BAB II
TINJAUAN PUSTAKA

II.1. Konsep Sistem

Istilah ”sistem” sudah sangat populer pada saat ini. Terminologi ini banyak digunakan untuk mendeskripsikan banyak hal, khususnya bagi aktivitas-aktivitas yang diperlukan didalam pemrosesan data. Usaha-usaha yang telah dilakukan pada masa yang lampau dalam mengaplikasikan teknologi data terfokus pada pengembangan mesin- mesin yang mampu menjalankan suatu operasi data yang lebih efisien (seperti halnya mesin ketik, mesin hitung, file-file penemuan *punched card*, sebagai media penyimpanan data yang dihasilkan di dalam pengembangan berbagai mesin (*keypunch, sorter, collator, printer, dan plotter*), juga menegaskan bahwa pengonversian bentuk data sedemikian rupa sehingga menjadi bentuk informasi adalah bentuk proses. Selain itu, pengembangan komputer digital berikut penggunaan terminologi ”sistem ” dlam memenuhi kebutuhan informasi bagi suatu organisasi yang modern (Eddy Prahasta ; 2009 : 89).

II.1.1. Sistem

Secara umum sistem dapat didefinisikan sebagai sekumpulan objek, ide, berikut saling ketergantungan (inter-relasi) di dalam usaha mencapai suatu tujuan atau dengan kata lain, sistem disebutkan sebagai kumpulan komponen (subsistem fisik maupun non-fisik/logika) yang saling berhubungan satu sama

lainnya dan bekerja sama secara harmonis untuk mencapai suatu tujuan (Eddy Prahasta ; 2009 : 89).

II.1.2. Sistem Informasi

Sistem informasi adalah sekumpulan komponen- komponen yang saling berhubungan dan bekerja sama untuk mengumpulkan, memproses, menyimpan, dan mendistribusikan informasi terkait untuk mendukung proses pengambilan keputusan, koordinasi, dan pengendalian (Eddy Prahasta ; 2009 : 93).

II.1.3. Akuntansi

Akuntansi adalah seni daripada pencatatan, penggolongan dan peringkasan daripada peristiwa-peristiwa dan kejadian-kejadian yang setidaknya sebagian bersifat keuangan dengan cara yang setepa-tepatnya dan dengan penunjuk atau dinyatakan dalam uang, serta penafsiran terhadap hal-hal yang timbul daripadanya (Drs. S. Munawir. Akuntan ; 2010: 05).

II.1.4. Sistem Informasi Akuntansi

Kumpulan beberapa sub sistem yang saling berhubungan satu sama lain dan bekerja sama dengan harmonis dalam mengolah data keuangan sedemikian rupa hingga menghasilkan informasi keuangan bagi (pihak) manajemen untuk membantu pengambilan keputusan dibidang keuangan (Eddy Prahasta ; 2009 : 108).

II.1.5. Penjualan

Penjualan adalah Penjualan berarti proses kegiatan menjual yaitu dari kegiatan penetapan harga jual sampai produk didistribusikan ketangan konsumen (M. Nafarin ; 2007:166).

Penjualan merupakan salah satu sumber pendapatan perusahaan, semakin besar aktivitas penjualan di suatu perusahaan, maka akan semakin besar pula biaya yang akan dikeluarkan oleh perusahaan tersebut.

Dari definisi diatas maka dapat diambil kesimpulan sebagai suatu bentuk perpindahan (transfer) dari penjual kepada pembeli sesuai dengan syarat dan kondisi yang disepakati. Tujuan dari penjualan yaitu menjual apa yang telah dihasilkan dan bukan penghasilan apa yang dapat dijual dalam perusahaan mempunyai tujuan diantaranya sebagai berikut:

1. Mencapai jumlah atau volume tertentu
2. Mendapat laba tertentu
3. Menunjang pertumbuhan perusahaan

Perusahaan yang kegiatan utamanya melakukan penjualan biasanya menyatakan jumlah penjualannya dalam bentuk unit-unit, ini merupakan bagian dari hasil penjualan produk yang terjual dibandingkan dengan produk yang tersedia yang bisa dinyatakan dalam bentuk data numerik atau deretan angka.

