

BAB II

LANDASAN TEORI

II.1. Pengertian Sistem

Sistem merupakan kumpulan dari unsur atau elemen-elemen yang saling berkaitan atau berinteraksi dan saling memengaruhi dalam melakukan kegiatan bersama untuk mencapai suatu tujuan tertentu. Karakteristik sistem terdiri dari :

1. Komponen (*Component*)

Suatu sistem terdiri dari sejumlah komponen yang saling berinteraksi, bekerja sama membentuk suatu kesatuan. Komponen-komponen sistem dapat berupa suatu subsistem atau bagian-bagian dari sistem.

2. Batas Sistem (*Boundary*)

Batasan merupakan daerah yang membatasi antara suatu sistem dengan sistem yang lainnya atau dengan lingkungan luarnya. Batasan sistem ini memungkinkan suatu sistem dipandang suatu kesatuan, karena dengan batas sistem ini fungsi dan tugas dari subsistem yang satu dengan lainnya berbeda tetapi tetap saling berinteraksi. Batas suatu sistem menunjukkan ruang lingkup (*scope*) dari sistem tersebut.

3. Lingkungan Luar Sistem (*Environment*)

Environment merupakan segala sesuatu di luar batas sistem yang memengaruhi operasi dari suatu sistem. Lingkungan luar sistem dapat bersifat menguntungkan atau merugikan.

4. Penghubung Sistem (*Interface*)

Merupakan media penghubung antara satu subsistem dengan subsistem yang lainnya untuk membentuk satu kesatuan sehingga sumber-sumber daya mengalir dari subsistem yang satu ke subsistem yang lainnya. Dengan kata lain, *output* dari suatu subsistem akan menjadi *input* dari subsistem yang lainnya.

5. Masukan Sistem (*Input*)

Merupakan energi yang di masukan ke dalam sistem. Masukan dapat berupa masukan perawatan (*maintenance input*) adalah energi yang di masukkan supaya sistem tersebut dapat beroperasi. Masukan sinyal (*signal input*) adalah energi yang diproses untuk mendapatkan keluaran.

6. Keluaran Sistem (*Output*)

Merupakan hasil dari energi yang diolah oleh sistem, meliputi *output* yang berguna.

7. Pengolahan Sistem (*Process*)

Merupakan bagian yang memproses masukan untuk menjadi keluaran yang diinginkan.

8. Tujuan Sistem (*Goal*)

Suatu sistem pasti mempunyai tujuan ataupun sasaran yang memengaruhi *input* yang dibutuhkan dan *output* yang dihasilkan. Dengan kata lain, suatu sistem akan dikatakan berhasil kalau pengoperasian sistem itu mengenai sasaran atau tujuannya. (Asbon Hendra : 2012 ; 157-160)

II.1.2. Klasifikasi Sistem

Suatu sistem dapat diklasifikasikan menjadi sebagai berikut dan didefinisikan :

1. Sistem Abstrak (*Abstract System*)

Sistem yang berupa pemikiran atau ide-ide yang tidak tampak secara fisik, sedangkan sistem fisik merupakan sistem yang ada secara fisik.

2. Sistem fisik (*Physical System*)

Merupakan sistem yang ada secara fisik sehingga setiap makhluk dapat melihatnya.

3. Sistem Alamiah (*Natural System*)

Sistem yang terjadi melalui proses alam, dalam artian tidak dibuat oleh manusia.

4. Sistem Buatan Manusia (*Human Made System*)

Sistem yang dirancang oleh manusia. Sistem buatan manusia yang melibatkan interaksi manusia dengan mesin disebut *human machine system*.

5. Sistem Tertentu (*deterministic System*)

Beroperasi dengan tingkah laku yang sudah dapat diprediksi. Interaksi bagian-bagiannya dapat dideteksi dengan pasti sehingga keluaran dari sistem dapat diramalkan.

6. Sistem Tak Tentu (*Probabilistic System*)

Sistem yang kondisi masa depannya tidak dapat diprediksi karena mengandung unsur probabilitas.

7. Sistem Tertutup (*Closed System*)

Sistem yang tidak berhubungan dan tidak terpengaruh dengan sistem luarnya. Sistem ini bekerja secara otomatis tanpa adanya turut campur tangan dari pihak luarnya.

