BAB II

LANDASAN TEORI

II.1. Sistem

Sistem adalah kumpulan elemen yang saling berkaitan dan bekerja sama dalam melakukan kegiatan untuk mencapai suatu tujuan. Pengertian sistem dilihat dari masukan dan keluarannya. Sistem adalah suatu rangkaian yang berfungsi menerima *input* (masukan), mengolah *input*, dan menghasilkan *output* (keluaran). Sistem yang baik akan mampu bertahan dalam lingkungannya. Pengertian sistem dilihat dari prosedur/kegiatannya. Sistem adalah suatu rangkaian prosedur/kegiatan yang dibuat untuk melaksanakan program perusahaan (V.Wiratna Sujarweni ; 2015 : 1-2).

Sistem ialah serangkaian bagian yang saling tergantung dan bekerja sama untuk mencapai tujuan tertentu (Anastasia Diana & Lilis Setiawati ; 2011 : 3).

Dari kesimpulan diatas, Sistem adalah entitas atau satuan yang terdiri dari dua atau lebih komponen atau subsistem (sistem yang lebih kecil) yang saling terhubung dan terkait untuk mencapai suatu tujuan.

II.1.1. Akuntansi

Akuntansi adalah proses dari transaksi yang dibuktikan dengan faktur, lalu dari transaksi dibuat jurnal, buku besar, neraca lajur, kemudian akan menghasilkan informasi dalam bentuk laporan keuangan yang digunakan pihak-pihak tertentu (V.Wiratna Sujarweni ; 2015 : 3)

Akuntansi merupakan proses mengidentifikasi, mengukur, mencatat dan mengkomunikasikan peristiwa-peristiwa ekonomi dari suatu organisasi (bisnis maupun *nonbisnis*) kepada pihak-pihak yang berkepentingan dengan informasi bisnis tersebut (pengguna informasi) (Anastasia Diana & Lilis Setiawati ; 2011 : 14).

Dari kesimpulan diatas, Akuntansi adalah bahasa bisnis. Sebagai bahasa bisnis akuntansi menyediakan cara untuk menyajikan dan meringkas kejadian-kejadian bisnis dalam bentuk informasi keuangan kepada pemakainya.

II.1.2. Sistem Informasi

Sistem Informasi adalah cara yang terorganisir untuk mengumpulkan, memasukkan, dan memproses data dan menyimpannya, mengelola, mengontrol, dan melaporkannya sehingga dapat mendukung perusahaan atau organisasi untuk mencapai tujuan (Rudy Tantra ; 2012 : 2).

Sistem informasi dapat bersifat formal maupun informal. Sistem informasi akuntansi, produksi dan penjualan merupakan contoh informasi formal yang memang secara resmi memiliki tanggung jawab untuk menghasilkan informasi yang akurat. Sedangkan sistem informasi informal adalah kebalikannya, berasal dari bagian-bagian organisasi yang tidak secara resmi memberikan informasi, seperti misalnya bagian *legal* (Rudy Tantra ; 2012 : 2).

Sistem Informasi, yang kadang kala disebut sebagai sistem pemrosesan data, merupakan sistem buatan manusia yang biasanya terdiri dari sekumpulan komponen – baik *manual* ataupun berbasis komputer – yang *terintegrasi* untuk

mengumpulkan, menyimpan, dan mengelola data serta menyediakan informasi kepada pihak-pihak yang berkepentingan sebagai pemakai informasi tersebut (Anastasia Diana & Lilis Setiawati ; 2011 : 4).

Gambar II.1. Komponen Sistem Informasi
(Sumber : Anastasia Diana & Lilis Setiawati ; 2011)

Input dalam sistem informasi adalah data-data yang *relevan* untuk menghasilkan informasi yang diinginkan. Proses adalah langkah-langkah yang perlu dilakukan untuk mengolah data menjadi informasi. Sedangkan *output* adalah berupa informasi yang merupakan hasil dari pemrosesan data (Anastasia Diana & Lilis Setiawati ; 2011 : 4).

Dari kesimpulan diatas, Sistem Informasi adalah kombinasi dari teknologi informasi dan aktivitas orang yang menggunakan teknologi itu untuk mendukung operasi dan manajemen.

II.1.3. Sistem Informasi Akuntansi

Sistem Akuntansi adalah kumpulan elemen yaitu formulir, jurnal, buku besar, buku pembantu, dan laporan keuangan yang akan digunakan oleh manajemen untuk mencapai tujuan perusahaan (V.Wiratna Sujarweni ; 2015 : 3).

Sistem akuntansi terdiri dari *input* yang berupa transaksi yang dicatatkan dalam formulir *(input)* kemudian diproses (dengan menjurnal, membuat buku besar, membuat buku pembantu) dan hasil akhirnya *(output)* berupa laporan

keuangan yang digunakan manajemen untuk mencapai tujuan perusahaan (V.Wiratna Sujarweni ; 2015 : 4).

Dari kesimpulan diatas, Sistem Akuntansi adalah metode dan prosedur untuk mencatat dan melaporkan informasi keuangan yang disediakan bagi perusahaan atau suatu organisasi bisnis.