II.2. Basis Data (*Database*)

Database atau basis data adalah sekumpulan data yang memiliki hubungan secara logika dan diatur dengan sesunan tertentu serta disimpan dalam media

penyimpanan komputer. Data itu sendiri adalah representasi dari semua fakta yang ada pada dunia nyata. *Database* sering digunakan untuk melakukan proses terhadap data-data tesebut untuk menghasilkan informasi tetentu. misalnya dari data nama siswa yang berulang tahun pada hari ini. Tentu saja informasi tersebut akan anda dapatkan dari s*ofware*/pemroses database dengan cara anda memberikan perintah dalam bahasa tertentu yaitu SQL(*Structured Query Language*).

Pada era kemajuan teknologi seperti sekarang ini, nilai informasi sangatlah penting, terlebih bagi kemajuan perusahaan. Oleh karena itu penggunaan dan penguasaan database sangat penting. Dalam database ada sebutan-sebutan utnuk satuan data yaitu:

1. Karakter, ini adalah satuan data terkecil. data terdiri atau susunan karakter yang pada akhirnya mewakili data yang memiliki arti dari sebuah fakta.
2. Field, adalah kumpulan dari karakter yang mewakili fakta tertentu misalnya seperti jenis bibit, satuan, dan lain-lain. Dalam dunia perancangan database, field juga disebut atribut. Bila dipandang dari sudut pemrogaman berorientasi obyek maka *name* dan properti *type*. Properti *name* atau nama adalah properti dari field yang berisi field yang mewakili data sejenis yang disimpannya. Sedangkan properti type adalah properti yang mengatur tipe data dari data yang akan ditampungnya. Misalnya nama fieldnya adalah jenis bibit maka tipe datanya adalah *char*, bila nama fieldnya adalah tanggal lahir maka tipe datanya adalah date. Field dilihat seperti kolom.

3. Record, adalah kumpulan dari field. Pada *record* anda dapat menemukan banyak sekali informasi penting dengan cara mengombinasikan field-field yang ada.
4. Tabel, adalah sekumpulan dari *record-record* yang memiliki kesamaan entity dalam dunia nyata. Kumpulan dari tabel adalah database, wujud fisik sebuah database dalam komputer adalaha sebuah file yang didalamnya terdapat berbagai tingkatan data yang telah disebutkan di atas.
5. File, adalah bentuk fisik dari penyimpanan data. File database berisi semua data yang telah disusun dan diorganisasikan sedemikian rupa sehingga memudahkan pemberian informasi. (Wahana Komputer : 2010 : 24)

II.3. Entity Relationship Diagram

Pada dasarnya ERD(*Entity Relationship Diagram*) adalah sebuah diagram yang secara konseptual memetakan hubungan antar penyimpanan pada diagram DFD di atas. ERD ini digunakan untuk melakukan permodelan terhadap struktur data dan hubungannya. Penggunaannya ERD ini dilakukan untuk mengurangi tingkat kerumitan penyusunan sebuah database yang baik.

Entity dapat berarti sebuah obyek yang dapat dibedakan dengan obyek lainnya. Obyek tersebut dapat memiliki komponen-komponen data (atribut atau field) yang membuatnya dapat dibedakan dari obyek yang lain. Dalam dunia database *entity* memiliki atribut yang menjelaskan karakteristik dari entity tersebut. Ada dua macam atribut yang di kenal deskriptif. Hal ini berarti setiap entity memiliki himpunan yang diperlukan sebuah primary key untuk membedakan anggota-anggota dalam himpunan tersebut.

Atribut dapat memiliki sifat-sifat sebagai berikut:

1. *Atomic*, atomik adalah sifat dari atribut yang menggambarkan bahwa atribut tesebut berisi nilai yang spesifik dan tidak dapat dipecah lagi. Contoh dari sifat atomik adalah field status dari tabel karyawan yang hanya berisi menikah atau single
2. *Multivalued*, sifat ini menandakan atribut ini bisa memiliki lebh dari satu nilai untuk tiap entity tertentu. Misalnya adalah field hobi, hobi dari tiap karyawan mungkin dan hampir pasti lebih dari satu. Misalnya karyawan A mimiliki hobi membaca , nonton TV dan bersepeda.
3. *Composite,* atribut yang bersifat komposit adalah atribut yang nilainya adalah gabungan dari beberapa atribut yang bersifat atomik. Contohnya adalah atribut alamat yang dapat dipecah menjadi atribut atomik berupa alamt, kode pos, no telepon, dan kota. (Wahana Komputer : 2010 : 30)