8. Sistem Terbuka (*Open System*)

Sistem yang berhubungan dan terpengaruh dengan lingkungan luarnya. Lebih spesifik dikenal juga yang disebut dengan sistem terotomasi, yang merupakan bagian dari sistem buatan manusia dan berinteraksi dengan kontrol oleh satu atau lebih computer sebagai bagian dari sistem yang digunakan dalam masyarakat modern. (Asbon Hendra : 2012 ; 160-162)

II.2. Pengertian Sistem Informasi

Sistem informasi dapat diartikan sebagai suatu sistem terintegrasi atau sistem manusia-mesin, untuk menyediakan informasi untuk mendukung operasi, manajemen dalam suatu organisasi. (Asbon Hendra : 2012 ; 168)

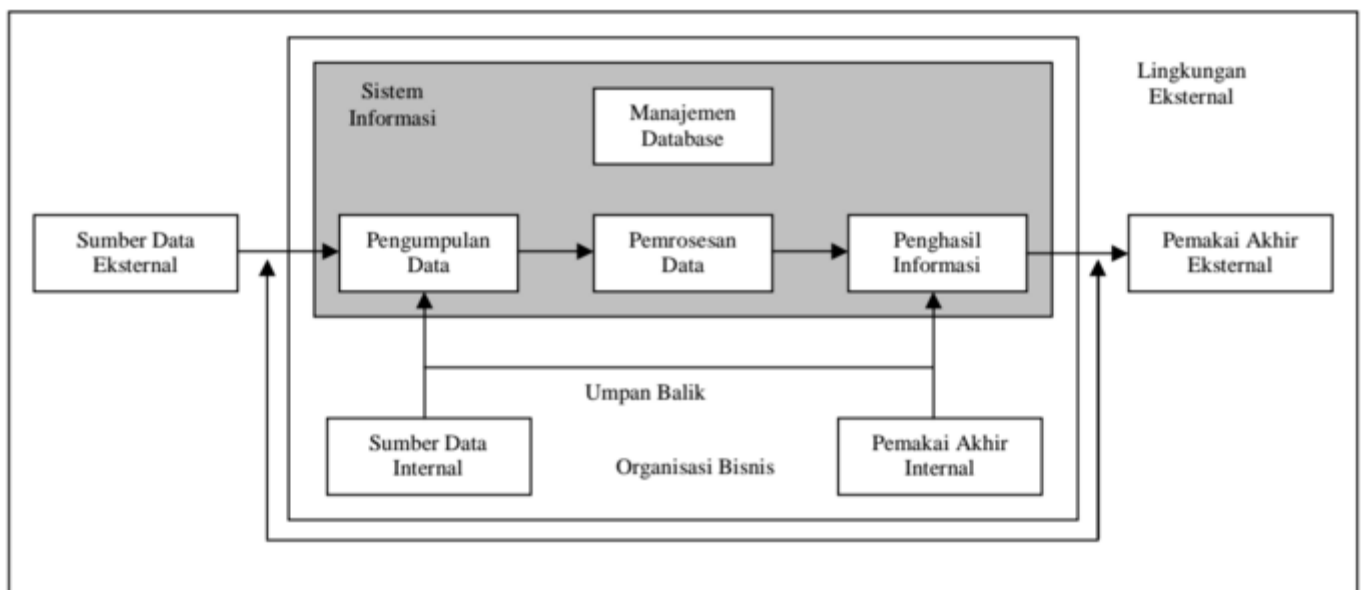
Dalam suatu sistem informasi terdapat komponen - komponen sebagai berikut :

1. Perangkat keras komputer : CPU, *storage*, perangkat *input/output*, terminal untuk interaksi, serta media komunikasi data.
2. Perangkat lunak komputer : perangkat lunak sistem (sistem operasi dan utility-nya), perangkat lunak umum aplikasi (bahasa pemrograman), dan perangkat lunak aplikasi (aplikasi akuntansi)
3. Basis data : penyimpanan data pada media penyimpanan komputer.