Sistem Informasi Akuntansi adalah sistem yang bertujuan untuk mengumpulkan dan memproses data serta melaporkan informasi yang berkaitan dengan transaksi keuangan (Anastasia Diana & Lilis Setiawati ; 2011 : 4).

Misalnya, salah satu *input* dari sistem informasi akuntansi penjualan sebuah toko baju. Kita memproses transaksi dengan mencatat penjualan tersebut ke dalam jurnal penjualan, mengklasifikasikan transaksi dengan menggunakan kode rekening, dan *memposting* transaksi ke dalam jurnal. Kemudian, secara periodik sistem informasi akuntansi akan menghasilkan *output* berupa laporan keuangan yang terdiri dari neraca dan laporan laba rugi (Anastasia Diana & Lilis Setiawati ; 2011 : 4-5).

Tujuan Sistem Informasi Akuntansi dapat dijelaskan dari manfaat yang didapat dari informasi akuntansi. Manfaat atau tujuan Sistem Informasi Akuntansi tersebut adalah sebagai berikut:

1. Mengamankan harta/kekayaan perusahaan.
2. Menghasilkan beragam informasi untuk pengambilan keputusan.
3. Menghasilkan informasi untuk pihak *eksternal*.
4. Menghasilkan informasi untuk penilaian kinerja karyawan atau *divisi*.
5. Menyediakan data masa lalu untuk kepentingan *audit* (pemeriksaan).

6. Menghasilkan informasi untuk penyusunan dan evaluasi anggaran perusahaan.
7. Menghasilkan informasi yang diperlukan dalam kegiatan perencanaan dan pengendalian (Anastasia Diana & Lilis Setiawati ; 2011 : 5-7).

Dari kesimpulan diatas, Sistem Informasi Akuntansi (SIA) adalah sebuah sistem informasi yang menangani segala sesuatu yang berkenaan dengan Akuntansi.

II.2. Sistem Pengendalian Intern

Sistem Pengendalian *Intern* adalah suatu sistem yang dibuat untuk memberi jaminan keamanan bagi unsur-unsur yang ada dalam perusahaan. Adapun tujuan perusahaan membuat sistem pengendalian *intern* adalah:

1. Untuk menjaga kekayaan organisasi
2. Untuk menjaga keakuratan laporan keuangan perusahaan
3. Untuk menjaga kelancaran operasi perusahaan
4. Untuk menjaga kedisiplinan dipatuhinya kebijakan manajemen
5. Agar semua lapisan yang ada diperusahaan tunduk pada hukum dan aturan yang sudah ditetapkan di perusahaan (V.Wiratna Sujarweni ; 2015 : 69).

Pengendalian *intern* diharapkan dapat melindungi kekayaan perusahaan yang diakibatkan dari pencurian, penggelapan keuangan oleh karyawan, penyalahgunaan, atau penempatan aktiva pada lokasi yang tidak tepat, dan lain sebagainya.

Menurut *COSO* pengendalian *internal* merupakan rangkaian tindakan yang mencakup keseluruhan proses dalam organisasi. Pengendalian *internal* berada dalam proses manajemen dasar, yaitu perencanaan, pelaksanaan, dan pemantauan.

COSO (Committee of Sponsoring Organizations of the Treadway Commission) adalah sektor swasta yang dibentuk pada tahun 1985. Tujuan utamanya untuk mengidentifikasi faktor-faktor yang menyebabkan penggelapan laporan keuangan dan untuk mengurangi kejadian tersebut. Selama ini *COSO* telah menyusun suatu definisi umum untuk pengendalian, standar, dan kriteria *internal* yang dapat digunakan perusahaan untuk menilai sistem pengendalian mereka.

COSO disponsori dan didanai oleh lima *asosiasi* dan lembaga akuntansi profesional yaitu:

1. *American Institute of Certified Public Accountants (AICPA)*
2. *American Accounting Association (AAA)*
3. *Financial Executives Institute (FEI)*
4. *The Institute of Internal Auditors (IIA)*
5. *The Institute of Management Accountants (IMA)* (V.Wiratna Sujarweni ; 2015 : 70).

Efektivitas merupakan kemampuan suatu unit untuk mencapai atau melampaui sasaran, target, atau tujuan yang diinginkan (yang telah ditetapkan lebih dahulu). *Efektivitas* menggambarkan hubungan suatu pusat pertanggungjawaban dengan tujuan yang dicapai (Islahuzzaman ; 2012 : 132).

Kegiatan pengendalian adalah suatu tindakan yang dibutuhkan untuk mengatasi resiko. Pada kegiatan ini antara lain menetapkan pelaksanaan prosedur kebijakan yang sudah dibuat serta memastikan apakah tindakan untuk mengatasi resiko sudah dilaksanakan secara *efektif* dan *efisien*. Kegiatan pengendalian *intern* sebagai berikut:

1. Pemberian otorisasi atas transaksi dan kegiatan
2. Pembagian tugas dan tanggung jawab
3. Dokumen yang akan digunakan sebaiknya dirancang terlebih dahulu
4. Perlindungan yang cukup ketat terhadap kekayaan dan catatan perusahaan
5. Pemeriksaan terhadap kinerja perusahaan (V.Wiratna Sujarweni ; 2015 : 74-75).