Ada beberapa derajat relasi tabg daoat terjadi, yaitu :

1. *One to one*, menggambarkan bahwa antara 1 anggota entity A hanya dapat berhubungan dengna 1 anggota entity B. Biasanya derajat relasi ini digambarkan dengna simbol 1-1.
2. *One to many*, menggambarkan bahwa 1 anggota entity A dapat memiliki hubungan dengan lebih dari 1 angota entity B. Biasanya derajat relasi ini digambarkan denga simbol 1-N.
3. *Many to many*, menggambarkan bahwa lebih dari satu anggota A dapat memiliki hubungan dengan lebih dari satu angota entity B. Simbol yang digunakan adalah N-N. (Wahana Komputer : 2010 : 31).

II.3.1. Normalisasi

Setelah melalui tahapan di atas atau ERD, maka hasil pada diagram tersebut mulai direlasasikan pada tabel-tabel database. Untuk itu dperlukan sebuah tahapan yang disebut normalisasi. Normalisasi data adalah proses di mana tabel-tabel pada database dites dalam hal kesalingtergantungan di antara field-field pada sebuah tabel. Misalnya jika pada sebuah tabel terdapat ketergantungan terhadap lebih dari satu field dalam tabel tersebut, maka tabel tersebut harus dipecah menjadi banyak tabel.

Pada prose normaslisasi data, aturan yang dijadikan acuan adalah metode ketergantungan fungsional. Teorinya adalah bahwa tiap kolom dalam sebuah tabel selalu memiliki hubungan yang unik dengan sebuah kolom kunci. Misalnya pada sebuah tabel data_bibit ada field kode bibit data field jenis bibit serta field satuan. Maka ketergantungan fungsionalnya dapat dinyatakan sebagai berikut: kd_bibit -> jns_bibit dan kd_bibit -> satuan. Artinya jns_bibit memiliki ketergantungan fungsional terhadap kd_bibit. Field jns_bibit isinya juga ditentukan oleh field kd_bibit. Maksud dari semua itu adalah kd_bibit adalah field kunci yang menentukan karena tidak ada kode bibit yang sama pada satu perusahaan, jadi field kd_bibit dapat dijadikan patokan untuk mengisi jns_bibit dan field lainnya.

Ada beberapa langkah dalam normalisasi tabel, yaitu:

1. *Decomposition*, dekomposisi adalah proses mengubah bentuk tabel supaya memenuhi syarat tertentu sebagai tabel yang baik. Dekomposisi dapat dikatakan berhasi jika tabel yang dikenal dekomposisi bila digabungkan kembali dapat menjadi tabel awal sebelum di –dekomposisi. Dekomposisi

akan sering dilakukan dalam proses normalisasi untuk memenuhi syarat-syaratnya

2. Bentuk tidak normal, pada bentuk ini semua data yang ada pada tiap entity (diambil atributnya) masih ditampung dalam satu tabel besar. Data yang ada pada tabel ini masih ada yang redundansi dan ada juga yang kosong. Semuanya masih tidak tertata rapi.
3. Normal Form pertama(1^{st} Normal Form), pada tahapan ini tabel di-dekomposisi dari tabel bentuk tidak normal yang kemudian dipisahkan menjadi tabel-tabel kecil yang memiliki kriteria tidak memiliki atribut yang bernilai ganda dan komposit. Semua atribut harus bersifat atomik.
4. Normal Form kedua(2^{nd}Normal Form), pada tahapan ini tabel dianggap memenuhi normal kedua jika pada tabel tersebut semua atribut yang bukan kunci primer bergantung penuh terhadap kunci primer tabel tersebut .
5. Normal Form ketiga(3^{rd}Normal Form), setiap atribut pada tabel selain kunci primer atau kunci utama harus bergantung penuh pada kunci utama. Bentuk normal ketiga biasanya digunakan bila masih ada tabel yang belum efisien. Biasanya penggunaan bentuk normal(normalisasi) hanya sampai pada bentuk ketiga, dan tabel yang dihasilkan telah memiliki kualitas untuk membentuk sebuah database yang dapat diandalkan. Semua tabel diatas juga telah memenuhi bentuk normal tahap ketiga. (Wahana Komputer : 2010 : 32)

