

BAB 2

LANDASAN TEORI

Pada bab ini menjelaskan tentang landasan teori yang akan digunakan untuk perancangan *e-Learning* di Rumah Sakit PELNI. Teori yang digunakan terdiri dari 3 bagian, antara lain teori umum, teori khusus, dan teori metodologi. Teori umum yang akan dibahas, yaitu mengenai sistem informasi, internet, data, dan teknologi. Teori khusus yang akan dibahas yaitu mengenai *e-Learning*, *web-based*, dan *mobile*. Teori metodologi yang akan dibahas terdiri dari 4 bagian, antara lain metode pengumpulan data (studi pustaka, examining documentation, wawancara, dan kuesioner), teori analisis SWOT menggunakan SWOT Matriks, teori model desain pembelajaran menggunakan ADDIE, dan teori perancangan menggunakan *Object-Oriented Analysis and Design*, *Unified Modeling Language (UML)* (*Activity Diagram*, *Use Case Diagram*, *Use Case Description*, *System Sequence Diagram*, *First-Cut Sequence Diagram*, *Multi Layer Sequence Diagram*, *Domain Class Diagram*, *First-Cut Class Diagram*, *User Interface*), *Axure*, *Visual Paradigm*, *Android Studio*, dan *React-Native*.

2.1 Teori Umum

Pada bagian ini membahas mengenai teori umum yang akan digunakan dalam perancangan *e-Learning* pada Rumah Sakit PELNI.

2.1.1 Sistem

Pengertian sistem menurut (Romney & Steinbart, 2015) sistem adalah sebuah rangkaian dari dua atau lebih komponen-komponen yang saling berhubungan, yang saling berinteraksi untuk mencapai suatu tujuan. Sebagian besar sistem terdiri dari subsistem yang lebih kecil yang mendukung sistem yang lebih besar.

Sistem merupakan sebuah set komponen-komponen yang saling terkait, dengan adanya batasan yang jelas, saling bekerja sama untuk mencapai tujuan umum dengan menerima input dan menghasilkan *output* dalam proses transformasi yang terorganisasi. Sistem mempunyai 3 fungsi dasar, yaitu:

a. *Input*

Input melibatkan elemen yang diambil dan dimasukkan kedalam sistem untuk diproses.

b. *Processing*

Processing melibatkan proses transformasi yang mengubah input menjadi *output*.

c. *Output*

Output melibatkan elemen yang telah dihasilkan dari proses transformasi ke tujuan akhir.

2.1.2 Informasi

Pengertian informasi menurut (Romney & Steinbart, 2015), informasi (*information*) adalah data yang telah dikelola dan diproses untuk memberikan arti dan memperbaiki proses pengambilan keputusan. Sebagaimana perannya, pengguna membuat keputusan yang lebih baik sebagai kuantitas dan kualitas dari peningkatan informasi.

Menurut buku (O'Brien & Marakas, 2010), informasi dapat didefinisikan sebagai data yang telah diubah menjadi konteks yang mempunyai arti dan berguna untuk pengguna terakhir, lebih baik melihat data sebagai sumber bahan baku yang diproses menjadi informasi.

2.1.3 Sistem Informasi

Perkembangan sistem informasi saat ini dapat mempengaruhi perkembangan teknologi seperti perkembangan teknologi yang menggunakan media elektronik sebagai salah satu metode untuk melakukan proses belajar mengajar.

Pengertian Sistem Informasi menurut (Krismiaji, 2015), sistem informasi adalah cara-cara yang diorganisasi untuk mengumpulkan, memasukkan, dan mengolah serta menyimpan data, dan cara-cara yang diorganisasi untuk menyimpan, mengelola, mengendalikan, dan melaporkan informasi sedemikian rupa sehingga sebuah organisasi dapat mencapai tujuan yang telah ditetapkan.

Menurut (O'Brien & Marakas, 2010), suatu sistem informasi bisa berupa kombinasi yang terorganisir dari orang, *hardware*, *software*, jaringan komunikasi, data, kebijakan dan adanya prosedur untuk menyimpan, mengambil, mengubah, dan menyebarluaskan informasi dalam suatu organisasi.

Jadi berdasarkan pengertian diatas bisa disimpulkan bahwa sistem informasi adalah kumpulan data yang terintegritasi dan saling melengkapi serta menghasilkan output yang baik untuk memecahkan masalah dan pengambilan keputusan.

2.1.4 Internet

Internet merupakan teknologi yang menyimpan segudang fasilitas dan layanan yang patut dipahami dan dikuasai oleh siapa pun pada zaman modern saat ini. Namun internet bagaikan hutan rimba, bagi penjelajah yang belum berpengalaman tentu membutuhkan peta dan pemahaman baik konsep maupun teknis aksesnya agar tidak tersesat dan dapat menikmati kegiatan penjelajahan.

Ada beberapa kelebihan internet yaitu salah satunya mampu mempersingkat jarak dan waktu, karena sifatnya yang real time, artinya pada saat konten tertentu dimasukkan dalam internet, maka *user* (pengguna) lainnya langsung dapat melihat konten tersebut walaupun di tempat yang berbeda waktu bahkan jarak yang sangat jauh. Dengan adanya internet memungkinkan terjadinya komunikasi yang super cepat antara satu pihak dengan pihak lainnya, tanpa mengenal batasan ruang dan waktu.

Internet, di era konvergensi digital membawa pengaruh di berbagai sektor kehidupan, menjadikan dunia tanpa batas, dan men-trigger munculnya berbagai teknologi pendukung, penyedia akses atau penyelenggara baru, hingga produk yang mendorong penggunaan lebih masif (Curran, Fenton, & Freedman, 2016).

Menurut (Sherlyanita & Rakhmawati, 2016), internet merupakan sebuah sistem informasi global yang saling terhubung dengan logika oleh address yang memiliki keunikan secara global yang berbasis pada Internet Protocol (IP), dapat mendukung komunikasi dengan adanya TCP/IP, dan membuatnya dapat diakses dengan baik secara umum dan khusus.

Internet dapat diartikan sebagai jaringan komputer yang sangat luas dan mendunia yang menghubungkan pemakai komputer dari satu negara ke negara lain yang ada di seluruh dunia, dimana didalamnya mempunyai berbagai sumber informasi dan fasilitas layanan internet (Hastanti, Purnama, & Wardati, 2015).

Menurut (Hill, 2014), internet adalah pola pikir untuk berkomunikasi yang bebas dan terbuka, melalui protokol, layanan, dan aplikasi yang luas dan sekumpulan jaringan yang saling terhubung.

2.1.5 Data

Pada perkembangan teknologi di zaman sekarang ini data sangatlah diperlukan dalam mendukung pencarian informasi untuk suatu sistem yang akan dibangun.

Menurut (O'Brien & Marakas, 2010), data adalah fakta-fakta mentah atau observasi, biasanya tentang fenomena fisik atau transaksi bisnis. Data dapat berupa dalam berbagai bentuk seperti sensus, statistik, hasil penelitian, angka, grafik, transkrip wawancara, jurnal dan lain-lain (Oyelude, 2017).

2.1.6 Teknologi

Seiring berjalannya waktu, kemajuan teknologi terus berkembang dengan pesat dari zaman ke zaman. Tidak dapat dipungkiri bahwa teknologi merupakan salah satu hal yang tidak bisa lepas dari manusia dalam menjalankan aktivitas sehari-hari.

Menurut (Darimi, 2017), teknologi merupakan sesuatu yang mengacu pada objek benda yang dipergunakan untuk kemudahan aktivitas manusia, seperti mesin, perkakas, atau perangkat keras.

