

BAB 2

LANDASAN TEORI

1.1 Teori-Teori Umum

1.1.1 *Pengertian Sistem*

Definisi sistem ditemukan dan banyak variasinya, tergantung pada kebutuhannya. Menurut (Satzinger, Jackson, & Burd, 2010), sistem adalah kumpulan komponen-komponen yang saling berkaitan untuk mencapai beberapa hasil. Menurut (O'Brien & Marakas, 2014), sistem adalah sekumpulan komponen yang saling terkait untuk dapat menghasilkan output yang teroganisir dengan baik. Menurut (Robbins & Coulter, 2012), sistem adalah sekumpulan komponen yang saling terkait dan saling berhubungan yang disusun dengan cara tertentu yang menghasilkan keseluruhan yang utuh. Sedangkan, menurut Yakub (2012:1), Sistem adalah sekelompok elemen-elemen dengan tujuan yang sama untuk mencapai hasil yang sudah di rancang.

1.1.2 *Pengertian Informasi*

Menurut (Rainer & Cegielski, 2011) informasi dapat mengacu pada data yang telah diatur sehingga memiliki makna dan nilai bagi penerima. Pengertian yang sama juga diberikan oleh (O'Brien & Marakas, 2014), Informasi adalah data yang telah di ubah menjadi konteks yang bermakna sehingga dapat berguna untuk pengguna (*user*).

1.1.3 *Pengertian Sistem Informasi*

Menurut (Satzinger, Jackson, & Burd, 2010), Sistem informasi adalah sekumpulan komponen komputer yang saling terhubung, yang berfungsi untuk mengumpulkan, memproses, menyimpan dan menyediakan hasil akhir berupa informasi yang dibutuhkan untuk menyelesaikan sebuah tugas bisnis. Menurut (Rainer & Cegielski, 2011), Sistem Informasi adalah proses untuk mengumpulkan, memproses, menyimpan , menganalisis dan menyebarkan suatu informasi untuk tujuan tertentu.

Menurut (Rainer & Cegielski, 2011), terdapat beberapa komponen sistem informasi yang terdiri dari :

1. Perangkat Keras

Seperti perangkat *processor, monitor, keyboard, dan printer*. Perangkat ini secara bersamaan akan menerima, memproses data dan informasi.

2. Perangkat Lunak

Sebuah kumpulan program yang memungkinkan perangkat keras memproses data.

3. Basis Data

Adalah kumpulan file atau *table* yang terkait data- data yang tersimpan.

4. Jaringan

Adalah sistem penghubung (*wireline/wireless*) yang memungkinkan beberapa komputer yang berbeda untuk berbagi data bersama.

5. Prosedur

Adalah seperangkat intruksi tentang cara menggabungkan komponen secara berurutan untuk memproses informasi dan menghasilkan output yang di inginkan.

6. Orang

Adalah orang menggunakan perangkat keras dan perangkat lunak, atau menggunakan hasil pemrosesan.

1.1.4 *Pengertian Database*

Menurut (Satzinger, Jackson, & Burd, Systems Analysis and Design : In a Changing World, 2016) *Database* adalah kumpulan dari data-data terintegrasi yang tersimpan dan dikendalikan secara terpusat. Sebuah *database* dapat menyimpan informasi kedalam puluhan atau ratusan *class*.

1.1.5 *Pengertian Analisis Sistem*

Menurut (Satzinger, Jackson, & Burd, Systems Analysis and Design : In a Changing World, 2016), analisis sistem merupakan serangkaian kegiatan yang dilakukan oleh seseorang untuk memahami dan menentukan apa saja yang harus bisa dicapai oleh sistem yang baru.

1.1.6 *Pengertian Desain Sistem*

Menurut (Satzinger, Jackson, & Burd, Systems Analysis and Design : In a Changing World, 2016) desain sistem adalah kegiatan yang dilakukan oleh seseorang untuk

mendefinisikan dan mendeskripsikan secara rinci apa saja masalah yang harus bisa diselesaikan oleh sistem yang baru.

1.1.7 *Pengertian Perancangan Sistem*

Menurut (Saragih, Simarmata, & Maslam, 2015), perancangan adalah satu langkah untuk memberikan gambaran secara umum kepada manusia atau pengguna tentang sistem yang diusulkan. Perancangan sistem atau desain secara umum mendefinisikan komponen-komponen sistem informasi pemetaan yang akan dirancang.