4. Prosedur : langkah-langkah penggunaan sistem
5. Personel untuk penelolan operasi (SDM).

II.2.1. Arsitektur Sistem Informasi Akuntansi

Arsitektur sistem informasi akuntansi merupakan kerangka model umum yang menggambarkan semua sistem yang digunakan dalam pembelajaran SIA dari mulai sumber data, proses pengumpulan, pemrosesan dan penyajian hasil informasi baik di lingkungan internal maupun eksternal perusahaan, yang dapat kami paparkan sebagai berikut :



Gambar II.1. Arsitektur Sistem Informasi Akuntansi
(Sumber : Ebook aris koerniawan 2012 : 8)

II.3. Kebangkrutan

Istilah “*pailit*” dijumpai dalam perbendaharaan bahasa Belanda, Perancis, Latin dan Inggris. Dalam bahasa Perancis, istilah “*failite*” artinya pemogokan atau

kemacetan dalam melakukan pembayaran. Orang yang mogok atau macet atau berhenti membayar hutangnya disebut dengan *Le falli*. Di dalam bahasa Belanda dipergunakan istilah *faillit* yang mempunyai arti ganda yaitu sebagai kata benda dan kata sifat. Sedangkan dalam bahasa Inggris dipergunakan istilah *to fail*, dan di dalam bahasa Latin dipergunakan istilah *failire*. Di negara-negara yang berbahasa Inggris, untuk pengertian pailit dan kepailitan dipergunakan istilah “*bankrupt*” dan “*bankruptcy*”.

Kesulitan keuangan jangka pendek bersifat sementara dan belum begitu parah. Tetapi kesulitan semacam itu apabila tidak ditangani bisa berkembang menjadi kesulitan tidak solvabel (hutang lebih besar dibanding aset). Kalau tidak solvabel, perusahaan bisa dilikuidasi atau direorganisasi. Likuidasi dipilih apabila nilai likuidasi lebih besar dibandingkan dengan nilai perusahaan kalau diteruskan. Reorganisasi dipilih kalau perusahaan masih menunjukkan prospek dan dengan demikian nilai perusahaan kalau diteruskan lebih besar dibandingkan nilai perusahaan kalau dilikuidasi. (Siti Khairani : 2013; 3)

II.4. Metode Z – Score Altman

Penelitian yang dilakukan oleh Edward I, Altman yaitu mencari kesamaan rasio keuangan yang biasa di pakai untuk memprediksi kebangkrutan untuk semua negara studinya. Analisis kebangkrutan Z adalah suatu alat yang digunakan untuk meramalkan tingkat kebangkrutan suatu perusahaan dengan menghitung nilai dari beberapa rasio lalu kemudian dimasukkan dalam suatu persamaan diskriminasi. Altman telah mengkombinasikan beberapa rasio menjadi model prediksi dengan

teknik statistik yaitu analisis diskriminan yang digunakan untuk memprediksi terjadinya kebangkrutan perusahaan dengan istilah *Z-Score*. *Z-Score* merupakan *score* yang ditentukan dari hitungan standar yang akan menunjukkan tingkat kemungkinan kebangkrutan perusahaan. Formula *Z-Score* untuk memprediksi kebangkrutan dari Altman merupakan sebuah *multivariate formula* yang digunakan untuk mengukur kesehatan finansial dari sebuah perusahaan.

Altman menemukan lima jenis rasio keuangan yang dapat dikombinasikan untuk melihat perbedaan antara perusahaan yang bangkrut dan yang tidak bangkrut. Langkah proses perhitungan Altman *Z-Score* ditentukan dengan menggunakan rumus sebagai berikut:

$$\mathbf{Z-Score} = 1,2 \mathbf{x}_1 + 1,4 \mathbf{x}_2 + 3,3 \mathbf{x}_3 + 0,6 \mathbf{x}_4 + 1,0 \mathbf{x}_5 \text{ (Altman, 1968:594)..(1)}$$

Keterangan :

\mathbf{x}_1 = Modal kerja terhadap Total Aktiva (*Net Working Capital to Total Assets*)

\mathbf{x}_2 = Rasio laba ditahan terhadap Total Aktiva (*Retained Earnings to Total Assets*)

\mathbf{x}_3 = Pendapatan sebelum pajak dan bunga terhadap Total Aktiva (*Earnings Before Interest and Taxes to Total Assets*)

\mathbf{x}_4 = Nilai pasar ekuitas terhadap nilai buku dari hutang (*market value equity to book value of total debt*).