Menurut (Andri, 2017), kata teknologi seringkali diartikan oleh masyarakat sebagai alat elektronik. Tapi oleh ilmuwan dan ahli filsafat ilmu pengetahuan diartikan sebagai pekerjaan ilmu pengetahuan untuk memecahkan masalah praktis. Jadi teknologi lebih mengacu pada usaha untuk memecahkan masalah manusia.

2.2 Teori Khusus

Pada subbab ini akan dibahas mengenai teori khusus yang akan digunakan dalam pembangunan aplikasi *e-Learning* pada Rumah Sakit PELNI.

2.2.1 *E-learning*

Seiring dengan kemajuan teknologi informasi yang semakin pesat, semua hal bisa dilakukan dengan mudah dan efisien, salah satu teknologi yang memudahkan dan praktis untuk digunakan tidak hanya di sekolah maupun universitas namun sudah merambah ke perusahaan-perusahaan, sebagai media melakukan proses pembelajaran atau sering dikenal dengan pelatihan pada saat ini yaitu *e-Learning*.

Kemajuan teknologi dapat menimbulkan kebutuhan akan konsep belajar mengajar berbasis media online menjadi tidak dapat dihindari terutama dari segi pendidikan. Konsep yang kemudian disebut dengan *e-Learning* ini membawa dampak terjadinya proses perubahan atau transformasi pendidikan dalam bentuk konvensional kedalam bentuk digital (*online*), baik secara isi (*content*) dan sistemnya (Suharyanto & Mailangkay, 2016).

Dalam sebuah jurnal menurut (Karmakar & Nath, 2014) *e-Learning* dapat mengubah model pembelajaran pada saat ini seperti untuk murid, mahasiswa, karyawan dan untuk pelatihan yang berkelanjutan serta pengembangan profesional seperti dokter, perawat dan guru tentang bagaimana seseorang tersebut menemukan sesuatu yang bersifat formal maupun non-formal. Sehingga pengetahuan akan diperoleh secara terupdate dan sesuai dengan perkembangan zaman.

Pengembangan dan implementasi *e-Learning* sudah banyak dilakukan di berbagai lembaga, dengan tujuan pembelajaran maupun pelatihan. Untuk itu *e-Learning* mempunyai kelebihan, antara lain:

1. Dapat diakses dimana saja dan bersifat global, yang mampu mengatasi permasalahan waktu dan tempat sehingga seseorang yang mengakses *e-Learning* dapat melakukan pembelajaran dengan mudah kapan pun dan dimana saja dengan menggunakan komputer maupun smartphone yang terhubung jaringan internet.

2. Pengalaman belajar berbeda sehingga pengguna dapat mengatasi kebosanan dan kejenuhan dalam belajar dengan metode biasa.
3. Mendorong kesadaran pengguna untuk belajar secara mandiri.

Menurut (Ghirardini, 2011), kegiatan *e-Learning* bisa terdiri dari *Synchronous* dan *Asynchronous*. *Synchronous* terjadi dalam waktu yang *real/realtime*, pengguna diminta untuk hadir dalam waktu yang bersamaan. Sedangkan *Asynchronous* terjadi dengan waktu yang independen, belajar mandiri adalah contoh dari *asynchronous e-Learning* karena belajar online dapat dilakukan setiap saat.

Table 2.1 *Synchronous dan Asynchronous*

Synchronous	Asynchronous
Chat and IM	email
Telekonferen	Discussion forum
Live webcasting	wiki
Aplication sharing	blog

2.2.2 *Web Based*

Pada era teknologi yang berkembang saat ini, *website* mulai dikenal sebagai halaman yang menyediakan informasi melalui jalur internet sehingga dapat diakses di berbagai belahan dunia selama terkoneksi dengan jaringan internet. *Website* merupakan salah satu cara menampilkan informasi mengenai perusahaan yang dapat dikunjungi tanpa batas ruang dan waktu. Maka dengan adanya *website* sekarang ini akan menjadi strategi pemasaran yang baik dalam bidang bisnis. Penelitian-penelitian terdahulu menjadi bahan acuan dan kajian terhadap penelitian yang dilakukan saat ini. Dengan adanya penelitian terdahulu di dapatlah perbandingan perbandingan yang sesuai dari topik penelitian yaitu mengenai sistem

aplikasi berbasis *website* yang digunakan sebagai media pemasaran *online* dari suatu perusahaan.

Menurut (Wasim, Sharman, Khan, & Siddiqui, 2014), *web based learning* adalah pembelajaran *online* atau *e-Learning* karena termasuk konten kursus *online*. Forum diskusi melalui *email*, konferensi video, dan ceramah langsung (*video streaming*) semua dimungkinkan melalui *web*. *Web Based* juga dapat menyediakan tampilan halaman statis seperti materi kursus.

Menurut (Mannade & Hazare, 2017) *web-based learning* dapat berupa, *learning management systems*, *learning content management systems delivers a life-long education*, *training applications* dan layanan pembelajaran *online* kepada siapa saja, kapan saja, dimana saja dan *real-time*, yang menghasilkan munculnya sistem *e-Learning*. Digital teknologi terus mempengaruhi cara kita mencari, membuat, berbagi, berdiskusi dan ide, bahkan mempengaruhi cara kita berpikir tentang pengetahuan itu sendiri.

2.2.3 *Mobile*

Pada saat ini, kebanyakan orang lebih memilih perangkat *mobile* seperti *handphone* atau *smartphone* untuk mencari informasi ketimbang menggunakan perangkat atau *device* seperti komputer atau laptop yang sulit diakses kapan saja dan dapat dibawa kemana saja.

Menurut (Link, 2018), perangkat *mobile* mempunyai beragam sistem operasi seperti (*iOS*, *Android*, dll.) yang dapat mendorong beberapa jenis fitur pada *device* dan bagaimana fitur-fitur ini dapat diakses dan ditampilkan.

Menurut (Sönmez, Göçmez, Uygun, & Ataizi, 2018), *mobile learning* dapat didefinisikan sebagai wadah yang memberi informasi umum secara elektronik kepada pengguna, yang dapat menyediakan konten pendidikan yang membantu mendapatkan pengetahuan tanpa mempersoalkan waktu dan tempat. Aspek yang membedakan adalah *mobile learning* berlaku untuk alat teknologi portabel kecil (seperti *smartphone*) sementara *e-Learning* dapat digunakan untuk semua pembelajaran dan teknologi pengajaran, termasuk *mobile learning*. Berikut adalah kelebihan dari perangkat *mobile*, yaitu:

- Mempunyai sifat yang fleksibel sehingga memudahkan pengguna dalam mengakses kapanpun dan dimanapun.
- Ukuran perangkatnya yang lebih kecil dan mudah untuk dibawa kemana-mana.
- Kebanyakan perangkat *mobile* mempunyai harga yang lebih murah jika dibandingkan dengan PC *desktop*.
- Dapat mencakup banyak *user* sebab perangkat *mobile* telah menyediakan teknologi yang dapat digunakan sehari-hari.

Berikut adalah kekurangan perangkat *mobile*, yaitu:

- Beberapa perangkat *mobile* yang sulit untuk digunakan karena memiliki tombol keyboard yang berukuran kecil.
- Daya baterai yang mempunyai kapasitas terbatas, yang mengharuskan pengguna perangkat *mobile* harus mengisi ulang baterai.

2.3 Teori Metodologi

Metode yang digunakan dalam penyusunan skripsi ini meliputi 3 hal, yaitu:

2.3.1 Metode Pengumpulan Data

Dalam pengambilan atau pengumpulan data untuk metode ini membutuhkan 4 kegiatan antara lain:

2.3.1.1 Studi Pustaka

Menurut Danang Sunyoto (Sunyoto, 2013), studi pustaka merupakan salah satu metode pengumpulan data dengan memahami buku-buku yang ada mendukung obyek penelitian atau sumber-sumber lain yang memahami penelitian. Dimaksudkan untuk mendapat landasan teoritis dan acuan yang menjadi sumber data guna mendukung penelitian dalam perancangan sistem yang diusulkan.