1.1.8 *Pengertian Internet*

Menurut (Nugroho & Purnama, 2012), Sebuah jaringan yang menghubungkan komputer di seluruh dunia sehingga terbentuk ruang maya jaringan komputer.

1.1.9 *Pengertian Website*

Menurut (Prayitno & Safitri, 2015), *Website* adalah keseluruhan halaman-halaman web yang terdapat dari sebuah domain yang mengandung informasi.

1.1.10 *Pengertian E-Licensing*

E-Licensing merupakan sistem yang digunakan untuk mengelola informasi yang berkaitan dengan lisensi dan dapat diperoleh secara *on-line*. Proses ini dilakukan secara *on-line* dengan penggunaan sistem komputerisasi dan sangat bertentangan dengan manual tradisional. (Alzahrani, Alfarraj, & Alalwan, 2016)

1.1.11 *Prototyping*

Prototyping adalah penyesuaian untuk *System Development Life Cycle* (SDLC) yang digunakan untuk mempercepat pengembangan dan pengiriman aplikasi (Kendall dan Kendall, 2010; Turban et al, 2004).

1.2 Teori-Teori Khusus

1.2.1 *System Development Life Cycle (SDLC)*

Menurut (Satzinger, Jackson, & Burd, *Systems Analysis and Design : In a Changing World*, 2016), *System Development Life Cycle* adalah seluruh proses yang terdiri dari

aktivitas-aktivitas yang diperlukan untuk membangun, meluncurkan dan menjaga sebuah sistem informasi.

Berikut adalah proses inti yang diperlukan untuk mengembangkan sebuah aplikasi baru:

- a. Melakukan identifikasi terhadap masalah, dan mendapatkan persetujuan
- b. Merencanakan seperti apa *project* yang akan dijalankan untuk menyelesaikan masalah yang telah diidentifikasi
- c. Melakukan analisis dan berusaha untuk memahami secara detail tentang masalah yang sedang dihadapi
- d. Mendesain komponen-komponen dan rancangan sistem yang bisa digunakan untuk menyelesaikan masalah yang ada
- e. Membuat, menguji, dan mengintegrasikan komponen-komponen yang ada dalam sistem yang telah dibuat
- f. Menyelesaikan *system testing* dan mengimplementasikan solusi yang telah dihasilkan

1.2.2 *Object Oriented Analysis (OOA)*

Menurut (Satzinger, Jackson, & Burd, Systems Analysis and Design : In a Changing World, 2016) *Object Oriented Analysis* adalah proses untuk mengidentifikasi dan menentukan kumpulan *use case* dan juga kumpulan objek (*class*) didalam sistem baru yang akan dirancang.

1.2.3 *Object Oriented Design (OOD)*

Menurut (Satzinger, Jackson, & Burd, Systems Analysis and Design : In a Changing World, 2016) *Object Oriented Design* adalah adalah proses yang mengidentifikasi dan menggambarkan kelas di dalam setiap lapisan yang digunakan untuk memanggil metode kelas yang terlibat.