Dari kesimpulan diatas, Pengendalian *Internal* adalah semua rencana organisasional, metode, dan pengukuran yang dipilih oleh suatu kegiatan usaha untuk mengamankan harta kekayaannya, mengecek keakuratan dan keandalan data akuntansi usaha tersebut, meningkatkan efisiensi operasional, dan mendukung dipatuhinya kebijakan manajerial yang telah ditetapkan.

II.3. Sistem Akuntansi Pengeluaran Kas

Sistem akuntansi pengeluaran kas merupakan sistem yang membahas keluarnya uang yang digunakan untuk pembelian tunai maupun kredit dan untuk pembayarannya. Pengeluaran kas berupa pembayaran bisa menggunakan uang tunai maupun cek (V.Wiratna Sujarweni ; 2015 : 123).

Transaksi pengeluaran kas (sebagian perusahaan menyebutnya kas besar) digunakan untuk memfasilitasi pembayaran yang nilainya *material* dan dapat dibayar dengan menggunakan *transfer* atau cek. Dokumen yang digunakan untuk mencatat transaksi pengeluaran kas ini adalah bukti kas keluar (Anastasia Diana & Lilis Setiawati ; 2011 : 165).

Dari kesimpulan diatas, Sistem Akuntansi Pengeluaran Kas adalah suatu sistem pengolahan data akuntansi yang digunankan untuk mengelola kas, yang merupakan koordinasi dari manusia, alat dan metode yang berinteraksi secara harmonis untuk mengahasilkan informasi akuntansi pengeluaran kas sehingga dapat mengatur *likuiditas* kas.

II.4. Database

Database secara sederhana, dapat kita sebut sebagai gudang data. Secara teori, *database* adalah kumpulan data atau informasi yang *kompleks*, data-data tersebut disusun menjadi beberapa kelompok dengan tipe data yang sejenis (disebut tabel), dimana setiap datanya dapat saling berhubungan satu sama lain atau dapat berdiri sendiri, sehingga mudah diakses. Aplikasi yang dapat membuat dan mengelola *database* sering disebut SMBD (Sistem Manajemen Basis Data) atau DBMS *(Data Base Management System)* (Bunafit Nugroho ; 2008 : 1).

Apabila dikategorikan aplikasi *database* dapat dibedakan menjadi tiga, yaitu:

1. Aplikasi *database* berbasis *stand alone*, yaitu aplikasi yang hanya berjalan pada satu komputer dan hanya mampu diakses oleh satu orang dalam satu waktu.

2. Aplikasi *database* berbasis *multi user*, yaitu program tersebut dapat digunakan oleh banyak pengguna dalam satu waktu dan dalam tempat yang berbeda.
3. Aplikasi *database* berbasis *client/server*, untuk membuat aplikasi yang berbasis *Client/Server*, kita pasti membutuhkan aplikasi *database* yang bertindak sebagai *server* (pusat) data dan komputer yang dijadikan sebagai *Client* (pengakses). Sehingga kita harus menggunakan *database server* sebagai media penyimpanan datanya (Bunafit Nugroho ; 2008 : 3-4).

Basis data merupakan tempat yang dipergunakan untuk menyimpan data. Data tersebut merupakan data yang digunakan untuk melayani kebutuhan pemakai informasi. Data yang disimpan di basis data dapat berasal dari dalam perusahaan dan luar perusahaan (misal: data yang berasal dari *internet*) (V.Wiratna Sujarweni ; 2015 : 10).

II.5. Normalisasi

Normalisasi yaitu suatu proses untuk mengorganisasikan data dalam tabel secara *efisien*. Tujuan normalisasi ini adalah mengurangi redudansi data, yaitu agar tidak ada data yang sama yang tersimpan pada beberapa tabel. Tujuan lainnya adalah memastikan relasi antar tabel secara baik, sehingga hanya informasi yang *relevan* saja yang tersimpan dalam satu tabel. Proses normalisasi ini terdiri atas lima tahap *normal form (*NF), namun umumnya proses normalisasi hanya sampai NF3 dan dalam beberapa kasus NF4 dan Boyce-Codd Normal Form (BCNF) (Rudy Tantra ; 2012 : 163).

Ada beberapa langkah dalam normalisasi tabel, yaitu:

1. *First Normal Form (1NF)*

   Bentuk 1NF memberikan aturan paling dasar untuk *database* yang terorganisir yaitu:

   a. Menghilangkan kolom-kolom yang duplikat (berisi data yang sama) dalam satu tabel.
   b. Mengelompokkan data yang sejenis ke dalam tabel-tabel terpisah dan memberikan identifikasi khusus untuk naik setiap baris data *(Primary Key)* (Rudy Tantra ; 2012 : 164).

2. *Second Normal Form* (2NF)

   Bentuk 2NF lebih mengarahkan *database* untuk menghilangkan data yang duplikaf, dengan aturan berikut:

   a. Memenuhi semua aturan 1NF.
   b. Hilangkan subset data pada baris-baris data yang ada pada tabel dan pindahkan pada tabel lainnya.
   c. Buat hubungan antara tabel baru tersebut dengan tabel asalnya dengan menggunakan *foreign key*, yaitu kolom dari satu tabel yang sesuai dengan *primary key* dari tabel lainnya (Rudy Tantra ; 2012 : 165).