\mathbf{x}_5 = Penjualan terhadap Total Aktiva (*Sales to Total Asset*). (Siti Khairani : 2013; 3).

Adapun rasio-rasio yang di miliki Altman Z-Score yaitu :

1. *Net Working Capital to Total Asset*

Rasio ini menunjukkan kemampuan perusahaan untuk menghasilkan modal kerja bersih dari keseluruhan total aktiva yang dimilikinya. Rasio ini dihitung dengan membagi modal kerja bersih dengan total aktiva. Modal kerja bersih diperoleh dengan cara aktiva lancar dikurangi dengan kewajiban lancar. Modal kerja bersih yang negatif kemungkinan besar akan menghadapi masalah dalam menutupi kewajiban jangka pendeknya karena tidak tersedianya aktiva lancar yang cukup untuk menutupi kewajiban tersebut. Sebaliknya, perusahaan dengan modal kerja bersih yang bernilai positif jarang sekali menghadapi kesulitan dalam melunasi kewajibannya.

2. *Retained Earnings to Total Assets*

Rasio ini mengukur keuntungan kumulatif terhadap umur perusahaan yang menunjukkan kekuatan pendapatan. Rasio ini menunjukkan kemampuan perusahaan untuk menghasilkan laba ditahan dari total aktiva perusahaan. Laba ditahan merupakan laba yang tidak dibagikan kepada para pemegang saham. Dengan kata lain, laba ditahan menunjukkan berapa banyak pendapatan perusahaan yang tidak dibayarkan dalam bentuk dividen kepada para pemegang saham.

3. *Earnings Before Interest and Tax to Total Asset*

Rasio ini menunjukan kemampuan perusahaan untuk menghasilkan laba dari aktiva perusahaan, sebelum pembayaran bunga dan pajak.

4. *Market Value of Equity to Book Value of Debt*

Rasio ini menunjukkan kemampuan perusahaan untuk memenuhi kewajiban-kewajiban dari nilai pasar modal sendiri (saham biasa). Nilai pasar ekuitas sendiri diperoleh dengan mengalikan jumlah lembar saham biasa yang beredar dengan harga pasar per lembar saham biasa. Nilai buku hutang diperoleh dengan menjumlahkan kewajiban lancar dengan kewajiban jangka panjang.

5. *Sales to Total Asset*

Rasio ini menunjukkan apakah perusahaan menghasilkan *volume* bisnis yang cukup dibandingkan investasi dalam total aktivasnya. Rasio ini mencerminkan efisiensi manajemen dalam menggunakan keseluruhan aktiva perusahaan untuk menghasilkan penjualan dan mendapatkan laba. Dengan kriteria penilaian apabila nilai $S \geq 2,99$ maka perusahaan tidak mempunyai masalah keuangan yang serius (tidak bangkrut), apabila $1,81 < Z < 2,99$ maka perusahaan akan mengalami permasalahan keuangan jika tidak melakukan perbaikan yang berarti dalam manajemen maupun struktur keuangan (daerah rawan), dan apabila $Z \leq 1,81$ maka perusahaan mengalami masalah keuangan yang serius (bangkrut). (Siti Khairani : 2013;3-4)

II.5. Microsoft Visual Basic 2010

Visual Basic 2010 adalah inkarnasi dari bahasa Visual Basic yang sangat populer dan telah dilengkapi dengan fitur serta fungsi yang setara dengan bahasa

tingkat tinggi lainnya seperti C++. Anda dapat menggunakan Visual Basic 2010 untuk membuat aplikasi Windows, mobile, Web, dan Office yang kompleks dengan menggunakan kode yang Anda tulis, atau kode yang telah ditulis oleh orang lain dan kemudian dimasukkan kedalam program Anda. (Christopher Lee : 2014 ; 1).