II.4. Pemrograman Visual Basic 2010

Visual basic 2010 merupakan salah satu bagian dari produk pemograman terbaru yang dikeluarkan oleh Microsoft, sebagai produk lingkungan pengembangan terintegritasi atau IDE andalan yang dikeluarkan. Visual basic 2010 menambahkan perbaikan – perbaikan filtur dan filtur baru yang lebih lengkap dibandingkan versi visual studio pendahulunya yaitu Vb 2008.

Visual studio merupakan produk pemograman andalan dari Microsoft corporation, yang didalamnya berisi beberapa jenis IDE pemrograman seperti visual basic , visual C ++, visual Web developer, Visual C#, dan Visual f#. semua IDE pemrograman tersebut sudah mendukung penuh implementasi. Net Framework terbaru, yaitu Net Framework 4.0 yang merupakan pengembangan dari Net Framework 3.5. adapun database standar yang disertakan adalah Microsoft SQL Server 2008 Express (Andi Yogyakarta : 2010 : 2).

II.5. SQL Server 2008

SQL Server 2008 adalah sebuah terobosan baru dari Microsoft dalam bidang database. SQL Server adalah sebuah *DBMS (Database Management System)* Yang dibuat oleh *Microsof*t untuk ikut berkecimpung dalam persaingan dunia pengolahan data menyusul pendahulunya seperti IBM dan *Oracle. Sql server 2008* dibuat pada saat kemajuan dalam bidang hardware sedemikian pesat. Oleh karena itu sudah dapat dipastikan bahwa *SQL Server 2008* membawa

beberapa terobosan dalam bidang pengolahan dan penyimpanan data (Wahana Komputer ; 2010 : 5).

SQL server 2008 termasuk salah satu mesin database relasional. Ide tentang sistem database relasional pertama kali dicetuskan oleh seorang yang bernama E.F. Codd lewat sebuah artikel yang berjudul " *A Relation Model of Data for Large Shared Data Banks"*. Artikel tersebut ditulis pada tahun 1970. Sistem database relational berdasarkan pada metode matematika. Sistem ini menggantikan metode lama yang berdasarkan *network* dan *hierarki*. Metode relasional ini bekerja berdasarkan keterkaitan antartabel (Wahana Komputer ; 2010 : 25).

II.6. UML

Unified Modeling Language (UML) adalah keluarga notasi grafis yang didukung oleh meta-model tunggal, yang membandu pendeskripsian dan desain sistem perangkat lunak, khususnya sistem yang dibangun menggunakan pemrograman berorientasi objek (OOP).

Bahasa pemodelan grafis telah ada di industru perangkat lunak sejak lama. Pemicu umum dibalik semuanya adalah bahwa bahasa pemrograman berada pada tingkat abstraksi yang tidak terlalu tinggi untuk memfasilitasi diskusi tentang desain.

UML merupakan standar yang relatif terbuka yang dikontrol oleh *Object Management Group* (OMG), sebuah konsorium terbuka yang terdiri dari banyak perusahaan. OMG dibentuk untuk membuat standar-standar yang mendukung

interoperabilitas, khususnya interoperabilitas sistem yang berorientasi objek. OMG mungkin lebih dikenal dengan standar-standar CORBA (*Common Object Request Broker Architecture*).

UML lahir dari penggabungan banyak bahasa permodelan grafis berorientasi objek yang berkembang pesat pada akhir 1980-an dan awal 1990-an. Sejak kehadirannya pada tahun 1997, UML menghancurkan menara Babel tersebut menjadi sejarah. (Prabowo Pudjo Widodo : 2011 : 6-9)

II.6.1. Diagram Dasar Dalam UML

Terdapat sembilan jenis diagram UML, namun Penulis akan menjabarkan empat jenis diantaranya :

1. *Class Diagram*

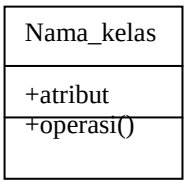
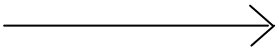
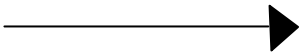
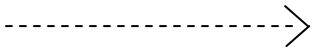
   Bersifat statis. Menggambarkan struktur object sistem. Diagram ini menunjukkan *class object* yang menyusun sistem dan juga hubungan antara *class object* tersebut. Kelas memiliki apa yang disebut atribut dan metode atau operasi.

   a. Atribut merupakan *varabel-variabel* yang dimiliki oleh suatu kelas.
   b. Operasi atau metode adalah fungsi-fungsi yang dimiliki oleh suatu kelas.