2.3.1.2 Examining Documentation

Menurut (List, Alexander, & Stephens, 2016) *examining documentation* merupakan pengumpulan data yang dapat memberikan informasi dari dokumen-dokumen yang memberikan kontekstualisasi yang

penting untuk membantu dalam mengevaluasi dan mendapatkan informasi secara lebih detail.

2.3.1.3 Wawancara

Menurut (Sugiyono, 2016), wawancara digunakan sebagai teknik pengumpulan data apabila peneliti ingin melakukan penelitian untuk menentukan permasalahan yang harus diteliti, dan juga apabila peneliti ingin mengetahui informasi dari responden yang lebih mendalam dan jumlah respondennya cukup sedikit. Wawancara dilakukan kepada kepala Divisi Rumah Sakit PELNI. Dimaksudkan Metode ini digunakan untuk mengumpulkan data dengan menanyakan berbagai pertanyaan berdasarkan daftar pertanyaan yang telah dibuat sebelumnya kepada pihak rumah sakit yang hubungannya dengan perancangan dan desain sistem yang diusulkan.

2.3.1.4 Kuesioner

Kuesioner digunakan sebagai alat untuk pengumpulan data menurut pendapat (Sugiyono, 2017), yaitu kuesioner merupakan metode pengumpulan data yang dilakukan dengan cara memberikan beberapa pernyataan tertulis kepada responden untuk di berikan jawabannya. Metode ini digunakan untuk mengumpulkan data dengan menyebarkan kuesioner terkait rancangan aplikasi yang dikembangkan dengan tujuan untuk mengetahui pendapat dari karyawan mengenai rancangan aplikasi.

Pendekatan kuisioner yang digunakan adalah menggunakan *skala likert*. Menurut (Sugiyono, 2017), “merupakan metode yang digunakan untuk mengembangkan instrumen yang digunakan untuk mengukur sikap, persepsi, dan pendapat seseorang atau sekelompok orang terhadap potensi dan permasalahan suatu objek, rancangan suatu produk, proses membuat produk dan produk yang telah dikembangkan atau diciptakan”.

Perhitungan yang dilakukan untuk pengolahan sampel yang digunakan pada penelitian ini ialah rumus *Slovin*. Rumus *slovin* merupakan salah satu teknik pengolahan sampel yang berupa rumus atau formula untuk

menghitung jumlah sampel minimal apabila perilaku dari sebuah populasi tidak diketahui secara pasti dengan estimasi error yang dapat diterima 10% atau persen kelonggaran karena adanya pengambilan sampel yang dapat ditolerir atau diinginkan. Rumus yang digunakan:

$$n = \frac{N}{1 + N e^2}$$

Penjelasan:

n = jumlah sampel

N = ukuran populasi

e = error sampel / estimasi error

2.3.2 Teori Analisis SWOT

Menurut (Emet, 2017) Analisis SWOT adalah alat atau tools yang digunakan untuk perencanaan strategis dan manajemen strategis dalam suatu perusahaan atau organisasi.

Analisis SWOT memiliki empat komponen, yaitu *strength*, *weakness*, *opportunities*, dan *threats*. *Strength* dan *weakness* merupakan faktor internal sedangkan *opportunities* dan *threats* merupakan faktor eksternal. Berikut ini penjelasan dari masing-masing komponen yang ada pada Analisis SWOT:

- *Strength*
Sesuatu yang lebih menguntungkan bila dibandingkan sesuatu yang lain. Dalam pengertian ini, kekuatan mengacu pada karakteristik positif, menguntungkan dan kreatif.
- *Weakness*
Sesuatu yang lebih tidak menguntungkan jika dibandingkan dengan sesuatu yang lain. Dalam hal ini, kelemahan adalah karakteristik yang negatif dan tidak menguntungkan.
- *Opportunities*

Keuntungan dan kekuatan pendorong untuk suatu kegiatan yang terjadi. Untuk alasan ini, itu memiliki karakteristik positif dan menguntungkan.

- *Threats*

Situasi atau kondisi yang membahayakan aktualisasi suatu kegiatan. Ini merujuk pada situasi yang tidak menguntungkan. Untuk alasan ini, ia memiliki karakteristik negatif itu harus dihindari.

2.3.2.1 SWOT Matriks

Menurut (David & David, 2017), SWOT Matriks adalah sebuah alat untuk membantu para manajer dalam mengembangkan empat jenis strategi yaitu: Strategi SO (*strength-opportunities*), Strategi WO (*weakness-opportunities*), Strategi ST (*strength-threats*), dan Strategi WT (*weakness-threats*).

Strategi SO (*strength-opportunities*) merupakan pemanfaatan kekuatan internal yang dimiliki perusahaan untuk menarik keuntungan dari peluang eksternal. Semua perusahaan tentunya menginginkan posisi dimana kekuatan internal dapat digunakan untuk mengambil keuntungan dari berbagai tren diluar.

Strategi WO (*weakness-opportunities*) bertujuan untuk memperbaiki kelemahan internal yang dimiliki perusahaan dengan cara mengambil keuntungan dari peluang eksternal.

Strategi ST (*strength-threats*) memanfaatkan kekuatan internal yang dimiliki perusahaan untuk menghindari atau mengurangi dampak dari ancaman eksternal.

Strategi WT (*weakness-threats*) merupakan strategi pertahanan bagi perusahaan untuk mengurangi kelemahan internal serta menghindari dampak dari ancaman eksternal. Perusahaan yang menghadapi kelemahan internal dan ancaman eksternal benar-benar dalam posisi yang membahayakan.

2.3.3 Teori Perancangan e-Learning

Menurut (Ghirardini, 2011) pada sub-bab ini dalam perancangan *e-Learning* ini menggunakan metode ADDIE yang terdapat 5 *stage*, antara lain:

a. *Analyst*

Menganalisa kebutuhan yang harus dilakukan pada awal upaya pengembangan untuk menentukan apakah pelatihan diperlukan untuk mengisi ilmu pengetahuan dan keterampilan dan apakah *e-Learning* adalah solusi yang terbaik untuk memberikan pelatihan.

Analisis *audiens target* adalah langkah penting lainnya. Desain dan penyampaian *e-Learning* dipengaruhi juga karakteristik utama peserta.

Analisis materi serta mengidentifikasi pekerjaan yang harus dipelajari atau ditingkatkan oleh para peserta. Jenis analisis ini digunakan di pembelajaran yang dirancang untuk membangun keterampilan terkait pekerjaan tertentu.

b. *Design*

Tahap desain meliputi kegiatan berikut:

- Merumuskan tujuan untuk pembelajaran yang diperlukan untuk mencapai tujuan.
- Mendefinisikan urutan tujuan yang harus dicapai
- Memilih strategi pengajaran, media, evaluasi dan penyampaian. Hasil dari tahap desain adalah *blueprint* yang akan digunakan sebagai referensi untuk mengembangkan pembelajaran.

c. *Development*

Perancangan konten interaktif multimedia terdiri dari 3 langkah utama:

- Perancangan konten
Menulis atau mengumpulkan semua pengetahuan dan informasi yang diperlukan.
- Perancangan *storyboard*
Mengintegrasikan alur pembelajaran yang disusun secara berurutan.
- Perancangan *courseware*

Mengembangkan komponen media dan interaktif, menghasilkan format yang berbeda dan mengintegrasikan konten ke dalam platform pembelajaran yang dapat diakses peserta.

d. *Implementation*

Pada tahap ini kursus telah disampaikan kepada peserta, tahap ini juga mencakup mengelola dan memfasilitasi kegiatan peserta.

e. *Evaluation*

Proyek *e-Learning* dapat di evaluasi untuk tujuan evaluasi yang spesifik.

