

BAB II

LANDASAN TEORI

2.1. Konsep Dasar Sistem

Pada umumnya setiap organisasi mempunyai sistem informasi dalam mengumpulkan, menyimpan, melihat dan menyalurkan informasi dalam membuat perancangan sistem informasi. Konsep dasar sistem merupakan sekelompok komponen berbasis komputer yang dibuat oleh manusia serta mengkoordinasikan sumber daya manusia dan computer untuk mengubah sistem masukan menjadi sistem keluaran untuk mencapai tujuan dan sasaran yang telah ditetapkan sebelumnya.

2.1.1. Pengertian Sistem

Menurut Kadir (2014 : 8) bahwa “sistem informasi mencakup sejumlah komponen (manusia, komputer, teknologi, dan prosedur kerja) ada sesuatu yang di proses (data menjadi informasi) dan dimaksudkan untuk mencapai suatu sasaran atau tujuan”.

Menurut Taufiq (2013:1) “sistem adalah sekumpulan dari sub-sub sistem yang saling berhubungan dan bekerja sama”.

Berdasarkan definisi diatas menyimpulkan pengertian sistem adalah sekumpulan elemen dari sub-sub sistem atau komponen atau unsur-unsur yang terpadu, saling terkait atau saling berinteraksi dan saling berhubungan satu sama lain untuk mencapai suatu tujuan.

2.1.2 Karakteristik sistem

Menurut Hartono (2013:14) bahwa sebuah sistem memiliki paling sedikit tujuh karakteristik sebagai berikut :

1. Komponen

Bagian-bagian atau elemen-elemen yang dapat berupa benda atau manusia, berbentuk nyata atau abstrak dan disebut subsistem.

2. Penghubung antar bagian (*interface*)

Sesuatu yang bertugas menjembatani satu bagian dengan bagian lain, dan memungkinkan terjadinya interaksi/ komunikasi antar bagian.

3. Batas (*Boundary*)

Sesuatu yang membedakan anatara satu sistem dengan sistem atau dengan sistem-sistem lainnya.

4. Lingkungan (*Environment*)

Segala sesuatu yang berada diluar sistem dan dapat bersifat menguntungkan atau merugikan sistem yang bersangkutan.

5. Masukan (*input*)

Sesuatu yang merupakan bahan untuk diolah atau di proses oleh sistem.

6. Mekanisme Pengolahan (*Processing*)

Perangkat dan prosedur untuk mengubah masukan menjadi eluaran dan menampilkannya.

7. Keluaran (*Output*)

Berbagai macam bentuk hasil atau produk yang dikeluarkan dari pengolahan.

Berdasarkan karakteristik sistem diatas menyimpulkan bahwa suatu karakteristik sistem merupakan sifat-sifat tertentu yang mencirikan bahwa hal tersebut bisa dikatakan sebagai suatu sistem.

2.1.3. Klasifikasi Sistem

Menurut Yakub (2013:4) pada buku pengantar informasi, sistem dapat diklasifikasikan dari beberapa sudut pandang diantaranya :

1. Sistem Abstrak (*abstract system*)

Sistem abstrak adalah sistem yang berupa pemikiran atau ide-ide yang tidak tampak secara fisik, sistem teologia yang berisi gagasan tentang hubungan manusia dengan tuhan merupakan contoh *abstract system*.

2. Sistem Fisik (*physical system*)

Sistem fisik adalah sistem yang ada secara fisik, sistem komputer, sistem akuntansi, sistem produksi, sistem sekolah dan sistem transportasi merupakan contoh *physical system*.

3. Sistem Tertentu (*deterministic system*)

Sistem tertentu adalah sistem *deterministic system* yang beroperasi dengan tingkah laku yang dapat diprediksi, interaksi antara bagian dapat dideteksi dengan pasti sehingga keluarannya dapat diramalkan. Sistem komputer sudah diprogramkan, merupakan contoh karena program komputer dapat diprediksi dengan pasti.

4. Sistem Tak Tentu (*Probabilistic system*)

Sistem tak tentu adalah suatu sistem yang kondisi masa depannya tidak dapat diprediksikan karena mengandung unsur probabilitas, sistem arisan merupakan

contoh *probabilistic system* karena sistem arisan tidak dapat diprediksikan dengan pasti.