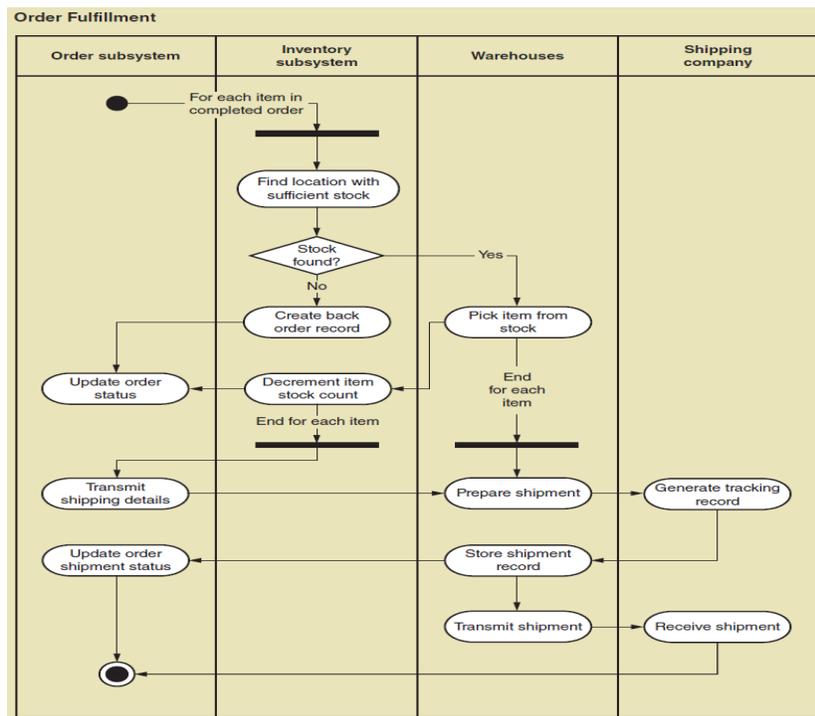
1.2.4 *Unified Modeling Language (UML)*

Menurut (Satzinger, Jackson, & Burd, Systems Analysis and Design : In a Changing World, 2016) *Unified Modeling Language* adalah set standar diagram dan konstruksi model yang digunakan dalam pengembangan sistem.

1.2.5 Activity Diagram

Menurut (Satzinger, Jackson, & Burd, Systems Analysis and Design : In a Changing World, 2016) *Activity Diagram* adalah diagram yang digunakan untuk menggambarkan aktivitas yang dilakukan, dan siapa saja orang yang melakukan aktivitas tersebut secara berurutan.

Dapat disimpulkan bahwa *Activity Diagram* digunakan untuk menjelaskan urutan aktivitas dari awal hingga akhir dalam sebuah proses bisnis, dan juga menggambarkan siapa yang melakukan aktivitas serta transisi dari satu aktivitas ke aktivitas lain.



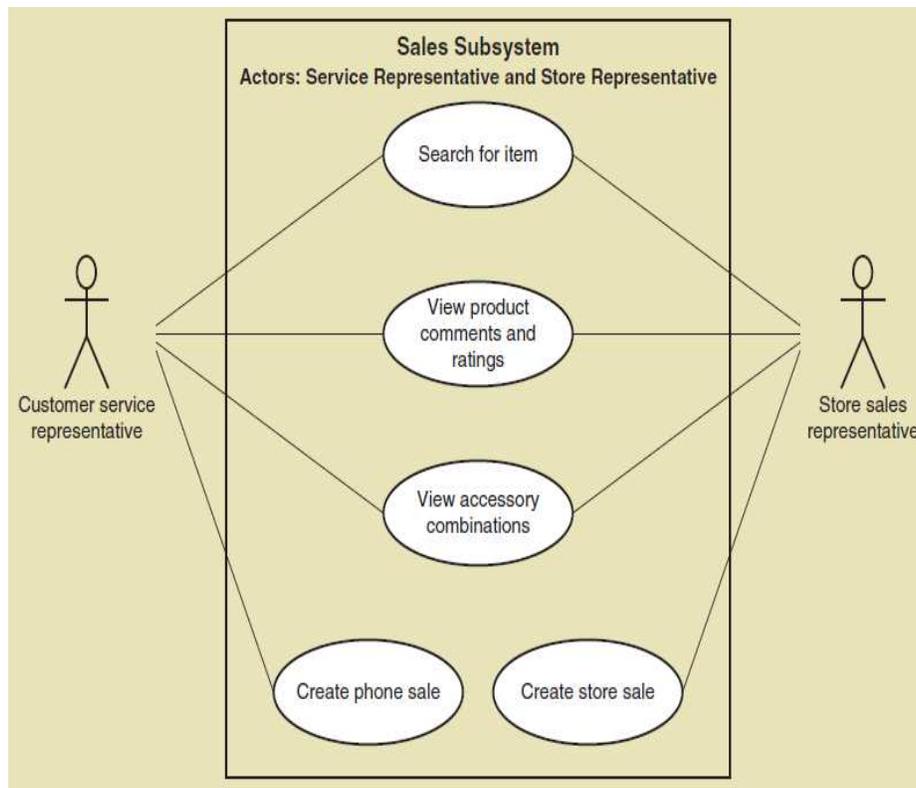
Gambar 2. 1 Contoh Activity Diagram

Sumber: Satzinger, J.W., Jackson, R.B & Burd S.D. (2016:61) *Systems Analysis and Design in a Changing World 7th Edition*.