2.3.4 Teori Perancangan

Pada sub-bab ini akan dijelaskan mengenai teori-teori perancangan diagram yang digunakan dalam perancangan aplikasi *e-Learning* pada Rumah Sakit PELNI.

2.3.4.1 *Object-Oriented Analysis and Design*

Dalam sebuah jurnal menurut (Joo & Woo, 2014), *Object-Oriented Analysis and Design* tersedia secara aman dalam siklus pengembangan sistem mulai dari *requirement*, analisis sampai langkah implementasi. *Nonfunctional requirement* digunakan pada fase *requirement* dan analisis sedangkan pada tahap desain menggunakan UML.

Menurut (Satzinger, Jackson, & Burd, 2016), *Object-Oriented Analysis and Design* itu adalah proses untuk mengidentifikasi dan mendefinisikan dari setiap kelas, metode, dan setiap pesan yang diperlukan untuk mengeksekusi dari *use case*.

2.3.4.2 *Unified Modeling Language*

Dalam suatu jurnal menurut (Zheng, Feng, & Zhao, 2014), UML (*Unified Modelling Language*) merupakan suatu standar, berorientasi objek, pemodelan sistem yang digunakan untuk menggambarkan model sebuah sistem yang akan dibangun. UML juga merupakan suatu bahasa grafik dalam visualisasi dan penggambaran suatu sistem dengan

mengkombinasikan metodologi pengembangan berorientasi objek, teknologi dan UML berorientasi objek.

Menurut (Satzinger, Jackson, & Burd, 2016), *Unified Modelling Language* sebuah standar set dari konstruk model informasi dan mempunyai notasi yang ditentukan oleh *Object Management Group*. Berikut merupakan bagian dari UML, antara lain:

- *Activity Diagram*

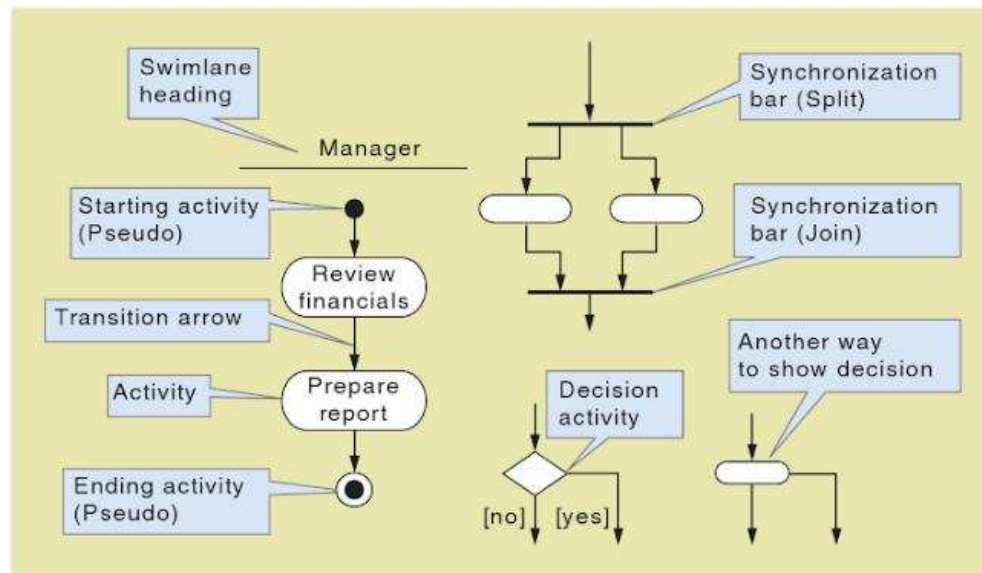
Didalam pengembangan sistem informasi, pemodelan proses bisnis sangat diperlukan dalam mendeskripsikan proses bisnis yang sedang berjalan maupun yang akan dilakukan pengembangan. Salah satu pemodelan yang biasa digunakan adalah *activity diagram*.

Menurut (Shelly & Rosenblatt, 2012), UML *activity diagram* menyerupai *flowchart* horizontal yang menunjukkan tindakan dan peristiwa pada saat terjadi.

Menurut (Dennis, Wixom, & Roth, 2012), *activity diagram* digunakan untuk semua jenis pemodelan aktivitas yang digambarkan dalam konteks pemodelan proses bisnis. Artinya menggambarkan proses atau aktivitas yang dilakukan dan bagaimana perpindahan objek ataupun data.

Menurut (Satzinger, Jackson, & Burd, 2016), *activity diagram* adalah diagram UML yang menggambarkan aktivitas pengguna (atau sistem), orang atau komponen yang menyelesaikan setiap aktivitas, dan urutan seluruh aktivitas yang dilakukan.

Elemen-elemen *activity diagram* dapat dilihat pada gambar berikut:



Gambar 2.1 Activity Diagram Element

Sumber: (Satzinger, Jackson, & Burd, 2016)

Keterangan dari Gambar *Activity Diagram Element* diatas:

1. *Swimlane heading* adalah sebuah kolom aktivitas diagram yang berisi semua kegiatan untuk agen tunggal atau unit organisasi.
2. *Synchronization bar* adalah sebuah komponen aktivitas diagram yang memisahkan jalur kontrol menjadi beberapa jalur konkuren (*split*) atau menggabungkan jalur konkuren (*join*).
3. *Starting activity* adalah sebuah komponen aktivitas diagram yang memberikan tanda awal dari dimulainya sebuah aktivitas diagram.
4. *Transition arrow* adalah sebuah komponen aktivitas diagram yang menandakan akan alur *workflow* yang terjadi pada aktivitas diagram.
5. *Activity* adalah sebuah komponen aktivitas diagram yang berisi tentang kegiatan yang berlangsung pada sebuah aktivitas diagram.

6. *Ending activity* adalah sebuah komponen aktivitas diagram yang menjadi penanda akan berakhirnya sebuah aktivitas diagram.
7. *Decison activity* adalah sebuah komponen aktivitas diagram yang digunakan ketika adanya aktivitas yang memiliki pilihan pada sebuah aktivitas diagram.