5. Sistem Tertutup (*close system*)

Sistem tertutup merupakan sistem yang tidak bertukar materi, informasi, atau energi dengan lingkungan, Sistem ini tidak berinteraksi dan tidak dipengaruhi oleh lingkungan, misalnya reaksi kimia dalam tabung terisolasi.

6. Sistem Terbuka (*open system*)

Sistem ini adalah sistem yang berhubungan dengan lingkungan dan dipengaruhi oleh lingkungan, sistem perdagangan merupakan contoh *open system*, karena dapat dipengaruhi oleh lingkungan.

2.1.4. Informasi

Menurut Hutahaean (2015:9) Informasi adalah “data yang diolah menjadi bentuk yang lebih berguna dan lebih berarti bagi penerimanya”.

Dalam melakukan perancangan sistem terlebih dahulu harus mengerti komponen-komponen suatu sistem, salah satunya adalah informasi. Dari mana data dan informasi tersebut diperoleh dan kemana hasil pengolahan data dan informasi tersebut diperlukan. Informasi merupakan salah satu sumber daya yang sangat diperlukan dalam sistem.

2.1.5. Sistem Informasi

Hutahaean (2014:13) menjelaskan bahwa:

Sistem Informasi adalah suatu sistem didalam suatu organisasi yang mempertemukan kebutuhan pengelolaan transaksi harian, mendukung operasi, bersifat manajerial dan kegiatan strategi dari suatu organisasi dan

menyediakan strategi dari suatu organisasi dan menyediakan pihak luar tertentu dengan laporan-laporan yang dibutuhkan.

Sistem informasi dalam suatu organisasi dapat dikatakan sebagai suatu sistem yang menyediakan informasi bagi semua tingkatan dalam organisasi tersebut kapan saja di perlukan. Sistem ini menyimpan, mengambil, mengubah, mengolah dan mengkomunikasikan informasi yang diterima dengan menggunakan sistem informasi atau peralatan sistem lainnya.

2.1.6. Pengertian Persediaan

Menurut Assauri dalam Tuerah (2014 :527), “ persediaan adalah suatu aktiva yang meliputi barang-barang milik perusahaan dengan maksud untuk dijual dalam suatu periode usaha yang normal atau persediaan barang yang masih dalam pengerjaan atau proses produksi, atau persediaan bahan baku yang menunggu penggunaannya dalam suatu proses produksi”.

Menurut Rangkuti dalam Tamodia (2013 : 15) persediaan memiliki fungsi penting bagi perusahaan, yaitu :

1. Agar dapat memenuhi permintaan yang diantisipasi akan terjadi
2. Untuk menyeimbangkan produksi dengan distribusi
3. Untuk memperoleh keuntungan dari potongan kuantitas, karena membeli dalam jumlah yang banyak
4. Untuk heading dari inflasi dan perubahan harga.
5. Untuk menghindari kekurangan persediaan yang dapat terjadi karena cuaca, kekurangan pasokan, mutu dan ketidak tepatan pengiriman
6. Untuk menjaga kelangsungan operasi dengan cara persediaan dalam proses.

2.1.6. Basis Data (*Database*)

Menurut Rosa dan Salahuddin (2013 :43) “ Sistem basis data adalah sistem terkomputerisasi yang tujuan utamanya adalah memelihara data yang sudah diolah atau informasi dan membuat informasi tersedia saat dibutuhkan”. Pada intinya basis data adalah media untuk menyimpan data agar dapat diakses dengan mudah dan cepat.

1. *Database Management System (DBMS)*

Menurut Rosa dan Salahuddin (2013:44) “(*DBMS Database Management System*) atau dalam bahasa Indonesia sering disebut sebagai sistem manajemen Basis Data adalah suatu sistem aplikasi yang digunakan untuk menyimpan, mengelola, dan menampilkan data”.

Menurut Rosa dan salahuddin (2013:45) berikut ini adalah 4 (empat) macam DBMS versi komersil yang paling banyak digunakan didunia saat ini adalah *Oracle, Microsoft SQL Server, IBMDB2, dan Microsoft Acces*.

Sedangkan DBMS versi *open source* yang cukup berkembang dan paling banyak digunakan saat ini adalah *MySQL, Firebird, dan SQLite*. Hampir semua DBMS mengadopsi SQL sebagai bahasa untuk mengelola data pada DBMS.

2. *Structured Query Language (SQL)*

Menurut Rosa dan salahuddin (2013:46) “SQL (*Structured Query Language*) adalah bahasa yang digunakan untuk mengelola data pada RDBMS”. SQL awalnya dikembangkan berdasarkan teori aljabar dan kalkulus.