3. *Third Normal Form* (3NF)

   Bentuk 3NF adalah langkah lebih jauh dalam proses normalisasi, yakni:

   a. Memenuhi semua aturan 2NF.
   b. Hilangkan semua kolom yang tidak terkait dengan *primary key* (Rudy Tantra ; 2012 : 167).

II.6. SQL

SQL (Structured Query Language) merupakan suatu bahasa permintaan yang terstruktur. Kenapa terstuktur? karena pada penggunaannya, *SQL* memiliki beberapa aturan yang telah distandarkan oleh *asosiasi* yang bernama *ANSI (American National Standards Institute)*. Jadi, *SQL* adalah bahasa permintaan yang melekat pada satu *database.* Sebagai suatu bahasa permintaan, *SQL* tidak hanya melekat pada *MySQL* server saja, tetapi juga didukung oleh SMBD lainnya, seperti: *MsQL, PostgreSQL, Interbase,* dan *Oracle*. Selain itu *SQL* juga didukung oleh *database* bukan *server* seperti *MS Access* maupun *Paradox* (Bunafit Nugroho ; 2008 : 3).

Telah dikatakan sebelumnya bahwa *SQL* merupakan sebuah bahasa permintaan yang melekat pada suatu SMBD (Sistem Manajemen Basis Data) termasuk *MySQL*. Perintahnya dapat kita sebut dengan *query*. Dalam penggunaannya, perintah *SQL* dikategorikan menjadi tiga sub perintah yaitu:

1. *Data Definition Language (DDL)*, merupakan sub bahasa *SQL* yang digunakan untuk membangun kerangka *database.*
2. *Data Manipulation Language (DML)*, merupakan sub bahasa *SQL* yang digunakan untuk memanipulasi data dalam *database* yang telah terbuat.
3. *Data Control Language (DLC)*, merupakan sub bahasa *SQL* yang digunakan untuk melakukan pengontrolan data dan *server database* (Bunafit Nugroho ; 2008 : 5).

Bahasa *query* merupakan bahasa khusus yang digunakan untuk melakukan manipulasi dan menanyakan pertanyaan *(query)* yang berhubungan dengan data

dalam basis data. Bahasa *query* tidak sama dengan bahasa pemrograman, dimana bahasa *query* tidak memiliki kemampuan untuk menyelesaikan banyak masalah seperti bahasa pemrograman pada umumnya. Dalam pemrograman basis data, salah satu bahasa yang harus kita kuasai adalah *SQL*. *SQL* merupakan bahasa komputer standar yang digunakan untuk berkomunikasi dengan sistem manjemen basis data relasional (RDBMS) (Ema Utami dan Anggit Dwi Hartanto ; 2012 : 63).

Dari kesimpulan diatas, *SQL (Structured Query Language)* adalah sebuah bahasa yang digunakan untuk mengakses data dalam basis data relasional. perintah *SQL* dibedakan menjadi tiga yaitu: *Data Definition Language (DDL)*, *Data Manipulation Language (DML)*, dan *Data Control Language (DLC).*

II.7. Microsoft Visual Studio 2010

Microsoft Visual Studio adalah sebuah *Integrated Development Environment* buatan *Microsoft Coroporation*. *Microsoft Visual Studio* dapat digunakan untuk mengembangkan aplikasi dalam *native code* (dalam bentuk bahasa mesin yang berjalan di atas *Windows*) ataupun *managed code* (dalam bentuk *Microsoft Intermediate Language* di atas *.NET Framework*) (Wahana Komputer ; 2013 : 2).

Visual basic 2008 merupakan aplikasi pemrograman yang menggunakan teknologi *.NET Framework*. Teknologi *.Net Framework* merupakan komponen *windows* yang terintegrasi serta mendukung pembuatan, penggunaan aplikasi dan halaman *web*. Teknologi *.NET Framework* mempunyai 2 komponen utama yaitu

CLR (Common Language Runtime) dan *Class Library*. *CLR* digunakan untuk menjalankan aplikasi yang berbasis *.NET* sedangkan *Library* adalah kelas pustaka atau perintah yang digunakan untuk mengembangkan aplikasi (Wahana Komputer ; 2010 : 2)

Dari kesimpulan diatas, *Visual Studio* merupakan sebuah perangkat lunak lengkap *(suite)* yang dapat digunakan untuk melakukan pengembangan aplikasi, baik itu aplikasi bisnis, aplikasi *personal*, ataupun komponen aplikasinya, dalam bentuk *console*, aplikasi *windows*, ataupun aplikasi *Web*.

II.8. Unified Modelling Language (UML)

Unified Modelling Language (UML) adalah sebuah "bahasa" yang telah menjadi standar dalam industri untuk visualisasi, merancang dan mendokumentasikan sistem piranti lunak. Dengan menggunakan *UML* kita dapat membuat model untuk semua jenis aplikasi piranti lunak, dimana aplikasi tersebut dapat berjalan pada piranti keras, sistem operasi dan jaringan apapun, serta ditulis dalam bahasa pemrograman apapun (Yuni Sugiarti ; 2013 : 34).