- *Use Case Diagram*

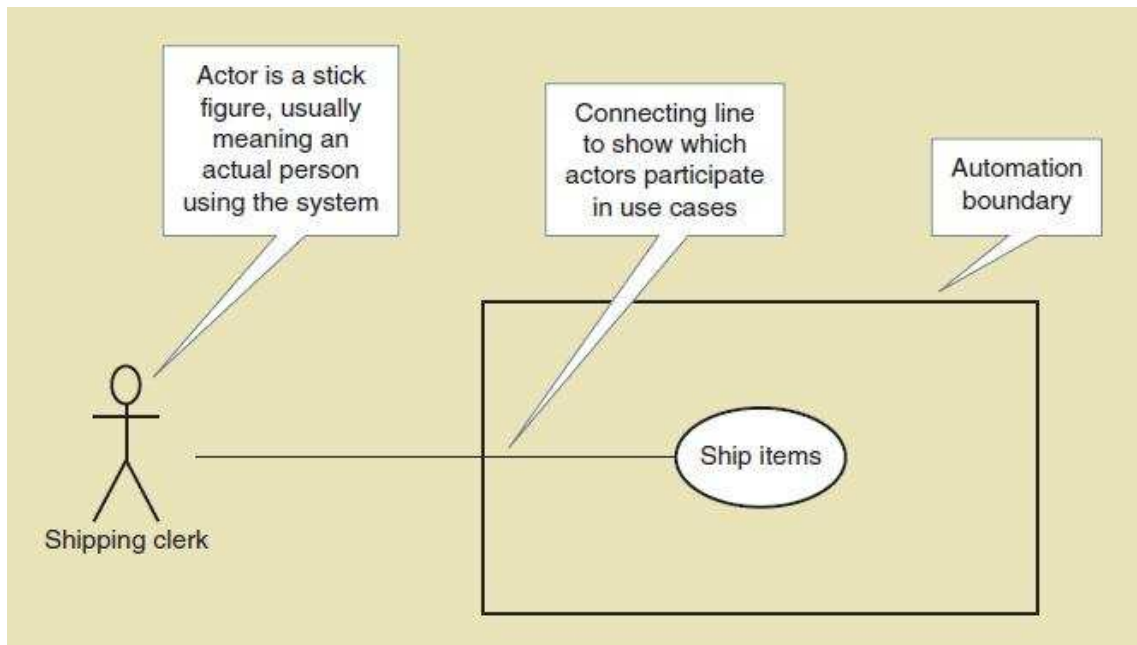
Didalam perancangan sistem informasi, diperlukan pemodelan bagaimana pengguna atau user berinteraksi dengan sistem yang akan dibangun. Salah satu pemodelan yang bisa digunakan adalah *use case diagram*.

Menurut (Shelly & Rosenblatt, 2012), *use case diagram* adalah representasi visual yang mewakili interaksi antara pengguna dan sistem informasi dalam UML.

Menurut (Dennis, Wixom, & Roth, 2012), *use case diagram* merupakan sebuah teknik diagram UML dimana fungsi utama dari sistem dengan jenis pengguna saling berinteraksi.

Menurut (Satzinger, Jackson, & Burd, 2016), *use case diagram* merupakan model UML yang digunakan untuk menggambarkan *use case* dan bagaimana relasinya dengan aktor.

Elemen-elemen *use case diagram* dapat dilihat pada gambar berikut:



Gambar 2.2 Use Case Diagram Element

Sumber: (Satzinger, Jackson, & Burd, 2016)

Keterangan dari Gambar *Use Case Diagram Element* diatas:

1. *Actor* adalah orang yang berinteraksi menggunakan sistem.
2. *Use Case* merupakan bagian utama dari fungsi sistem berbentuk oval dan terdapat nama proses didalamnya.
3. *Connection Line* merupakan garis yang menghubungkan *actor* dan use case.
4. *Boundary* merupakan batasan antara sistem dengan *actor* yang ditampilkan berbentuk persegi panjang yang berisi *use case*.

Jadi, *use case diagram* merupakan suatu jenis pemodelan berupa ringkasan visual yang menampilkan bagaimana interaksi antara *actor* dengan sistem.

- *Use Case Description*

Pada saat pemodelan *use case diagram* diperlukan bagaimana detail informasi dari setiap *use case* dibuat dalam bentuk *use case description*.

Menurut (Dennis, Wixom, & Roth, 2012), *use case description* merupakan sebuah sarana untuk mendokumentasikan setiap penggunaan *use case*. *Use case description* berisi semua informasi yang dibutuhkan untuk mendokumentasikan fungsi dari suatu proses bisnis.

Menurut (Satzinger, Jackson, & Burd, 2016), *use case description* adalah sebuah pemodelan yang dibuat secara tekstual mendaftarkan dan menggambarkan detail proses dari sebuah *use case*. Semua detail informasi untuk semua *use case* dapat dideskripsikan pada *use case description*.

Elemen-elemen *use case description* terdiri dari beberapa bagian seperti *use case name*, *scenario*, *triggering event*, dan beberapa komponen lainnya yang dapat dilihat pada gambar berikut:

Use case name:	Create customer account.									
Scenario:	Create online customer account.									
Triggering event:	New customer wants to set up account online.									
Brief description:	Online customer creates customer account by entering basic information and then following up with one or more addresses and a credit or debit card.									
Actors:	Customer.									
Related use cases:	Might be invoked by the <i>Check out shopping cart</i> use case.									
Stakeholders:	Accounting, Marketing, Sales.									
Preconditions:	Customer account subsystem must be available. Credit/debit authorization services must be available.									
Postconditions:	Customer must be created and saved. One or more Addresses must be created and saved. Credit/debit card information must be validated. Account must be created and saved. Address and Account must be associated with Customer.									
Flow of activities:	<table><tr><th>Actor</th><th>System</th></tr><tr><td>1. Customer indicates desire to create customer account and enters basic customer information.</td><td>1.1 System creates a new customer. 1.2 System prompts for customer addresses.</td></tr><tr><td>2. Customer enters one or more addresses.</td><td>2.1 System creates addresses. 2.2 System prompts for credit/debit card.</td></tr><tr><td>3. Customer enters credit/debit card information.</td><td>3.1 System creates account. 3.2 System verifies authorization for credit/debit card. 3.3 System associates customer, address, and account. 3.4 System returns valid customer account details.</td></tr></table>	Actor	System	1. Customer indicates desire to create customer account and enters basic customer information.	1.1 System creates a new customer. 1.2 System prompts for customer addresses.	2. Customer enters one or more addresses.	2.1 System creates addresses. 2.2 System prompts for credit/debit card.	3. Customer enters credit/debit card information.	3.1 System creates account. 3.2 System verifies authorization for credit/debit card. 3.3 System associates customer, address, and account. 3.4 System returns valid customer account details.	
Actor	System									
1. Customer indicates desire to create customer account and enters basic customer information.	1.1 System creates a new customer. 1.2 System prompts for customer addresses.									
2. Customer enters one or more addresses.	2.1 System creates addresses. 2.2 System prompts for credit/debit card.									
3. Customer enters credit/debit card information.	3.1 System creates account. 3.2 System verifies authorization for credit/debit card. 3.3 System associates customer, address, and account. 3.4 System returns valid customer account details.									
Exception conditions:	1.1 Basic customer data are incomplete. 2.1 The address isn't valid. 3.2 Credit/debit information isn't valid.									

Gambar 2.3 Use Case Description Element

Sumber: (Satzinger, Jackson, & Burd, 2016)

Keterangan dari Gambar *Use Case Description Element* diatas:

1. *Use case name* adalah nama fungsi atau *use case* dalam sistem.
2. *Scenario* adalah sekumpulan kegiatan unik dalam suatu *use case*.
3. *Triggering event* adalah bagian yang mengidentifikasi kejadian yang memicu *use case*.
4. *Brief descriptions* merupakan deskripsi singkat dari *use case*.
5. *Actors* adalah orang terlibat ke dalam *use case*.
6. *Related use case* adalah *use case* yang berhubungan dengan *use case* sekarang.
7. *Stakeholders* adalah bagian yang berkepentingan selain *actor* tertentu.
8. *Preconditions* adalah sebuah kondisi yang harus benar sebelum *use case* dimulai.

9. *Postconditions* adalah apa yang harus benar setelah berhasil menyelesaikan *use case*.
10. *Flow of activities* adalah bagian yang menjelaskan detail jalannya aktivitas dari *use case*.
11. *Exception Conditions* adalah alternative aktivitas atau kondisi pengecualian dari suatu *use case*.