3. MySQL

Menurut Abdul Kadir (2014:242) SQL (*Struktured Query Language*) adalah “bahasa yang digunakan untuk mengakses basis data yang tergolong relasional”.

Sesungguhnya SQL tidak terbatas hanya untuk mengambil data (*query*), tetapi juga dapat dipakai untuk menciptakan tabel (*CREATE*), menghapus tabel (*DELETE*), menambahkan data ke tabel (*INSERT*), menghapus data di tabel (*DROP*), mengganti data di tabel (*UPDATE*), dan berbagai operasi yang lain.

2.1.7. XAMPP

Menurut Buana (2014 : 4) XAMPP adalah perangkat lunak *opensource* yang biasa di unggah secara gratis dan bias dijalankan disemua sistem operasi seperti Windows, Linux, Solaris, dan Mac.

XAMPP ini sendiri dibuat dan di kembangkan oleh Apache Friends perangkat lunak tersebut berisi kumpulan beberapa perangkat lunak yang dibutuhkan antara lain PHP, MySQL, dan PHPMyadmin. Dengan melakukan instalasi XAMPP tersebut, maka semua perangkat lunak yang dibutuhkan sudah otomatis terinstall di hardisk termasuk pengaturan file-file yang diperlukan.

2.1.8. Pengertian Java

Menurut Rosa dan M.shalahuddin (2013:103) java dikembangkan oleh perusahaan Sun Microsystem. Java menurut definisi dari *Sun Microsystem* adalah nama untuk sekumpulan teknologi untuk membuat dan menjalankan

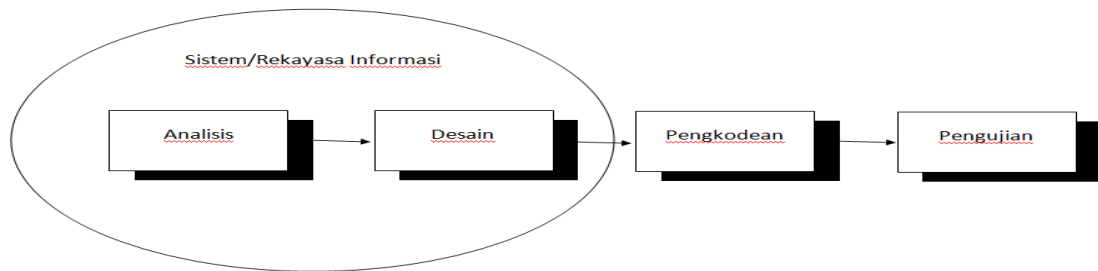
perangkat lunak pada komputer *standalone* ataupun pada lingkungan jaringan. Java 2 adalah generasi kedua dari *java platform*.

Java berdiri diatas sebuah mesin *interpreter* yang diberikan nama *java virtual machine* (JVM). JVM inilah yang akan membaca *bytecode* dan *file*. Class dari suatu program sebagai representasi langsung program yang berisi bahasa mesin. Oleh karena itu bahasa java disebut sebagai bahasa pemrograman yang *portable* karena dapat dijalankan pada berbagai sistem operasi, asalkan pada sistem operasi tersebut terdapat JVM.

Java merupakan bahasa pemrograman objek murni karena semua kode programnya dibungkus dalam kelas. Saat ini *Sunmicrosystem* sudah diakuisi *Oracle Corporation* sehingga pengembangan java diteruskan oleh *Oracle Corporation*.

2.1.9. Model Pengembangan Perangkat Lunak

Menurut Rosa dan Salahuddin (2013:28) Model SDLC air terjun (*waterfall*) sering juga disebut sekuensi linier (*sequential linier*) atau alur hidup klasik (*classic life cycle*). Model air terjun menyediakan pendekatan alur hidup perangkat lunak secara sekuensial atau terurut dimulai dari analisis desain, pengkodean, pengujian, dan tahap pendukung (*support*). Berikut adalah gambar model air terjun beserta penjelasannya.



Sumber : Rosa dan salahuddin (2013:29)

Gambar II.1. Ilustrasi Model Waterfall

Berikut ini penjelasannya:

1. Analisis Kebutuhan Perangkat Lunak

Proses pengumpulan kebutuhan dilakukan secara intensif untuk men spesifikasikan kebutuhan perangkat lunak agar dapat dipahami perangkat lunak seperti apa yang dibutuhkan oleh *user*. Spesifikasi kebutuhan perangkat lunak pada tahap ini perlu untuk didokumentasikan.