II.6. *SQL Server*

SQL Server 2008 adalah teknologi yang mendukung *development* dan administrasi dari *Business Intellegence (BI) Aplication*. *SQL Server Reporting and Integration* servis adalah element dari BI, tapi inti dari BI tersebut adalah *SQL Server 2008 Analysis Servis (SSAS)*. *SQL Server 2008* adalah sebuah DBMS (*Database Management System*) yang dibuat oleh *Microsoft* untuk ikut berkecimpung dalam persaingan dunia pengolahan data menyusul pendahulunya seperti IBM dan Oracle.(GI MDP : 2011 ; 2)

II.7. *Basis Data (Database)*

Basis data (*Database*) adalah kumpulan informasi yang disimpan didalam komputer secara sistematis, sehingga dapat digunakan oleh suatu program komputer untuk memperoleh informasi dari basis data tersebut. Basis data adalah sekumpulan data yang terhubung satu sama lain secara logika dan suatu deskripsi data yang dirancang untuk memenuhi kebutuhan informasi suatu organisasi. (Verdi Yasin : 2012 ; 274)

II.8. Normalisasi

“Normalisasi (*normalization*) adalah suatu proses formal untuk menghapus *field-field* data yang berulang (*redundant*) sambil tetap menjaga kemampuan basis data untuk menambah, mengubah dan menghapus tanpa menyebabkan kesalahan”. (Raymond McLeod Jr dan George P. Schell, 2008:160)

Normalisasi adalah suatu teknik untuk menghasilkan kumpulan relasi dengan properti yang diinginkan untuk mengetahui kebutuhan data dalam organisasi. Tujuan dari normalisasi adalah untuk menghilangkan kerangkapan data, untuk mengurangi kompleksitas dan untuk mempermudah modifikasi data. *Unnormalized Form* (UNF) adalah tabel yang berisi satu atau lebih *repeating group*. Dari bentuk inilah dilakukan proses normalisasi.

Bentuk normalisasi dapat di jelaskan sebagai berikut:

1. Bentuk tidak normal

Bentuk ini merupakan kumpulan data yang akan disimpan, tidak ada keharusan mengikuti suatu format tertentu, dapat saja data tidak lengkap atau terduplikasi dan data dikumpulkan apa adanya.

2. Bentuk normal pertama (1NF)

Suatu tabel dikatakan dalam bentuk normal pertama (1NF) bila setiap kolom bernilai tunggal untuk setiap baris. Ini berarti bahwa nama kolom yang berulang cukup diwakili oleh sebuah nama kolom (tidak perlu ada indeks dalam member nama kolom).

3. Bentuk normal kedua (2NF)

Suatu tabel berada dalam bentuk normal kedua (2NF) jika tabel berada dalam bentuk normal pertama, semua kolom bukan kunci primer dan tergantung sepenuhnya terhadap kunci primer. Suatu kolom disebut tergantung sepenuhnya terhadap kunci primer jika nilai pada suatu kolom selalu bernilai sama untuk suatu nilai kunci primer yang sama.

4. Bentuk normal ketiga (3NF)

Suatu tabel berada dalam bentuk normal ketiga (3NF) jika tabel berada dalam bentuk normal kedua dan semua atribut bukan primer tidak punya hubungan teransitif. Dengan kata lain setiap atribut bukan kunci haruslah bergantung hanya pada *primary key* dan pada *primary key* secara menyeluruh. (Verdi Yasin : 2012 ; 278)

II.9. *Unified Modelling Language (UML)*

Pengembangan sistem adalah aktivitas manusia. Tanpa adanya kemudahan untuk memahami sistem notasi, proses pengembangan kemungkinan besar akan mengalami kesalahan. UML adalah sistem notasi yang sudah dibakukan di dunia pengembangan sistem, hasil kerjasama dari Grady Booch, James Rumbaugh dan Ivar Jacobson. UML yang terdiri dari serangkaian diagram memungkinkan bagi sistem analis untuk membuat cetak biru sistem yang komperhesif kepada klien, programmer dan tiap orang yang terlibat dalam proses pengembangan tersebut. Dengan UML akan dapat menceritakan apa yang seharusnya dilakukan oleh sistem bukan bagaimana yang seharusnya dilakukan oleh sebuah sistem.