Unified Modelling Language (UML) adalah bahasa spesifikasi standar yang dipergunakan untuk mendokumentasikan, menspesifikasikan dan membangun perangkat lunak. *UML* merupakan metodologi dalam mengembangkan sistem berorientasi objek dan juga merupakan alat untuk mendukung pengembangan sistem. *UML* saat ini sangat banyak dipergunakan dalam dunia industri yang merupakan standar bahasa pemodelan umum dalam

industri perangkat lunak dan pengembangan sistem (Windu Gata dan Grace Gata ; 2013 : 4).

Dari kesimpulan diatas, *UML (Unified Modelling Language)* adalah bahasa spesifikasi standar untuk mendokumentasikan, menspesifikasikan dan membangun sistem *software*. *UML* adalah hasil pengembangan dari bahasa pemodelan berorientasi objek *(object oriented modelling language).*

Alat bantu yang digunakan dalam perancangan berorientasi objek berbasiskan *UML* adalah sebagai berikut:

II.8.1. Use Case Diagram

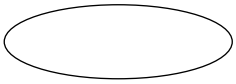
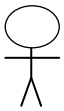
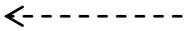
Use Case merupakan deskripsi dari sekumpulan urutan tindakan yang dilakukan oleh sistem dan menghasilkan nilai yang dapat diamati, kepada *actor* tertentu. Digunakan untuk melakukan strukturisasi aturan benda dalam model (Rudy Tantra ; 2012 : 152).

Use Case Diagram menggambarkan pemodelan 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 (Windu Gata dan Grace Gata ; 2013 : 4).

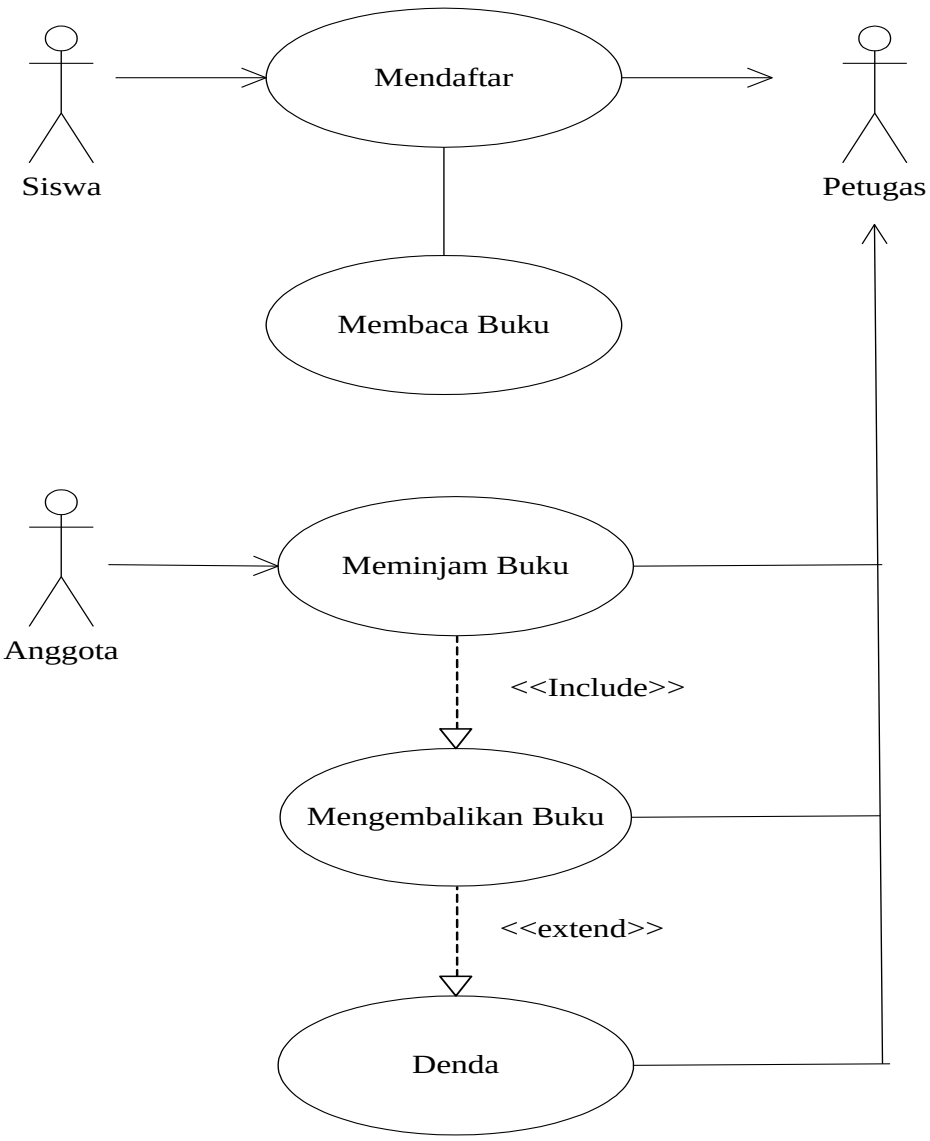
Dari kesimpulan diatas, *Use Case* adalah deskripsi fungsi dari sebuah sistem perspektif pengguna. *Use case* bekerja dengan cara mendeskripsikan tipikal interaksi antara *user* (pengguna) sebuah sistem dengan sistemnya sendiri

melalui sebuah cerita bagaimana sebuah sistem dipakai. Simbol-simbol yang digunakan dalam *use case diagram*, yaitu:

Tabel II.1. Simbol *Use Case*

Gambar	Keterangan
	*Use case* menggambarkan fungsionalitas yang disediakan sistem sebagai unit-unit yang bertukar pesan antar unit dengan aktor, biasanya dinyatakan dengan menggunakan kata kerja di awal nama *use case*.
	Aktor adalah *abstraction* dari orang atau sistem yang lain yang mengaktifkan fungsi dari target sistem. Untuk mengidentifikasikan aktor, harus ditentukan pembagian tenaga kerja dan tugas-tugas yang berkaitan dengan peran pada konteks target sistem. Orang atau sistem bisa muncul dalam beberapa peran. Perlu dicatat bahwa aktor berinteraksi dengan *use case*, tetapi tidak memiliki control terhadap *use case*.
	Asosiasi antara aktor dan *use case*, digambarkan dengan garis tanpa panah yang mengindikasikan siapa atau apa yang meminta interaksi secara langsung dan bukannya mengidikasikan aliran data.
	Asosiasi antara aktor dan *use case* yang menggunakan panah terbuka untuk mengidinkasikan bila aktor berinteraksi secara pasif dengan sistem.
	Include, merupakan di dalam *use case* lain (*required*) atau pemanggilan *use case* oleh *use case* lain, contohnya adalah pemanggilan sebuah fungsi program.
	*Extend*, merupakan perluasan dari *use case* lain jika kondisi atau syarat terpenuhi.

(Sumber : Windu Gata dan Grace Gata ; 2013)



Gambar II.2. Pembuatan *Use Case Diagram*
(Sumber : Windu Gata dan Grace Gata ; 2013)

II.8.2. Class Diagram (Diagram Kelas)

Diagram Kelas atau *Class Diagram* menggambarkan struktur sistem dari segi pendefinisian kelas-kelas yang akan dibuat dalam membangun sistem. Kelas memiliki apa yang disebut atribut, metode atau operasi (Yuni Sugiarti ; 2013 : 57).

Class Diagram merupakan hubungan antar kelas dan penjelasan detail tiap-tiap kelas di dalam model desain dari suatu sistem, juga memperlihatkan aturan-aturan dan tanggng jawab entitas yang menentukan perilaku sistem (Windu Gata dan Grace Gata ; 2013 : 8).

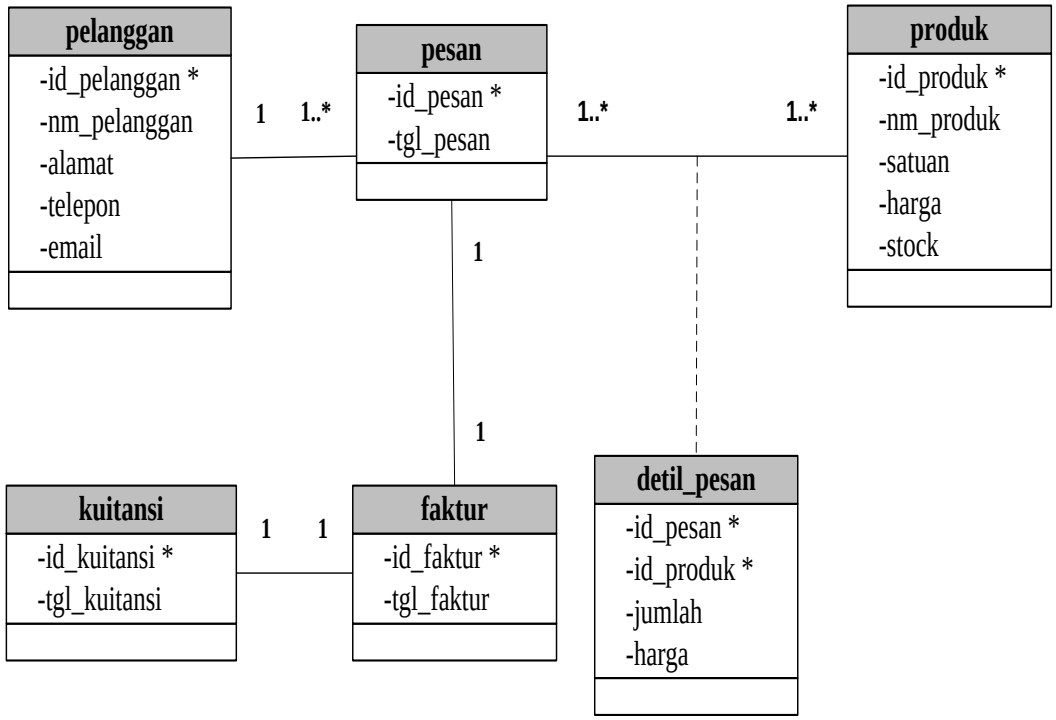
Dari kesimpulan diatas, *Class Diagram* adalah diagram kelas untuk menggambarkan struktur sistem, membaginya dalam kelas-kelas dengan koneksi dan relasi yang berbeda-beda.

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*), *Visibility*, tingkat akses objek eksternal kepada suatu operasi atau atribut. Hubungan antar kelas mempunyai keterangan yang disebut dengan *multiplicity* atau kardinaliti (Windu Gata dan Grace Gata ; 2013 : 8).