2. Desain

Desain perangkat lunak adalah proses multi langkah yang fokus pada desain pembuatan program pembuatan perangkat lunak termasuk struktur data, arsitektur perangkat lunak, representasi anta muka, dan prosedur pengkodean. Tahap ini mentranslasi kebutuhan perangkat lunak dari tahap analisis kebutuhan ke representasi desain agar dapat di implemantasikan menjadi program pada tahap selanjutnya desain perangkat lunak yang dihasilkan pada tahap ini juga perlu di dokumentasikan.

3. Pembuatan Kode Program

Desain harus ditranslasikan kedalam program perangkat lunak hasil dari tahap ini adalah program komputer sesuai dengan desain yang telah dibuat pada tahap desain.

4. Pengujian

Pengujian fokus pada perangkat lunak secara dari segi logik dan fungsional dan memastikan bahwa semua bagian sudah di uji. Hal ini dilakukan untuk meminimalisir kesalahan (*error*) dan memastikan keluaran yang dihasilkan sesuai dengan yang digunakan.

5. Pendukung (*support*) atau pemeliharaan (*maintenance*)

Tidak menutup kemungkinan sebuah perangkat lunak mengalami perubahan ketika sudah dikirimkan ke user. Perubahan bisa terjadi karena adanya kesalahan yang muncul dan tidak terdeteksi saat pengujian atau perangkat lunak harus beradaptasi dengan lingkungan baru.

2.2. Teori Pendukung

Peralatan pendukung merupakan alat yang tepat digunakan untuk menggambarkan bentuk *logical* model dari suatu sistem, dimana simbol-simbol, lambang-lambang dan diagram-diagram menunjukan secara tepat arti fisiknya.

2.2.1. *Entity Relationship Diagram* (ERD)

Menurut Ladjamudin (2013:142), “*Entity Relationship Diagram* (ERD) adalah suatu model jaringan yang menggunakan susunan data yang disimpan dalam sistem secara abstrak”. ERD digunakan oleh professional sistem untuk berkomunikasi dengan pemakai eksekutif tingkat tinggi dalam suatu organisasi.

Adapun simbol-simbol dari Entity Relationship Diagram (ERD) adalah sebagai berikut:

1. Entitas

Suatu kumpulan objek atau sesuatu yang dapat dibedakan atau dapat didefinisikan secara unik.

2. Relationship

Hubungan yang terjadi antara satu entitas atau lebih.

3. Atribut

Karakteristik dari entitas atau relationship yang menyediakan penjelasan detail entitas atau relation.

4. Link

Baris sebagai penghubung antara himpunan, relasi dan himpunan entitas dari atributnya.

2.2.2. Logical Record Structured (LRS)

Menurut Iadjamudin (2013:159) "Language Record Structure (LRS) merupakan hasil transformasi ERD ke LRS yang memulai proses kardinalitas dan menghilangkan atribut-atribut yang saling berelasi"

Dapat disimpulkan bahwa Language Record Structure (LRS) merupakan cara atau teknik untuk menggambarkan basis data berupa relasi antar table yang men transformasikan ERD ke LRS melalui proses kardinalitas pentransformasian ERD ke LRS ini memiliki aturan –aturan tertentu yang mempengaruhi langkah pentransformasian yaitu kardinalitas. Adapun kardinalitas tersebut Iadjamudin (2013:160) "yaitu:

1. 1:1(*one to one*)