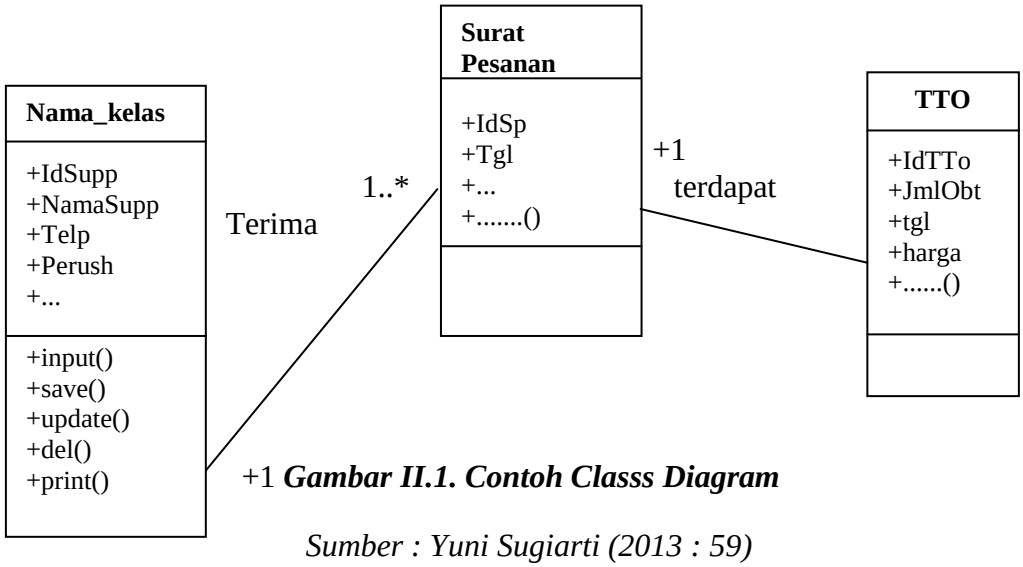
   Berikut adalah simbol-simbol yang ada pada diagram kelas :

Tabel II.1. Simbol-simbol Class Diagram

Simbol	Deskripsi

Kelas 	Kelas pada struktur sistem
Antarmuka / Interface Nama_interface 	Sama dengan konsep interface dalam pemrograman berorientasi objek
asosiasi / association	Relasi antar kelas dengan makna umum, asosiasi biasanya juga disertai dengan *multiplicity*.
Asosiasi berarah / directed association 	Relasi antar kelas dengan makna kelas yang satu digunakan oleh kelas yang lain, asosiasi biasanya juga disertai dengan *multiplicity.*
Generalisasi 	Relasi antar kelas dengan makna generalisasi-spesialisasi (umum khusus).
Kebergantungan / *dependency* 	Relasi antar kelas dengan makna kebergantungan antar kelas.
Agregasi / *aggregation*	Relasi antar kelas dengan makna

Sumber : Yuni Sugiarti (2013 : 59)



+1 ***Gambar II.1. Contoh Classs Diagram***

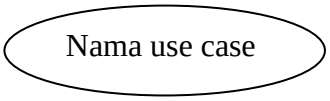
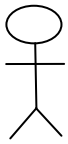
Sumber : Yuni Sugiarti (2013 : 59)