Tabel II.2. *Multiplicity Class Diagram*

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

(Sumber : Windu Gata dan Grace Gata ; 2013)



Gambar II.3. Pembuatan *Class Diagram*
(Sumber : Windu Gata dan Grace Gata ; 2013)

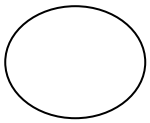
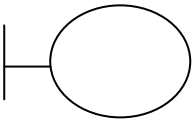
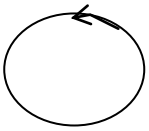
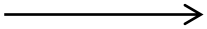
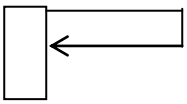
II.8.3. Diagram Urutan (Sequence Diagram)

Diagram Sekuence menggambarkan kelakuan/perilaku objek pada *use case* dengan mendeskripsikan waktu hidup objek dan *message* yang dikirimkan dan diterima antar objek. Oleh karena itu untuk menggambar *diagram sekuences* maka harus diketahui objek-objek yang terlibat dalam sebuah *use case* beserta metode-metode yang dimiliki kelas yang diinstansiasi menjadi objek itu (Yuni Sugiarti ; 2013 : 69).

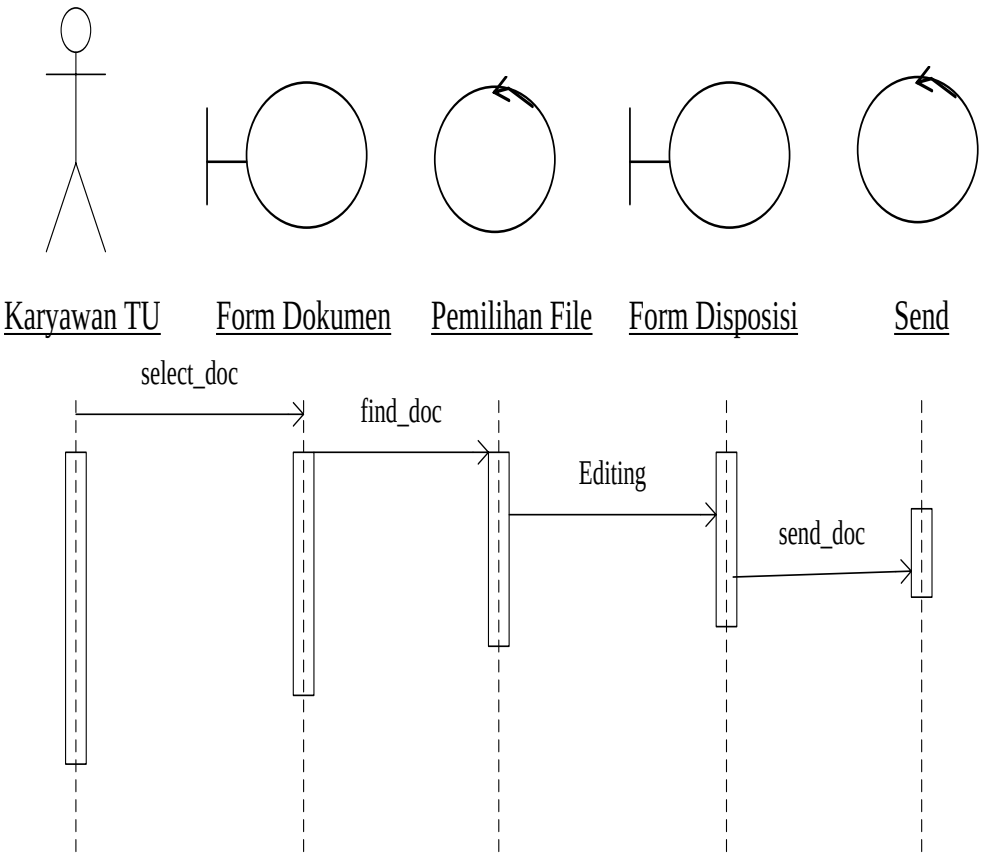
Sequence Diagram menggambarkan kelakuan objek pada *use case* dengan mendeskripsikan waktu hidup objek dan pesan yang dikirimkan dan diterima antar objek (Windu Gata dan Grace Gata ; 2013 : 7).

Dari kesimpulan diatas, *Sequence Diagram* menjelaskan interaksi objek yang disusun berdasarkan urutan waktu. Secara mudahnya *sequence diagram* adalah gambaran tahap demi tahap, termasuk kronologi (urutan) perubahan secara logis yang seharusnya dilakukan untuk menghasilkan sesuatu sesuai dengan *use case* diagram. Simbol-simbol yang digunakan dalam *sequence diagram*, yaitu:

Tabel II.3. Simbol *Sequence Diagram*

Gambar	Keterangan
	*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.
	*Boundary Class*, berisi kumpulan kelas yang menjadi *interface* atau interaksi antara satu atau lebih aktor dengan sistem, seperti tampilan formentry dan *form* cetak.
	*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.
	*Message*, simbol mengirim pesan antar *class*.
	*Recursive*, menggambarkan pengiriman pesan yang dikirim untuk dirinya sendiri.
	*Activation*, *activation* mewakili sebuah eksekusi operasi dari objek, panjang kotak ini berbanding lurus dengan durasi aktivitas sebuah operasi.
	Lifeline, garis titik-titik yang terhubung dengan objek, sepanjang *lifeline* terdapat *activation*