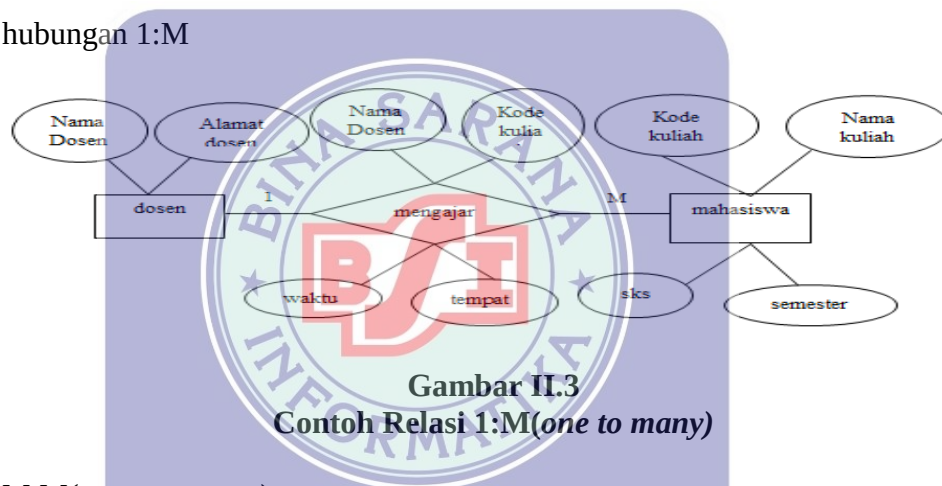
Relasi yang terjadi antara suatu *entity* dengan *entity* lainnya yang memiliki hubungan 1:1



Gambar II.2.
Contoh relasi 1:1(*one to one*)

2. 1:M (*one to many*)

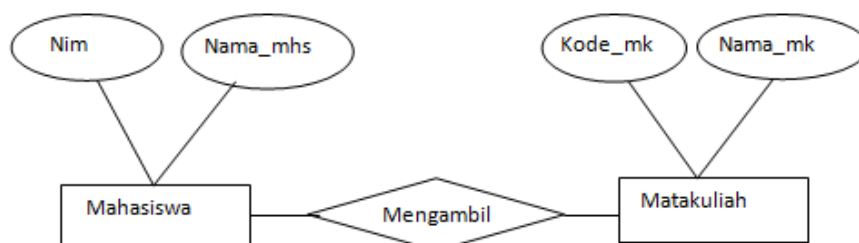
Relasi yang terjadi antar suatu *entity* dengan *entity*lainya yang memiliki hubungan 1:M



Gambar II.3
Contoh Relasi 1:M(*one to many*)

3. M:M(*many to many*)

Relasi yang terjadi antar suatu *entity* dengan *entity*lainya yang memiliki hubungan M:N pada relasi ini biasa digunakan table bantuan untuk memecakan relasi tersebut menjadi 1:1 atau 1:M



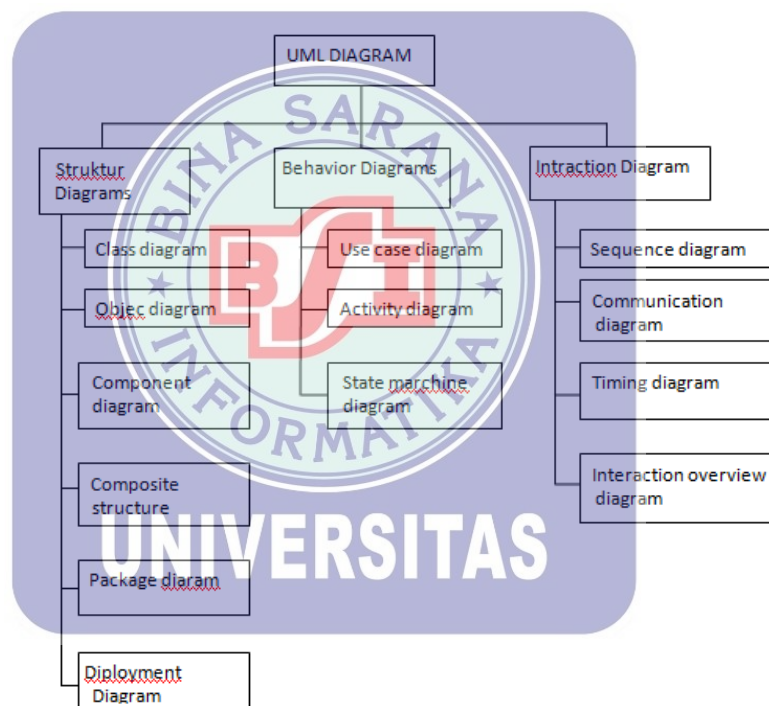
Gambar II.4.

Contoh Relasi M:N *many to many*

2.2.3. *Unified Modelling Language (UML)*

Menurut Rosa dan Salahuddin (2013:137) *Unified Modelling Language* atau biasa disingkat UML merupakan bahasa visual untuk pemodelan dan komunikasi mengenai sebuah sistem dengan menggunakan diagram dan teks-teks pendukung. UML hanya berfungsi untuk melakukan peodelan jadi penggunaan UML tidak terbatas pada metodologi tertentu, meskipun pada kenyataanya UML paling banyak digunakan pada metodologi berorientasi objek.