1.2.6 Use Case Diagram

Menurut (Satzinger, Jackson, & Burd, Systems Analysis and Design : In a Changing World, 2016) *Use Case Diagram* adalah diagram yang digunakan untuk menggambarkan *use case* yang ada pada sebuah sistem dan hubungannya dengan para *actor* pengguna sistem. Selain itu, menurut (Satzinger, Jackson, & Burd, Systems Analysis and Design : In a Changing World, 2016) *Automation Boundary* pada sistem merupakan pemisah antara

aplikasi dan *user* yang menggunakan sistem, tetapi merupakan bagian dari sebuah sistem secara keseluruhan.



Gambar 2. 2 Contoh Use Case Diagram

Sumber: Satzinger, J.W., Jackson, R.B & Burd S.D. (2016:86) *Systems Analysis and Design in a Changing World 7th Edition*.

1.2.7 Use Case Description

Menurut (Satzinger, Jackson, & Burd, *Systems Analysis and Design : In a Changing World*, 2016), *Use Case Description* adalah tulisan yang digunakan untuk mendaftarkan dan mendeskripsikan secara detail proses yang berjalan dari sebuah *use case*.

A. *Fully Developed Use Case Description*: Deskripsi secara formal untuk mendokumentasikan seluruh proses yang terjadi pada sebuah *use case*. Dalam *Fully Developed Use Case Description* harus terdapat *precondition* dan *postcondition*. *Precondition* adalah kondisi yang harus terpenuhi sebelum sebuah *use case* dijalankan, sedangkan *postcondition* adalah kondisi yang harus terjadi setelah sebuah *use case* selesai dijalankan.

Use case name:	<i>Create customer account.</i>
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.

Gambar 2. 3 Contoh *Fully Developed Use Case Description (Bagian 1)*

Sumber: Satzinger, J.W., Jackson, R.B & Burd S.D. (2016:134) *Systems Analysis and Design in a Changing World 7th Edition.*

Flow of activities:	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.	

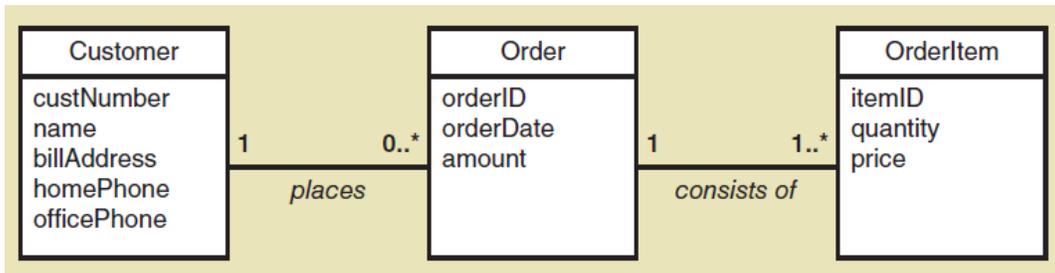
Gambar 2. 4 Contoh *Fully Developed Use Case Description (bagian 2)*

Sumber: Satzinger, J.W., Jackson, R.B & Burd S.D. (2016:134) *Systems Analysis and Design in a Changing World 7th Edition.*

1.2.8 Class Diagram

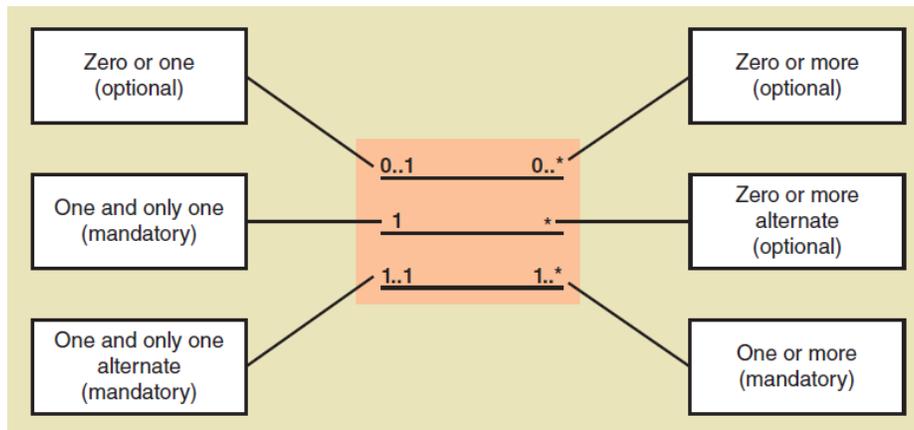
Menurut (Satzinger, Jackson, & Burd, *Systems Analysis and Design : In a Changing World*, 2016), *class* adalah kategori atau klasifikasi yang digunakan untuk mendeskripsikan kumpulan dari objek-objek yang ada. *Domain Classes* adalah *class* yang menggambarkan objek dari *problem domain*. *Class diagram* sendiri adalah diagram yang digunakan untuk menunjukkan *class* dan asosiasi antar *class* yang ada dalam sebuah sistem, sedangkan *domain*