UML adalah bahasa untuk menspesifikasi, memvisualisasi, membangun dan mendokumentasikan bagian dari informasi yang digunakan atau dihasilkan oleh proses pembuatan perangkat lunak, seperti pada pemodelan bisnis dan sistem non perangkat lunak lainnya. UML tidak hanya merupakan sebuah bahasa pemrograman visual saja, namun juga dapat secara langsung dihubungkan ke berbagai bahasa pemrograman seperti JAVA, C++, Visual Basic atau bahkan dihubungkan secara langsung kedalam sebuah *object oriented database*. Begitu juga mengenai pendokumentasian dapat dilakukan seperti *requirements*, arsitektur, *design*, *source*, *project plan*, *tests* dan *prototypes*. UML memiliki 8 tipe diagram, namun pada penulisan skripsi ini penulis akan menggunakan 4 tipe diagram UML yaitu *use case diagram*, *sequence diagram*, *activity diagram* dan *class diagram*.

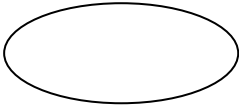
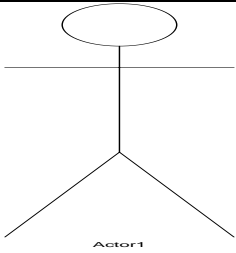


II.9.1. Tipe Diagram UML

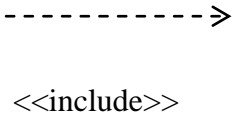
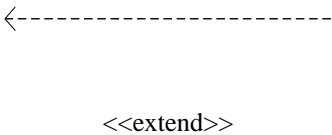
Adapun 8 tipe diagram UML adalah :

1. Use Case Diagram

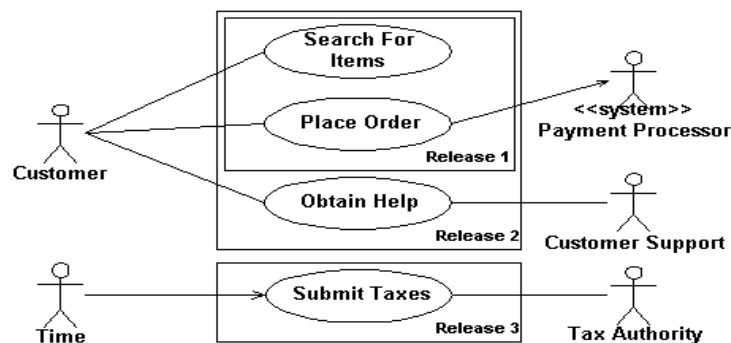
Use Case Diagram merupakan permodelan untuk kelakuan (behavior) sistem informasi yang akan dibuat. *Use Case* mendeskripsikan sebuah interaksi antara satu atau lebih aktor dengan sistem informasi yang akan dibuat. Dapat dikatakan *Use Case* digunakan untuk mengetahui fungsi apa saja yang ada di dalam sistem informasi dan siapa saja yang berhak menggunakan fungsi-fungsi tersebut.

Simbol-simbol yang digunakan dalam *Use Case Diagram* yaitu:

Gambar	Keterangan
	<p>Use Case menggambarkan fungsional yang disediakan sistem sebagai unit-unit yang bertukar pesan antar unit dengan aktor, biasanya dinyatakan dengan menggunakan kata kerja di awal nama Use Case.</p>
	<p>Actor atau aktor adalah abstraction dari orang atau sistem yang lain yang mengaktifkan fungsi dari target sistem. Perlu dicatat bahwa aktor berinteraksi dengan use case, tetapi tidak memiliki kontrol terhadap use case.</p>
	<p>Asosiasi antara aktor dan use case, di gambarkan dngan garis tanpa panah yang mengindikasikan siapa atau apa yang meminta interaksi secara langsung dan bukannya mengindikasikan aliran data.</p>
	<p>Asosiasi antara aktor dan usecase yang menggunakan panah terbuka untuk mengindikasikan bila aktor berinteraksi secara pasif dengan sistem.</p>

	<p>Include, merupakan di dalam use case lain (required) atau pemanggilan use case oleh use case lain, contohnya adalah pemanggilan sebuah fungsi program.</p>
	<p>Extend merupakan perluasan dari use case lain kondisi atau syarat terpenuhi.</p>