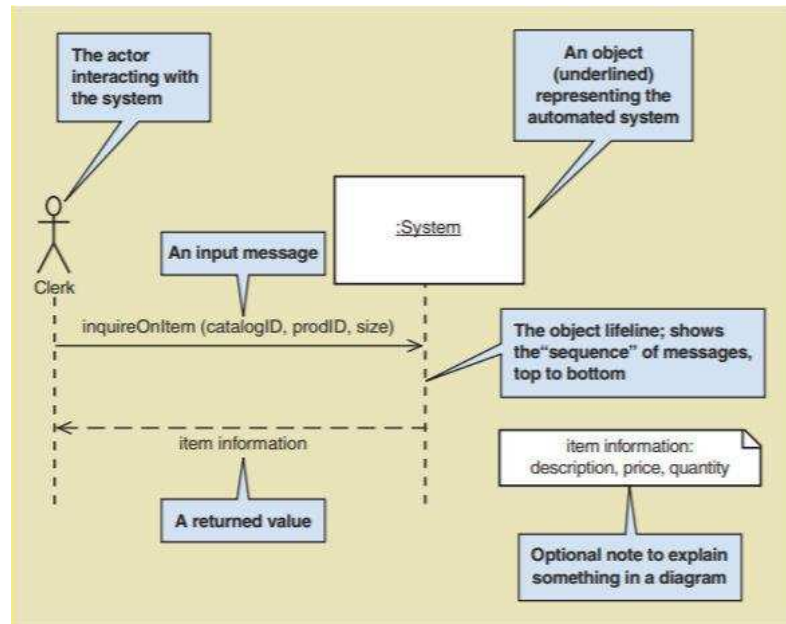
Jadi *use case description* menjelaskan rincian informasi dari setiap *use case* yang didokumentasikan sesuai fungsi dari proses bisnis.

- *System Sequence Diagram*

Pada saat perancangan sistem informasi, diperlukan gambaran interaksi antar objek dalam sistem. Salah satu permodelan yang bisa digunakan adalah *system sequence diagram*.

Menurut (Satzinger, Jackson, & Burd, 2016), *system sequence diagram* adalah sebuah diagram yang menunjukkan aliran pesan diantara seorang aktor dengan sistem dalam sebuah *use case* atau skenario.

Pada *system sequence diagram* diperlihatkan bagaimana aktor berinteraksi dengan sistem dengan memasukkan data (*input*) dan apa yang menjadi data keluaran (*output*) yang dihasilkan sebagai reaksi dari sistem terhadap data masukan tersebut.



Gambar 2.4 System Sequence Diagram Element

Sumber: (Satzinger, Jackson, & Burd, 2016)

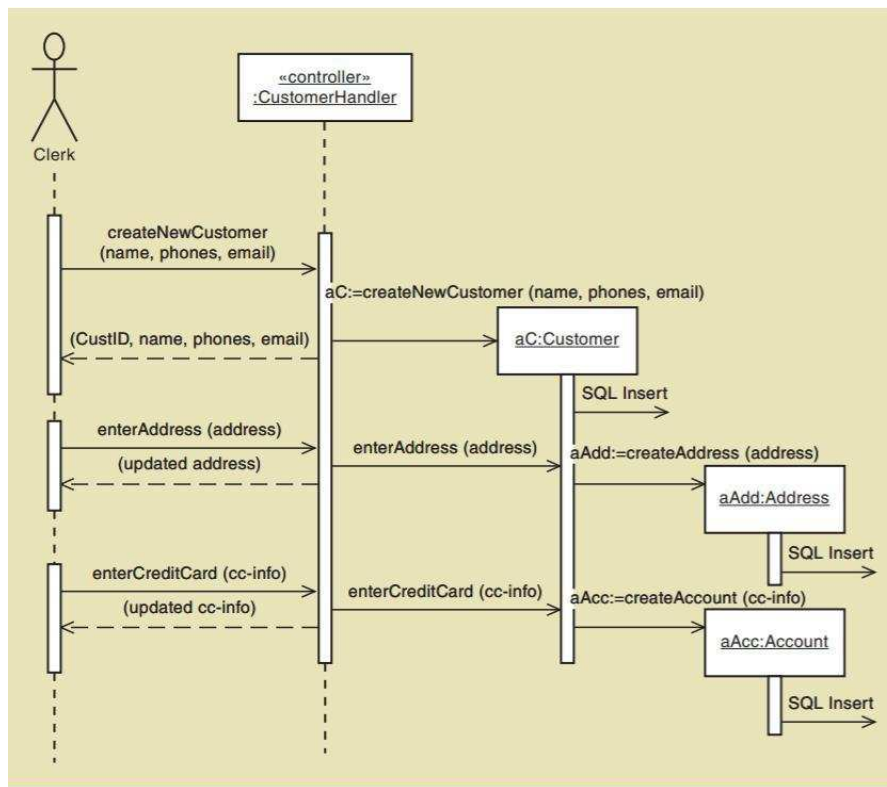
Keterangan dari Gambar *System Sequence Diagram Element* diatas:

- Actor*: Aktor-aktor yang berinteraksi dengan sistem.
- Object system*: sebuah *object* yang mempresentasikan sistem secara otomatis.
- Input message*: sebuah input data yang dilakukan oleh aktor yang bisa disertakan juga oleh parameter kosong ataupun parameter yang berisi data.
- The object lifeline*: sebuah garis putus-putus yang berada dibawah *object system* yang menggambarkan *message* dari atas hingga bawah dari suatu *sequence*.
- Returned value*: merupakan sebuah respon dari sistem yang merupakan *output* dari *input* yang dilakukan oleh aktor.
- Optional note*: untuk menjelaskan sesuatu yang ada pada diagram *ssd* yang telah kita buat.

- *Sequence Diagram (First-Cut)*

Pada saat perancangan sistem informasi, selain membuat system sequence diagram untuk memperjelas alur yang ada pada saat perancangan perlu dilakukan pembuatan sequence diagram (*First-Cut*), yang dimana *sequence diagram (First-Cut)* merupakan gambaran lebih jelas dari suatu proses yang akan di rancang.

Sama halnya dengan use case diagram, menurut (Satzinger, Jackson, & Burd, 2016), *sequence diagram* adalah jenis diagram interaksi yang menekankan urutan pesan yang dikirim antar objek untuk *use case* yang lebih spesifik.



Gambar 2.5 System Sequence Diagram (First-Cut) Element

Sumber: (Satzinger, Jackson, & Burd, 2016)

Keterangan dari Gambar *System Sequence Diagram (First Cut) Element* diatas:

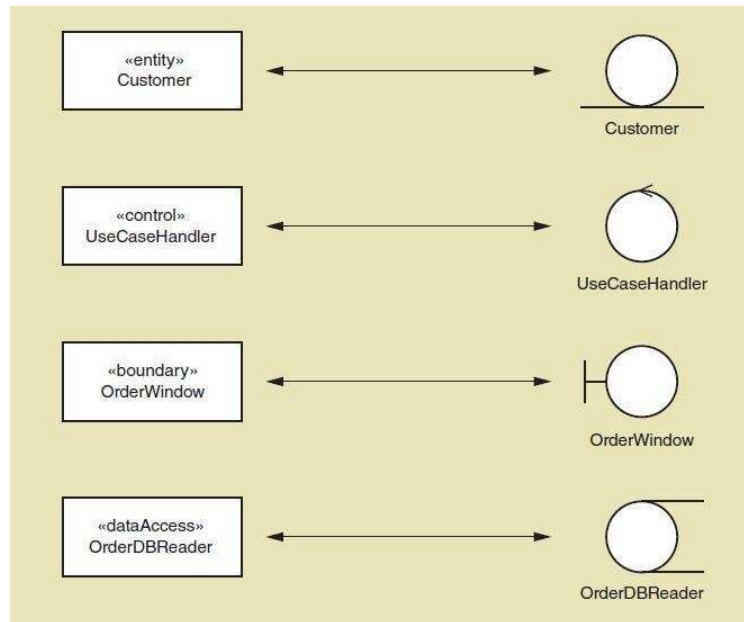
1. *Actor*: merupakan orang atau user yang berinteraksi dengan sistem.
2. *Object*: sesuatu yang ada didalam sistem informasi yang melakukan yang melakukan respon dengan mengeksekusi function atau method.
3. *Lifeline*: menunjukkan urutan pesan dari objek.
4. *Input message*: perintah yang dikirimkan oleh user kedalam sistem.
5. *Output message/ return message*: respon yang diberikan oleh sistem kepada objek berdasarkan perintah yang dilakukan oleh user.
6. *Optional note*: digunakan untuk menampilkan note tambahan.
7. *Controller handler*: sebuah objek yang akan menerima input messages, mencari objek yang berhubungan dan melanjutkan pesan kepada objek yang telah ditentukan.