Model Class Diagram adalah bentuk dari diagram UML yang menunjukkan hal-hal yang penting didalam *problem domain* yang ada.



Gambar 2. 5 Contoh Domain Model Class Diagram

Sumber: Satzinger, J.W., Jackson, R.B & Burd S.D. (2016:104) *Systems Analysis and Design in a Changing World 7th Edition*.



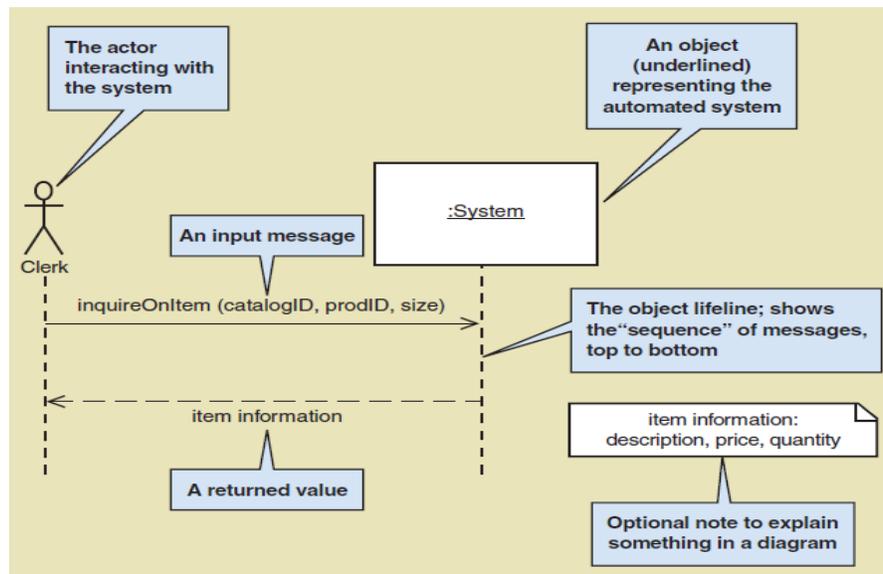
Gambar 2. 6 Notasi UML untuk Multiplicity pada Associations

Sumber: Satzinger, J.W., Jackson, R.B & Burd S.D. (2016:104) *Systems Analysis and Design in a Changing World 7th Edition*.

1.2.9 System Sequence Diagram

Menurut (Satzinger, Jackson, & Burd, *Systems Analysis and Design : In a Changing World*, 2016) *System Sequence Diagram* adalah diagram yang digunakan untuk menunjukkan urutan dari pesan-pesan yang dikirimkan antara *actor* dengan sistem yang digunakan pada saat menjalankan sebuah skenario atau *Use Case*.

System Sequence Diagram sendiri merupakan salah satu tipe dari *Interaction Diagram* yang digunakan untuk menggambarkan interaksi antara *user* dengan sistem yang digunakan.