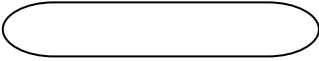
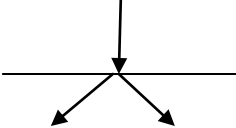
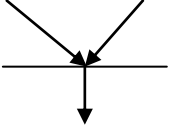
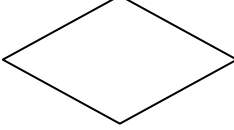
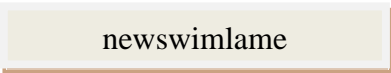
Gambar II.2. Simbol Use Case Diagram
(Sumber : Windu Gata ; 2013 : 4)



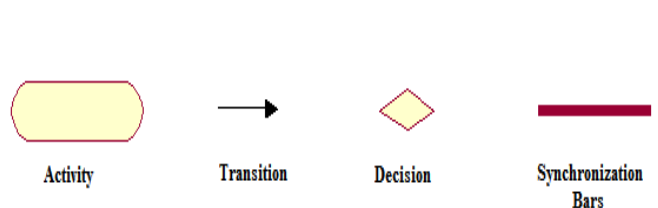
Gambar II.3. Notasi Use Case Diagram
(Sumber : Havaluddin ; 2011 : 4)

2. Diagram Aktivitas (*Activity Diagram*)

Activity diagram menggambarkan *workflow* (aliran kerja) atau aktivitas dari sebuah sistem atau proses bisnis. Simbol-simbol yang digunakan dalam *activity diagram*, yaitu:

Gambar	Keterangan
	Start point diletakkan pada pojok kiri atas dan merupakan awal aktifitas.
	End point, akhir aktifitas
	Activities, menggambarkan suatu proses/kegiatan bisnis.
	Fork (percabangan), digunakan untuk menunjukkan kegiatan yang dilakukan secara paralel atau untuk menggabungkan dua kegiatan paralel menjadi satu.
	join (penggabungan) atau rake, digunakan untuk menunjukkan adanya dekomposisi.
	Decision points, menggambarkan pilihan untuk pengambilan keputusan, true atau false.
	Swimlane, pembagian activity diagram untuk menunjukkan siapa melakukan apa.

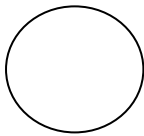
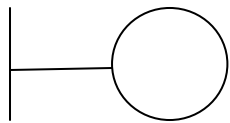
Gambar II.4. Simbol Activity Diagram
 (Sumber : Windu Gata ; 2013 : 6)

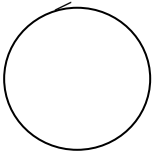

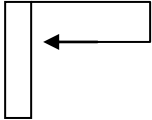
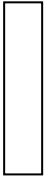


Gambar II.5. Activity Diagram
 (Sumber : Haviluddin ; 2011 : 3)

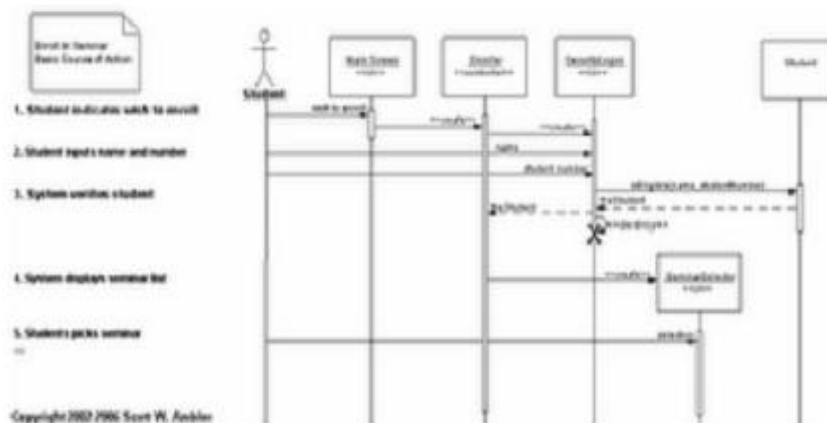
3. Diagram Urutan (*Sequence Diagram*)