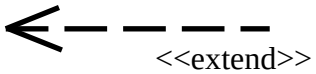
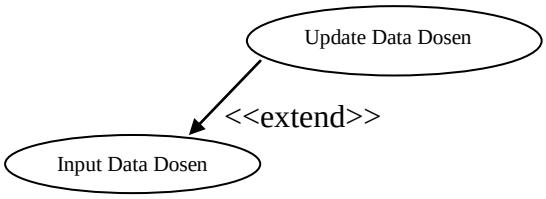
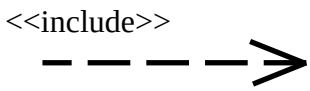
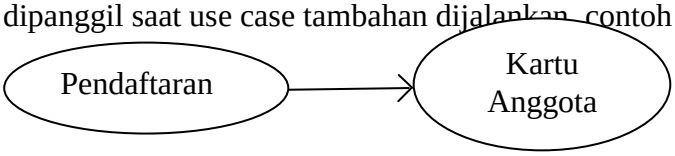
2. Diagram *Use-Case*

Use case diagram secara grafis menggambarkan interaksi antara sistem, sistem *eksternal,* dan pengguna. Dengan kata lain *Use Case* diagram secara grafis mendeskripsikan siapa yang akan menggunakan sistem dan dalam cara apa pengguna *(user)* mengharapkan interaksi dengan sistem itu. *Use Case* secara naratif digunakan untuk secara tekstual menggambarkan sekuensi langkah-langkah dari setiap interaksi. Diagram ini memperlihatkan himpunan *use-case* dan aktor-aktor (suatu jenis khusus dari kelas).

Berikut adalah simbol-simbol yang ada pada diagram Use Case :

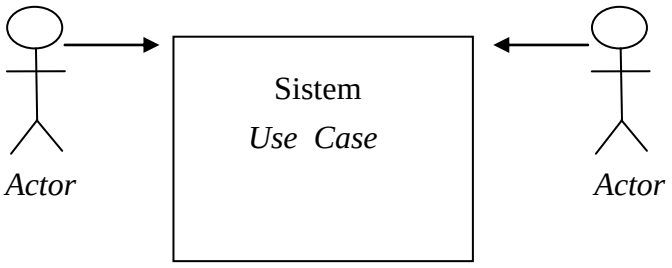
Tabel II.2. Simbol-simbol Use Case

Simbol	Deskripsi
Use Case 	Fungsionalitas yang disediakan sistem sebagai unit-unit yang saling bertukar pesan antar unit atau aktor; biasanya dinyatakan dengan menggunakan kata kerja diawal frase nama use case.
Actor  *Nama*	Orang, proses, atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat di luar sistem informasi yang akan dibuat itu sendiri, jadi walaupun simbol dari actor adalah gambar orang , tapi aktor belum tentu merupakan orang, biasanya dinyatakan meggunakan kata benda di awal frase nama aktor.
asosiasi / association _____	Komunikasi antara aktor dan use case yang berpartisipasi pada use case atau use case memiliki interaksi dengan aktor.

Extend  	Relasi use case tambahan ke sebuah use case dimana use case yang ditambahkan dapat berdiri sendiri walau tanpa use case tambahan itu; mirip dengan prinsip inheritance pada pemrograman berorientasi objek; biasanya use case tambahan memiliki nama depan yang sama dengan use case yang ditambahkan, arah panah menunjuk pada use case yang dituju. 
Include  	Relasi use case tambahan ke sebuah use case dimana use case yang ditambahkan memerlukan use case ini untuk menjalankan fungsinya atau sebagai syarat dijalankan use case ini. Ada dua sudut pandang yang cukup besar mengenai include di use case, include berarti use case yang ditambahkan akan selalu dipanggil saat use case tambahan dijalankan, contoh : 

Sumber : Yuni Sugiarti (2013 : 42)

Berikut contoh gambar use diagram :