Berikut Contoh Diagram UML sebagai berikut :



Sumber Rosa dan Shalahudin (2014:140)

Gambar II.5.
Contoh UML Diagram

Berikut yang termasuk diagram UML, terdiri dari :

1. *Use Case Diagram*

Rosa dan Salahuddin (2013:155) *Use case* atau diagram *use case* merupakan pemodelan untuk melakukan (*hebehavior*) sistem informasi yang akan dibuat. *Use Case* mendeskripsikan sebuah interaksi antara suatu atau lebih aktor dengan aplikasi yang akan dibuat. Secara kasar *Use Case* digunakan untuk mengetahui fungsi atau proses apa saja yang ada di dalam sebuah aplikasi dan siapa saja yang berhak menggunakan fungsi-fungsi atau proses-proses itu.

Syarat penamaan pada *Use Case* adalah nama didefinisikan sesimple mungkin dan dapat dipahami. Ada dua hal utama pada *Use Case* yaitu pendefinisian apa yang disebut aktor dan *Use Case*.

- a. Aktor merupakan orang, proses, atau aplikasi lain yang berinteraksi dengan aplikasi yang akan dibuat diluar aplikasi yang akan dibuat itu sendiri, jadi walaupun simbol dari aktor adalah gambar orang, tapi aktor belum tentu merupakan orang.
- b. *Use Case* merupakan fungsi-fungsi atau proses-proses yang disediakan aplikasi sebagai unit-unit yang saling bertukar pesan atau berinteraksi antar unit, proses atau aktor

2. *Class Diagram*

Rosa dan Salahuddin (2013:141), menjelaskan tentang *class diagram* sebagai berikut :

Class diagram menggambarkan struktur sistem dari segi pendefinisian kelas-kelas yang akan dibuat untuk membangun sistem. Diagram kelas dibuat agar pembuat program atau programmer membuat kelas-kelas

sesuai rancangan didalam diagram kelas agar antara dokumentasi perancangan dan perangkat lunak sinkron.

Diagram kelas dibuat agar pembuat program atau *programmer* membuat kelas-kelas sesuai rancangan didalam diagram kelas agar antara dokumentasi perancangan dan perangkat lunak sinkron. Banyak berbagai kasus, perancangan kelas yang dibuat tidak sesuai dengan kelas-kelas yang dibuat pada perangkat lunak, sehingga tidaklah ada gunanya lagi sebuah perancangan karena apa yang dirancang dan hasil jadinya tidak sesuai.

3. Activity Diagram

Menurut Rosa dan Salahuddin (2013:161) menjelaskan tentang *activity* dengan sebagai berikut :

Activity diagram menggambarkan *workflow* (aliran kerja) atau aktivitas dari sebuah sistem atau proses bisnis atau semua yang ada pada perangkat lunak. Yang perlu di perhatikan disini adalah bahwa diagram aktivitas menggambarkan aktivitas sistem bukan apa yang dilakukan aktor, jadi aktivitas yang dapat dilakukan oleh sistem.

Diagram aktivitas juga banyak digunakan untuk mendefinisikan hal-hal berikut :

- a. Rancangan proses bisnis dimana setiap urutan aktivitas yang digambarkan merupakan proses bisnis sistem yang di definisikan.
- b. Urutan atau pengelompokan tampilan dari sistem / *user interface* dimana setiap aktivitas dianggap memiliki sebuah rancangan tampilan.
- c. Rancangan pengujian dimana setiap aktivitas dianggap memerlukan sebuah pengujian yang perlu di definisikan kasus ujinya.
- d. Rancangan menu yang ditampilkan pada perangkat lunak.

4. *Sequence Diagram*

Menurut Rosa dan Salahuddin (2013:165) menjelaskan “*sequence* Diagram menggambarkan kelakuan objek pada *use case* dengan mendeskripsikan waktu hidup objek dan message yang dikirimkan dan diterima antar objek”. Banyaknya diagram sekuen yang harus digambar adalah minimal sebanyak pendefinisian *use case* yang memiliki proses sendiri atau yang penting semua *use case* yang telah didefinisikan interaksi jalannya pesan sudah dicakup pada diagram sekuen sehingga semakin banyak *use case* yang didefinisikan maka diagram sekuen yang harus dibuat juga semakin banyak.

