning* (ERP) Menggunakan Odoo 14 (Studi Kasus: Usaha Garmen Pada Club Ink Bali). Penelitian tersebut menghasilkan kesimpulan berupa penerapan *Enterprise Resource Planning* (ERP) pada kasus ini meliputi modul *sales*, modul *purchase*, modul *point of sale*, modul *inventory*, modul *manufacturing*, modul *invoice*, dan modul *website*. Penerapan modul tersebut sangat baik dikarenakan Club Ink Bali belum memiliki sistem informasi yang terintegrasi.
6. Penelitian keenam dengan topik Pemodelan *Enterprise Resource Planning* Pada *Z-Tech* Komputer. Hasil implementasi menunjukkan bahwa pemodelan UML dan implementasi menggunakan *Odoo* dapat berjalan sesuai dengan kebutuhan. Hal ini dikarenakan *Odoo* memiliki beragam modul yang saling terintegrasi, baik modul manajemen pelanggan, akuntansi, penjualan, dan bahkan pembelian.

UMMN

UNIVERSITAS  
MULTIMEDIA  
NUSANTARA