Gambar 2. 7 Contoh System Sequence Diagram

Sumber: Satzinger, J.W., Jackson, R.B & Burd S.D. (2016:140) *Systems Analysis and Design in a Changing World 7th Edition*

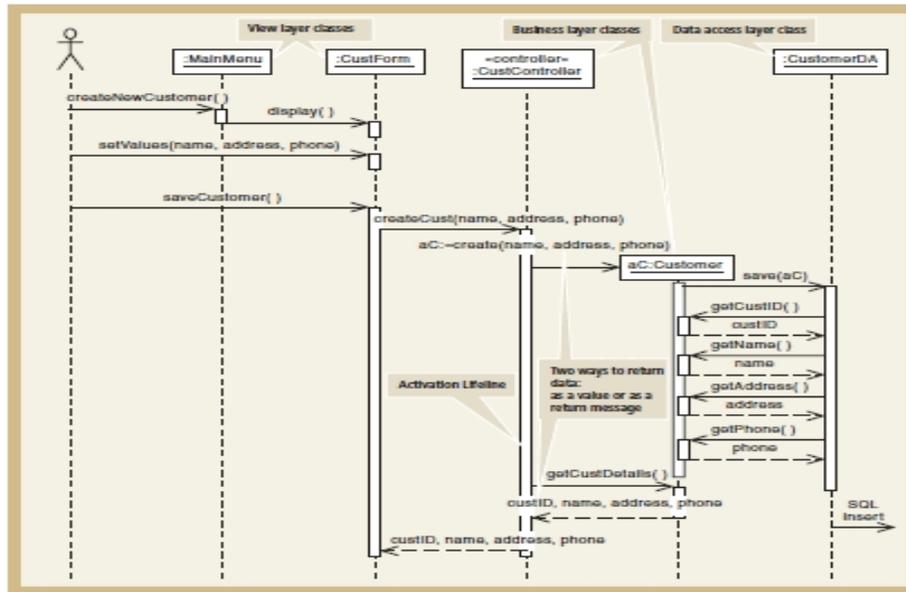
Terdapat empat langkah yang dapat digunakan untuk membantu dalam membuat *system sequence diagram*:

- Mengidentifikasi terlebih dahulu *input message* yang diberikan dari *user* ke sistem.
- Mendeskripsikan *message* yang diberikan oleh *externalactor* ke *system* dengan menggunakan notasi *message* yang telah ditentukan.
- Mengidentifikasi dan menambahkan *special condition* yang terdapat dalam sebuah *input message*.
- Mengidentifikasi dan menambahkan *return message* yang merupakan *output* dari sistem terhadap *input message* dari *user*.

1.2.10 Multi layer Sequence Diagram

Menurut (Satzinger, Jackson, & Burd, Systems Analysis and Design : In a Changing World, 2016) *Multi layer Sequence Diagram* adalah rincian dari seluruh sistem sequence diagram dimana objek sistem diganti dengan rincian objek yang ada di sistem.

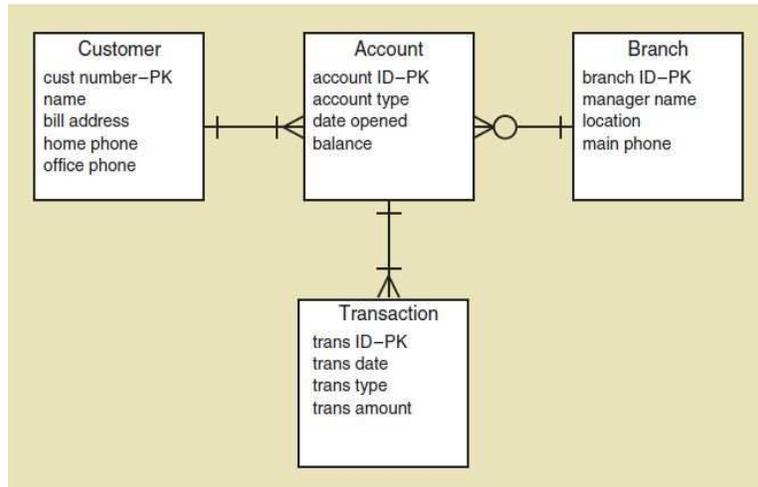
Gambar 2. 8 Contoh Multi layer Sequence Diagram



Sumber: Satzinger, J.W., Jackson, R.B & Burd S.D. (2016:140) *Systems Analysis and Design in a Changing World 7th Edition*

1.2.11 Entity Relationship Diagram

Menurut (Satzinger, Jackson, & Burd, Systems Analysis and Design : In a Changing World, 2016) *Entity Relationship Diagram (ERD)* adalah diagram yang terdiri dari entitas data dan hubungan antar entitas data yang ada. *ERD* bukanlah *UML diagram*, tetapi sangat umum digunakan dan sangat mirip dengan *Class Diagram* pada *UML diagram*.