- *Multi Layer Sequence Diagram*

Pada saat perancangan perancangan sistem infomasi, *sequence diagram* akan ada dilakukan pengembangan, setelah *sequence diagram (First Cut)* dibuat kemudian akan dilakukan lagi pembuatan *multi layer sequence diagram* yang dimana *multi layer sequence diagram* akan menjelaskan alur yang lebih detail dimana penggabungan *sequence diagram (First Cut)* hingga *multi layer sequence diagram*.

Menurut (Satzinger, Jackson, & Burd, 2016), *system sequence diagram* diperlukan dalam proses pemodelan yang menunjukkan aliran pesan antara aktor dengan sistem, namun untuk memperjelas interaksi tersebut adalah sebuah *multi layer system sequence diagram* yang akan menggambarkan interaksi tiga layer dalam sebuah sistem.

Ketiga layer tersebut adalah *view layer*, yang memiliki tanggung jawab utama dalam mengatur dan menampilkan informasi ke layar. *Business logic layer*, berupa objek dan layer ketiga yaitu *data access layer* yang menghubungkan objek ke database, membaca informasi data dan mengirimkannya kembali ke objek.



Gambar 2.6 Multilayer Sequence Diagram Element

Sumber: (Satzinger, Jackson, & Burd, 2016)

Keterangan dari Gambar Multilayer Sequence Diagram Element diatas:

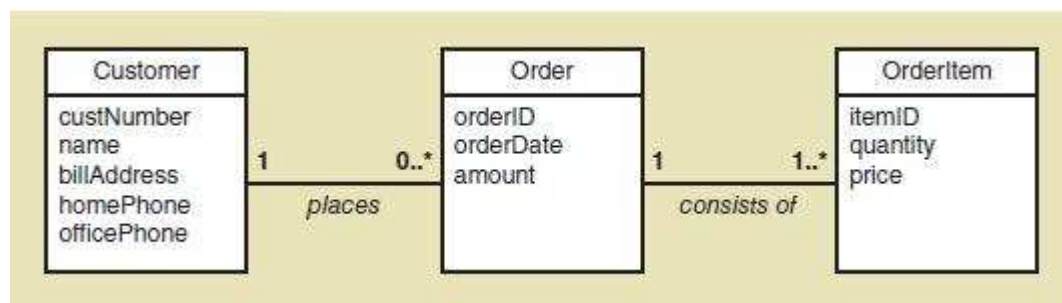
1. *Entity class*: suatu *class* yang dibutuhkan dalam proses identifikasi atau lanjutan desain sistem pada domain class.
2. *Boundary class*: *user interface* pada sistem *desktop* yang menghantarkan *message* ke *controller*.
3. *Control class*: merespon *message* atau pesan dari *boundary classes* dan mengirimkannya ke objek *entity classes* yang benar.
4. *Data access class*: tempat penyimpanan akhir atau *database* dari proses yang berjalan. Semua data pada proses yang berjalan akan

tersimpan pada *data access* dan dibutuhkan pada tiap akhir suatu proses.

- *Domain Class Diagram*

Dalam perancangan *class diagram* terkadang perancang tidak mengetahui proses untuk memulai menggambarkan kelas sehingga tidak sesuai dengan *domain*. Maka diperlukan *domain class diagram* untuk membantu perancangan *class diagram*.

Menurut (Satzinger, Jackson, & Burd, 2016), *domain class diagram* adalah sebuah *class diagram* yang mencakup kelas-kelas dari domain yang memiliki permasalahan.



Gambar 2.7 Domain Class Diagram Element

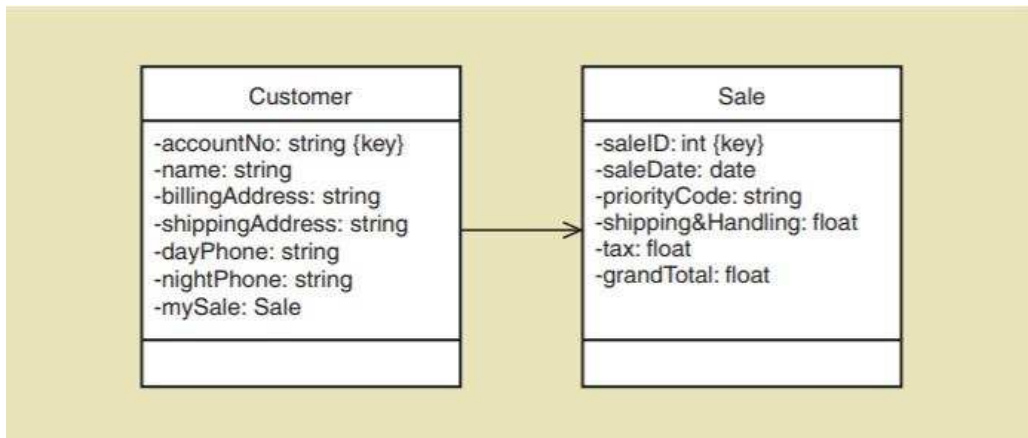
Sumber : (Satzinger, Jackson, & Burd, 2016)

Keterangan dari Gambar *Domain Class Diagram Element* diatas:

1. *Name Class*: Nama dari bagian sebuah *class* yang saling berhubungan dan memiliki relasi.
2. *Attribute Class*: Semua object didalam *class* yang memiliki sebuah nilai untuk setiap bagiannya.
3. Asosiasi, yaitu hubungan statis antar *class*. Umumnya menggambarkan *class* yang memiliki *attribute* berupa *class* lain, atau *class* yang harus mengetahui eksistensi *class* lain.

- *First-Cut Class Diagram*

Setelah domain class diagram, dibutuhkan juga *First-Cut Class Diagram* untuk memperjelas mengenai alur data beserta tipe datanya didalam sebuah perancangan. Dalam penggambaran proses perancangan sebuah design kita dapat memulai merancang *First-Cut Class Diagram* berdasarkan domain *Class Diagram*.



Gambar 2.8 *First-Cut Class Diagram Element*

Sumber : (Satzinger, Jackson, & Burd, 2016)

Keterangan dari Gambar *First-Cut Class Diagram* diatas:

1. *Name Class*: Nama dari bagian sebuah *class* yang saling berhubungan dan memiliki relasi.
2. *Attribute Class and Type Data*: Semua *objek* didalam *class* yang memiliki sebuah nilai dan tipe data untuk setiap bagiannya.
3. *Flow*: menjelaskan penjelasan navigasi dari objek yang memberikan informasi ke objek yang membutuhkan informasi maupun dari objek *child*-nya ke *parent*-nya pada sebuah hirarki.

- *User Interface*

Pada saat perancangan sistem informasi, semua data perancangan telah dibuat dimulai dari alur skenario hingga skenario proses bisnis

yang akan berjalan, langkah selanjutnya adalah pembuatan *User Interface* atau yang lebih dikenal dengan UI.