(Sumber : Windu Gata dan Grace Gata ; 2013)



Gambar II.4. Pembuatan *Sequence Diagram*
(Sumber : Windu Gata dan Grace Gata ; 2013)

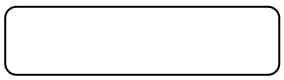
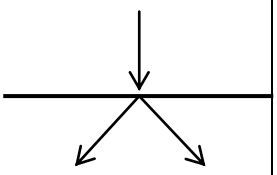
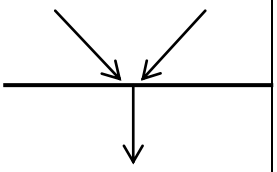
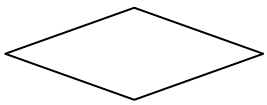
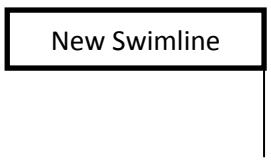
II.8.4. Diagram Aktivitas (Activity Diagram)

Activity Diagram merupakan *state diagram* khusus, dimana sebagian besar *state* adalah *action* dan sebagian transisi di *triggeer* oleh selesainya *state* sebelumnya *(internal processing)*. Oleh karena itu *Activity Diagram* tidak menggambarkan *behaviour internal* sebuah sistem (dan interaksi antar subsistem) secara eksak, tetapi lebih menggambarkan proses-proses dan jalur-jalur aktivitas dari level atas secara umum (Yuni Sugiarti ; 2013 : 58).

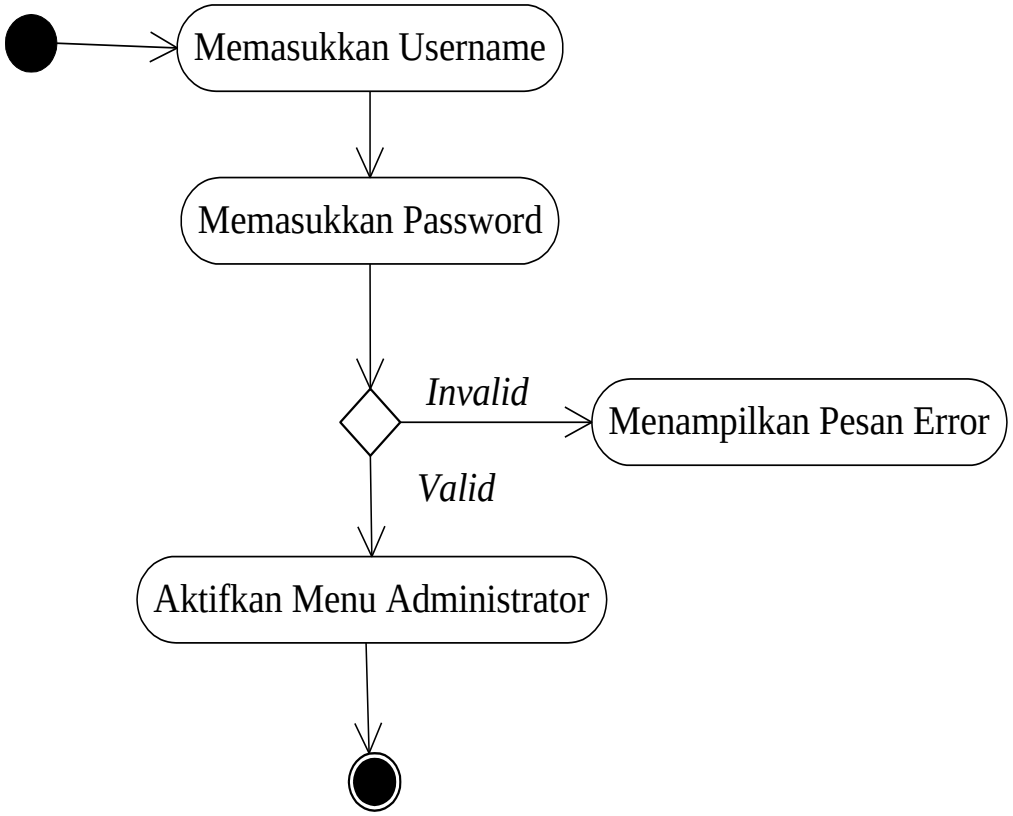
Activity Diagram menggambarkan *workflow* (aliran kerja) atau aktivitas dari sebuah sistem atau proses bisnis (Windu Gata dan Grace Gata ; 2013 : 6).

Dari kesimpulan diatas, *Activity Diagram* menggambarkan aktifitas-aktifitas, objek, *state, transisi state* dan *event*. Dengan kata lain kegiatan diagram alur kerja yang menggambarkan perilaku sistem untuk aktivitas. Simbol-simbol yang digunakan dalam *activity* diagram, yaitu:

Tabel II.4. Simbol *Activity Diagram*

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

(Sumber : Windu Gata dan Grace Gata ; 2013)



Gambar II.5. Pembuatan *Activity Diagram*
(Sumber : Windu Gata dan Grace Gata ; 2013)