Sequence diagram menggambarkan kelakuan objek pada *use case* dengan mendeskripsikan waktu hidup objek dan pesan yang dikirimkan dan diterima antar objek. Simbol-simbol yang digunakan dalam *Sequence Diagram*, yaitu:

	<p>Entity Class, merupakan bagian dari sistem yang berisi kumpulan kelas berupa entitas-entitas yang membentuk gambaran awal sistem dan menjadi landasan untuk menyusun basis data.</p>
	<p>Boundary Class, berisi kumpulan kelas yang menjadi interface atau interaksi antara satu atau lebih aktor dengan sistem, seperti tampilan formentry dan form cetak.</p>

	<p>Control Class, suatu objek yang berisi logika aplikasi yang tidak memiliki tanggung jawab kepada entitas, contohnya adalah kalkulasi dan aturan bisnis yang melibatkan berbagai objek. Control objek mengkoordinir pesan antar boundary dengan entitas.</p>
	<p>Message, simbol mengirim pesan antar Class.</p>
	<p>Recursive, menggambarkan pengiriman pesan yang dikirim untuk dirinya sendiri.</p>
	<p>Activation, activation mewakili sebuah eksekusi operasi dari objek, panjang kotak ini berbanding lurus dengan durasi kotak aktivasi sebuah operasi.</p>
	<p>Lifeline, garis titik-titik yang terhubung dengan objek, sepanjang lifeline terdapat activation.</p>

Gambar II.6. Simbol Sequence Diagram
 (Sumber : Windu Gata ; 2013 : 7)



Gambar II.7. Sequence Diagram
(Sumber : Havaluddin ; 2011 : 3)

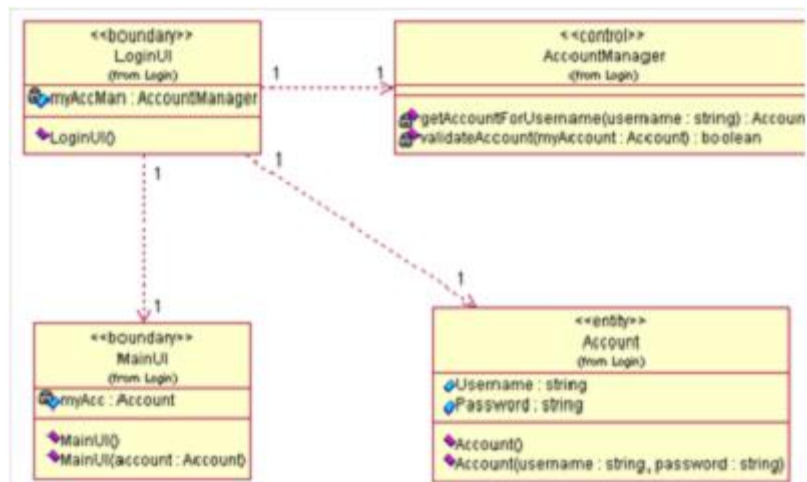
4. Diagram Kelas (*Class Diagram*)

Merupakan hubungan antar kelas dan penjelasan detail tiap-tiap kelas di dalam model desain dari suatu sistem, juga memperlihatkan aturan-aturan dan tanggung jawab entitas yang menentukan perilaku sistem. Class diagram juga menunjukkan atribut-atribut dan operasi-operasi dari sebuah kelas dan constraint yang berhubungan dengan objek yang dikoneksikan. Class diagram secara khas meliputi: kelas (*Class*), relasi, *associations*, *generalization* dan *aggregation*, atribut (*attributes*), operasi (*operations/method*), dan *visibility*, tingkat akses objek eksternal kepada suatu operasi atribut. Hubungan antar kelas mempunyai keterangan yang disebut dengan *Multiplicity* atau kardinaliti.

Multiplicity	Penjelasan
1	Satu dan hanya satu
0..*	Boleh tidak ada atau 1 atau lebih

1..*	1 atau lebih
0..1	Boleh tidak ada, maksimal 1
n..n	Batasan antara contoh: 2..4 mempunyai arti minimal 2 maksimum 4

Gambar II.8. Simbol Class Diagram
(Sumber : Windu Gata ; 2013 : 9)



Gambar II.9. Sequence Diagram
(Sumber : Havaluddin ; 2011 : 3)