Menurut (Satzinger, Jackson, & Burd, 2016), *user interface* atau UI adalah sekumpulan *input* dan *output* yang berinteraksi dengan pengguna atau *user* untuk menjalankan fungsi aplikasi. Jika *user interface* memiliki tampilan yang tidak baik, *user* tidak dapat memperoleh keuntungan dari sistem tersebut, dan sebaliknya jika *user interface* memiliki desain yang baik, seperti navigasi dan informatif akan meningkatkan penggunaan sistem.

Menurut (Shneiderman, 2010) ada beberapa point yang menjelaskan tentang pembuatan suatu rancangan interface berdasarkan “Eight Golden Rules” diantaranya :

1. *Strive for consistency*

Konsistensi antar halaman dalam satu aplikasi sangatlah penting terutama konsistensi dari sisi desain web/sistem, urutan perintah, tindakan, perintah, dan istilah yang digunakan pada *prompt*, menu, serta layar bantuan juga harus diperhatikan. Dengan konsistensi tersebut akan membuat *user* nyaman dalam mengeksplorasi aplikasi tanpa takut menggunakan aplikasi lain.

2. *Cater to universal usability*

Setiap *user* memiliki latar belakang budaya dan bahasa, juga jenis tingkat pemahaman *user* terhadap aplikasi yang berbeda-beda. *User* yang masih baru menggunakan aplikasi, atau *user* yang sudah terbiasa menggunakan aplikasi, tentu memiliki preferensi *interface* yang berbeda. Dengan membuat sebuah *shortcut* akan dapat memaksimalkan interaksi dan memudahkan *user* untuk menggunakan fungsi aplikasi dari halaman satu ke halaman lain.

3. *Offer informative feedback*

Memberikan umpan balik yang informatif atau *feedback*. Ketika *user* melakukan suatu *action*, aplikasi akan memberikan suatu *feedback*. Dengan demikian user paham bahwa *action* sudah direspon oleh aplikasi.

4. *Design dialogs to yield closure*

Pada poin ini sebenarnya termasuk dari bagian *feedback*, dengan merancang sebuah *dialog* untuk menghasilkan suatu penutupan akhir dari suatu proses. *User* paham bahwa dia tidak perlu menunggu apakah masih akan ada tahapan lain setelah menyelesaikan suatu proses.

5. *Prevent errors*

Sebisanya mungkin aplikasi dirancang untuk mencegah agar *user* tidak dapat melakukan kesalahan fatal dalam menjalankan suatu proses. Biasanya berupa petunjuk pengisian formulir sesuai format yang diterima oleh aplikasi, sehingga *user* dapat mengisi formulir dengan tepat pada percobaan pertama tanpa terjadi kesalahan.

6. *Permit easy reversal of actions*

Penggunaan tombol *back* pada aplikasi merupakan suatu hal yang penting. Hal tersebut dapat mengurangi kekuatiran *user*, karena *user* mengetahui kesalahan yang dilakukan dapat dibatalkan. Contoh paling nyata dari poin ini adalah pada aplikasi *online shop*, *user* dapat mencoba melakukan pembelian, tetapi pada saat selesai memilih barang, *user* dapat melakukan pembatalan barang yang ingin dibeli. Dengan demikian *user* merasa nyaman saat mencoba untuk melakukan eksplorasi pada aplikasi, karena barang yang dicoba untuk dibeli tidak langsung terproses beli, tetapi *user* dapat melakukan *cancel* / *delete* barang yang sudah dipilih.

7. *Support internal locus of control*

Bagi *user* yang sudah terbiasa menggunakan aplikasi, biasanya *user* ingin memiliki tampilan yang bisa diatur oleh *user* sendiri sesuai preferensi, *user* bisa disebut sebagai pemegang kendali atau kontrol terhadap suatu rancangan *interface*. Hal ini dapat meningkatkan kepuasan *user* terhadap aplikasi yang sangat mempengaruhi UX terhadap aplikasi tersebut.

8. *Reduce short-term memory load*

Merancang *user interface* sesederhana mungkin baik dari segi tata letak menu dan tombol agar mengurangi beban ingatan jangka pendek, sehingga tidak perlu mengingat terlalu banyak perintah, dan juga dapat menghindari terjadinya kebingungan pada para *user*. Dengan menerapkan poin ini, maka *user* tidak perlu mengingat data yang harus di-input ke sistem. Karena data yang harus di-input, sudah disediakan oleh sistem. Misalnya pada *online shop*, *user* memilih barang yang diinginkan, dan dari situ sistem langsung menerima input kode barang yang diinginkan oleh *user* untuk diproses pada proses pembayaran nantinya.

2.3.4.3 Axure

Axure merupakan aplikasi desktop yang digunakan untuk membuat *wireframe*, *prototype* interaktif, *flow diagrams*, dan dokumentasi untuk aplikasi bisnis, *website* dan aplikasi *mobile*. *Axure* berfokus pada fleksibilitas, menyediakan fitur untuk membuat tampilan visual yang bagus dan membantu *non-coders* (orang yang tidak dapat melakukan *coding*), dan memberikan fitur yang mudah dimengerti oleh user yaitu dengan *drag and drop*. *Axure* juga dapat di export menjadi file HTML agar dapat dilihat melalui web browser. *Axure* dapat dijalankan baik di Windows maupun Mac (*Axure*, 2016).

2.3.4.4 Visual Paradigm

Menurut (Reyes, Garcia, & Mué, 2014), *Visual Paradigm* adalah alat CASE (*Computer Aided Software Engineering*) yang berlaku di seluruh siklus hidup pengembangan perangkat lunak yang mendukung UML, SysML (*Systems Modeling Language*), BPMN (*Business Process Modeling Notation*) di antara teknologi lainnya. Dengan lisensi gratis dan komersial ini memungkinkan anda untuk menggambar semua jenis diagram kelas, menghasilkan kode dari diagram dan menghasilkan dokumentasi. Ini juga menyediakan banyak tutorial, demonstrasi interaktif dan proyek-proyek UML.

2.3.4.5 Android Studio

Android Studio merupakan sebuah *Integrated Development Environment* (IDE) khusus untuk membangun aplikasi yang berjalan pada *platform android*. *Android Studio* ini berbasis pada *IntelliJ IDEA*, sebuah IDE untuk bahasa pemrograman *Java*. Bahasa pemrograman utama yang digunakan adalah *Java*, sedangkan untuk membuat tampilan atau *layout*, digunakan bahasa XML. *Android studio* juga terintegrasi dengan *Android Software Development Kit* (SDK) untuk *deploy* ke perangkat *android*. *Android Studio* juga merupakan pengembangan dari *eclipse*,

dikembangkan menjadi lebih kompleks dan professional yang telah tersedia didalamnya *Android Studio IDE*, *Android SDK tools* (Android Developers, 2018).

2.3.4.6 *React-Native*

React-Native merupakan sebuah *framework* berbasis *JavaScript* untuk membuat aplikasi berbasis *mobile*, baik itu *android* maupun *iOS*. *React-Native* merupakan sekumpulan *library* berbasis *JavaScript* yang dikembangkan oleh *Facebook*. Sintaks *React-Native* merupakan gabungan antara *JavaScript* dan XML yang dapat disebut JSX.

React-Native merupakan *framework* yang dikembangkan oleh *Facebook* pada tahun 2015. *React-Native* dibuat dengan tujuan memudahkan *web developer* untuk membuat aplikasi berbasis *mobile*, baik itu *Android* maupun *iOS*. *React-Native* memiliki kemiripan dengan *React* untuk *web* (*ReactJS*). Dapat disimpulkan bahwa *React-Native* merupakan sebuah *framework* berbasis *JavaScript* yang dapat digunakan untuk membuat atau mengembangkan aplikasi berbasis *android* dan *iOS* (Eisnman, 2015).