Gambar 2. 9 Contoh Entity Relationship Diagram

Sumber: Satzinger, J.W., Jackson, R.B & Burd S.D. (2016 .102) *Systems Analysis and Design in a Changing World 7th Edition*.

1.2.12 User Interface

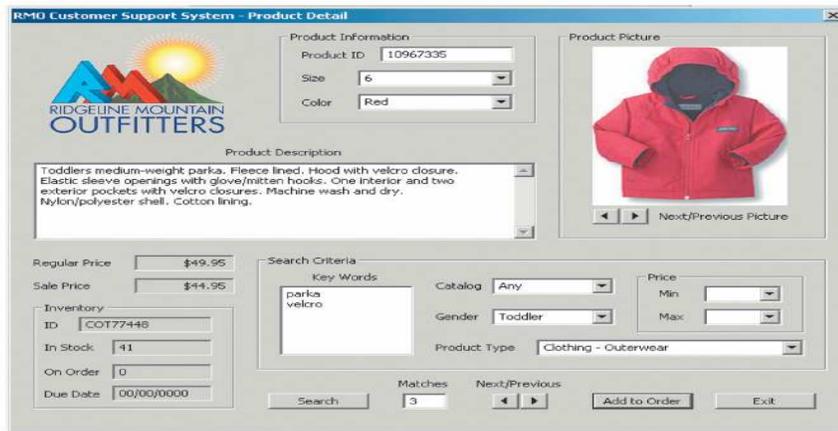
Menurut (Satzinger, Jackson, & Burd, *Systems Analysis and Design : In a Changing World*, 2016) *User Interface* adalah *input* dan *output* yang secara langsung melibatkan pengguna sistem. *User Interface* bisa digunakan oleh pihak internal dan eksternal. Desain *User Interface* bisa bervariasi tergantung pada faktor-faktor seperti tujuan penggunaan *interface*, karakteristik *user*, dan karakteristik dari *device* untuk menampilkan *interface*.

Donald Norman adalah peneliti terkemuka dalam bidang *Human Computer Interaction* atau Interaksi Manusia Komputer, bidang studi yang berkonsentrasi dengan efisiensi dan efektivitas *User Interface* untuk sistem computer, teknologi *input* dan *output* yang berorientasi pada manusia, dan aspek psikologis dari *User Interface*. Norman mengusulkan dua prinsip utama untuk memastikan interaksi yang baik antara orang dan mesin: *affordance* dan *visibility*.

Affordance adalah munculnya sebuah kontrol tertentu, dimana kontrol tersebut memiliki fungsi tertentu. *Visibility* adalah kondisi ketika sebuah kontrol bisa terlihat oleh *user* sehingga *user* mengetahui bahwa kontrol tersebut tersedia. *Visibility* juga berarti kontrol tersebut harus menyediakan *feedback* kepada *user*, untuk memastikan bahwa sistem merespon terhadap perintah yang diberikan oleh *user*.

Menurut (Satzinger, Jackson, & Burd, *Systems Analysis and Design : In a Changing World*, 2016) Selain *affordance* dan *visibility* ada beberapa aturan universal lain yang harus diperhatikan saat akan mendesain *User Interface*:

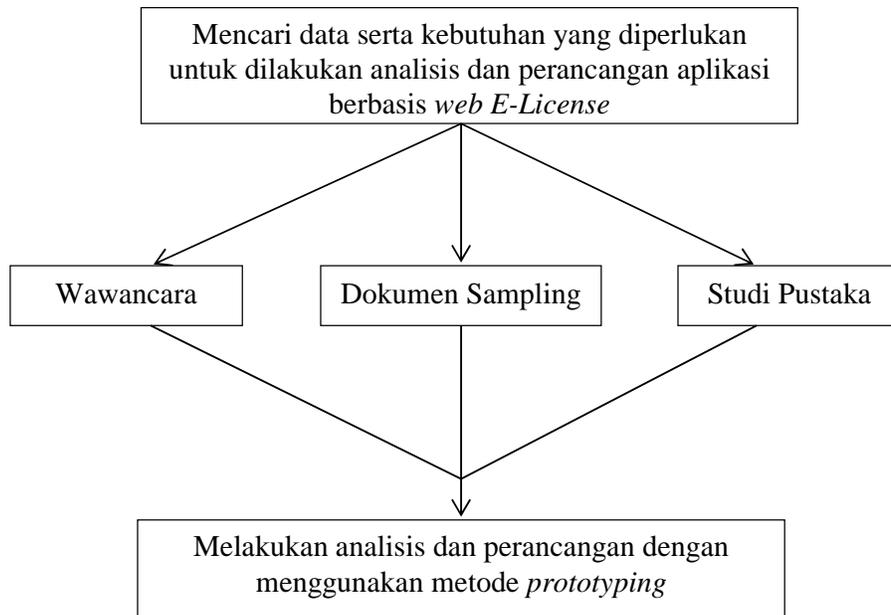
- a. *Consistency*: *User Interface* harus didesain secara konsisten dari segi fungsi dan tampilan. Informasi yang ditampilkan, susunan menu, bentuk serta ukuran *icon*, dan urutan penggunaan *User Interface* untuk menyelesaikan tugas-tugas tertentu juga harus konsisten.
- b. *Discoverability*: suatu fitur yang memberikan petunjuk, untuk membantu user menemukan fitur tersembunyi.
- c. *Feedback*: Dari setiap tindakan (*action*) yang dilakukan oleh *user*, komputer harus bisa menghasilkan *feedback* sehingga *user* mengetahui bahwa tindakan (*action*) tersebut memang sudah diproses oleh sistem. Ada 2 jenis *feedback* yang bisa diterapkan didalam *User Interface* yaitu:
 1. *Audible Feedback*: Suara pada saat menekan tombol *keyboard* dan suara pada saat menekan tombol *on/off*.
 2. *Visible Feedback*: Perubahan *progress bar* pada saat melakukan download *file* berukuran besar.
- d. *Usability and efficiency*: Salah satu pertimbangan desain utama dalam desain UI adalah bagaimana cara meningkatkan efisiensi penggunaan system.
- e. *Easy Reversal of Action*: *User* perlu merasa bahwa mereka bisa membatalkan atau mengulangi tindakan (*action*) tertentu tanpa harus merasa kesulitan. Dikarenakan *user* belajar tentang cara menggunakan sebuah sistem dengan cara bereksperimen, sehingga pada saat *user* melakukan sebuah kesalahan maka mereka bisa membatalkan tindakan (*action*) tersebut.
- f. *Reducing Short-Term Memory Load*: Berdasarkan percobaan yang dilakukan oleh para ahli psikologi, diketahui bahwa manusia hanya bisa mengingat sekitar 7 informasi saja dalam satu waktu. Desainer *User Interface* harus mengindari untuk mendesain *User Interface* yang meminta *user* untuk mengingat terlalu banyak perpindahan dari satu *form* ke *form* lain, maupun perpindahan dari satu *dialog box* ke *dialog box* yang lain pada saat menggunakan sebuah system.



Gambar 2.10 Contoh *User Interface*

Sumber: Satzinger, J.W., Jackson, R.B & Burd S.D. (2016:240) *Systems Analysis and Design in a Changing World 7th Edition*.

1.3 Kerangka Pikir



Gambar 2. 11 Kerangka Pikir

Kegiatan yang kami lakukan berdasarkan kerangka pikir adalah melakukan wawancara dengan pihak terkait di Bank Indonesia untuk mengetahui permasalahan yang ada di Bank Indonesia dan juga mengetahui proses perizinan yang sedang berlangsung. Selain itu kami juga mempelajari dokumen – dokumen yang diberikan oleh Bank Indonesia kepada kami. Setelah itu kami melakukan tahap mencari studi pustaka yaitu mencari teori-teori yang berkaitan dengan skripsi kami di dalam jurnal dan juga buku.

Setelah ketiga tahapan tersebut selesai, lalu kami melanjutkan ketahapan selanjutnya yaitu melakukan analisis dan perancangan dari *web E-License*. Pada tahap analisis kami membuat UML diagram yaitu *activity diagram*, *usecase diagram*, *usecase description*, *class diagram*, dan *system sequence diagram*. Selanjutnya setelah semua UML diagram selesai dibuat kami mulai melakukan perancangan *user interface*.