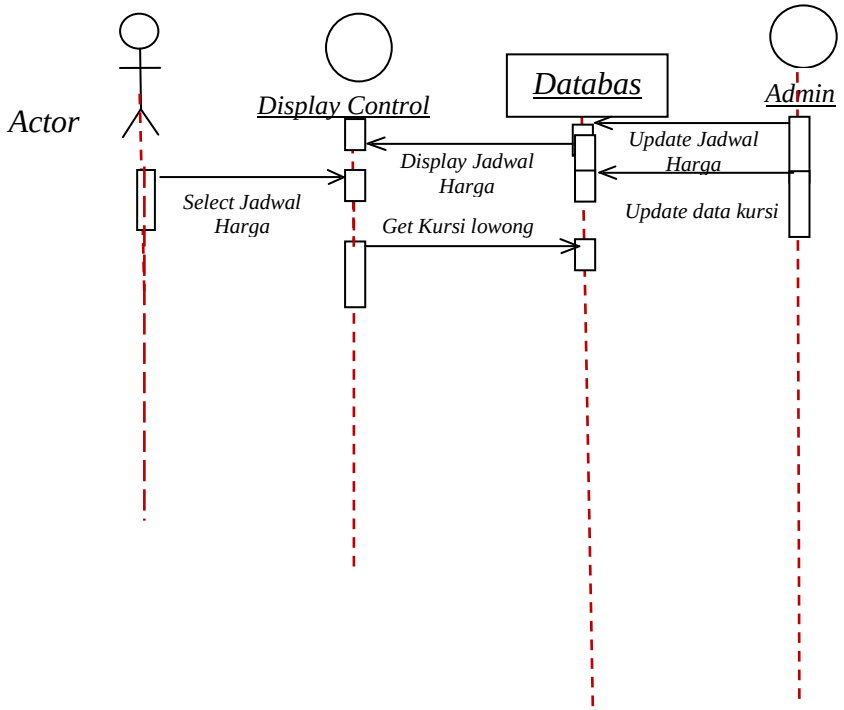


Gambar II.2. Use Case Model

Sumber : Yuni Sugiarti (2013 : 56)

3. Diagram interaksi dan *sequence* (urutan)

Diagram *sequence* meggambarkan kelakuan/prilaku objek pada *use case* dengan mendeskripsikan waktu hidup objek dan *message* yang dikirimkan dan diterima antar objek. Oleh karena itu untuk menggambar diagram *sequences* maka harus diketahui objek-objek yang terlibat dalam sebuah *use case* beserta metode-metode yang dimiliki kelas yang diinstansi menjadi objek itu.



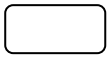
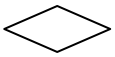
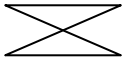
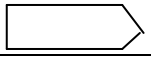
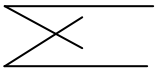
Gambar II.3. Display Sequence Diagram

Sumber : Yuni Sugiarti (2013 : 59)

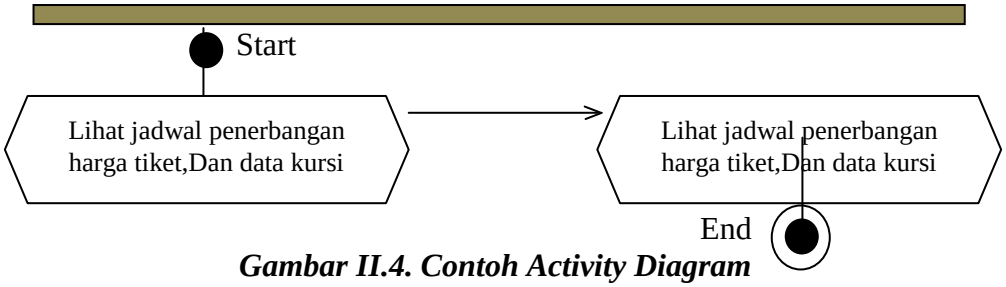
4. Diagram Aktifitas (*Activity Diagram*)

Bersifat dinamis. Diagra aktifitas adalah tipe khusus dari diagram status yang memperlihatkan aliran dari suatu aktifitas ke aktifitas lainnya dalam suatu sistem. Berikut adalah contoh *Activity diagram.*

Tabel II.3. Simbol-simbol Activity Diagram

Simbol	Keterangan
	Titik awal
	Titik Akhir
	Activity
	Pilihan untuk pengambilan keputusan
	Fork ; digunakan untuk menunjukkan kegiatan yang dilakukan secara paralel atau untuk menggabungkan dua kegiatan paralel menjadi satu.
	Rake; menunjukkan adanya dekomposisi
	Tanda Waktu
	Tanda Pengiriman
	Tanda Penerimaan
	Aliran akhir (Flow Final)

Berikut contoh dari Activity Diagram :



Gambar II.4. Contoh Activity Diagram

Sumber : Yuni Sugiarti (2013 : 78)