

BAB II

TINJAUAN PUSTAKA

II.1. Pengertian Sistem

Sistem merupakan sekumpulan elemen-elemen yang saling terintegrasi serta melaksanakan fungsinya masing-masing untuk mencapai tujuan yang telah ditetapkan. Karakteristik sistem terdiri dari :

1. Komponen Sistem

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

2. Batasan Sistem

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. Batasan suatu sistem menunjukkan ruang lingkup (*scope*) dari sistem tersebut.

3. Lingkungan Luar Sistem

Lingkungan luar dari suatu sistem adalah apapun diluar batas dari sistem yang mempengaruhi operasi sistem. Lingkungan luar sistem dapat bersifat menguntungkan dan dapat juga bersifat merugikan sistem tersebut.

4. Penghubung Sistem

Penghubung merupakan media penghubung antara satu subsistem dengan subsistem lainnya. Melalui penghubung ini memungkinkan sumber-sumber daya mengalir dari satu subsistem ke subsistem lainnya.

5. Masukan Sistem

Masukan sistem adalah energi yang dimasukkan ke dalam sistem. Masukan dapat berupa masukan perawatan (*maintenance input*) dan masukan sinyal (*signal input*).

6. Keluaran Sistem

Keluaran sistem adalah hasil energi yang diolah dan diklasifikasikan menjadi keluaran yang berguna dan sisa pembuangan.

7. Pengolah Sistem

Suatu sistem dapat mempunyai suatu bagian pengolah atau sistem itu sendiri sebagai pengolahnya. Pengolah akan mengubah masukan menjadi keluaran.

8. Sasaran Sistem

Suatu sistem mempunyai tujuan (*goal*) atau sasaran (*objective*). Kalau suatu sistem tidak mempunyai sasaran, maka operasi sistem tidak ada gunanya (Sulindawati ; 2010 : 375).

II.1.1. Klasifikasi Sistem

Sistem merupakan suatu bentuk integrasi antara satu komponen dengan komponen lain karena sistem memiliki sasaran yang berbeda setiap kasus yang

terjadi yang ada di dalam sistem tersebut. Oleh karena itu sistem dapat diklasifikasikan dari beberapa sudut pandangnya antara lain.

1. Sistem Abstrak Dan Sistem Fisik

Sistem abstrak adalah sistem yang berupa pemikiran atau ide-ide yang tidak tampak secara fisik, misalnya sistem teologia, yaitu sistem yang berupa pemikiran hubungan antara manusia dengan Tuhan, sedangkan sistem fisik merupakan sistem yang ada secara fisik, misalnya sistem komputer, sistem produksi, sistem penjualan, sistem administrasi personalia, dan lain sebagainya.

2. Sistem Alamiah Dan Sistem Buatan Manusia.

Sistem alamiah adalah sistem yang terjadi melalui proses alam, tidak dibuat oleh manusia, misalnya sistem perputaran bumi, terjadinya siang malam, pergantian musim. Sedangkan sistem buatan manusia dengan mesin, merupakan melibatkan interaksi manusia dengan mesin, yang disebut "*human machine system*". Sistem informasi berbasis komputer merupakan contoh *human machine system* karena menyangkut penggunaan komputer yang berinteraksi dengan manusia.

3. Sistem *Deterministik* Dan Sistem *Probabilistik*.

Sistem yang beroperasi dengan tingkah laku yang dapat diprediksi disebut sistem deterministic. Sistem komputer adalah contoh dari sistem yang tingkah lakunya dapat dipastikan berdasarkan program-program komputer yang dijalankan. Sedangkan sistem bersifat probabilistik adalah sistem yang mana

kondisi masa depannya tidak dapat diprediksi karena mengandung unsur *probabilistic*.

4. Sistem Terbuka Dan Sistem Tertutup.

Sistem tertutup merupakan sistem yang tidak berhubungan dan tidak terpengaruh oleh lingkungan luarnya. Sistem ini bekerja secara otomatis tanpa campur tangan pihak luar. Sedangkan sistem terbuka adalah sistem yang berhubungan dan dipengaruhi oleh lingkungan luarnya. Sistem ini menerima masukan dan menghasilkan keluaran untuk subsistem lainya (Tata Sutabri, 2012).

II.2. Pengertian Informasi

Informasi adalah data yang diolah menjadi suatu bentuk yang berarti bagi penerimaannya dan bermanfaat dalam pengambilan keputusan pada saat sekarang atau yang akan datang. Informasi juga merupakan fakta-fakta atau data yang telah diproses sedemikian rupa atau mengalami proses transformasi data sehingga berubah bentuk menjadi informasi. Kualitas dari suatu informasi tergantung pada tiga hal yaitu :

1. Akurat, berarti informasi harus bebas dari kesalahan-kesalahan dan tidak bisa atau menyesatkan. Akurat juga berarti informasi harus jelas mencerminkan maksudnya. Informasi harus akurat karena dari sumber informasi sampain ke penerima informasi kemungkinan banyak terjadi gangguan (*noise*) yang dapat merubah atau merusak informasi tersebut.
2. Tepat pada waktunya, berarti informasi yang pada penerima tidak boleh terlambat. Informasi yang sudah using tidak akan mempunyai nilai lagi.

Karena informasi merupakan landasan di dalam pengambilan keputusan. Bila pengambilan keputusan tersebut, maka dapat berakibat fatal untuk organisasi.

Relevan, berarti informasi tersebut mempunyai manfaat untuk pemakainya. relevansi informasi untuk tiap-tiap orang satu dengan yang lainnya berbeda (Sulindawati ; 2010 : 2).

II.3. Sistem Informasi Akuntansi

Sistem informasi akuntansi adalah kumpulan sumberdaya, seperti manusia dan peralatan, yang diatur untuk mengubah data menjadi informasi. Informasi ini dikomunikasikan kepada beragam pengambil keputusan. SIA mewujudkan perubahan ini secara manual atau terkomputerisasi.

Sistem informasi akuntansi juga merupakan sistem yang paling penting di organisasi dan merubah cara menangkap, memproses, menyimpan, dan mendistribusikan informasi. Saat ini, digital dan informasi online semakin digunakan dalam sistem informasi akuntansi. Organisasi perlu menempatkan sistem di lini depan, dan mempertimbangkan baik segi sistem ataupun manusia sebagai factor yang terkait ketika mengatur sistem informasi akuntansi. (Agustinus , mujilan : 2012 : 3)

Sistem informasi, yang kadang kala disebut sebagai sistem pemrosesan data, merupakan sistem buatan manusia yang biasanya terdiri dari sekumpulan komponen, baik manual maupun berbasis computer yang terintegrasi untuk mengumpulkan, menyimpan, maupun mengelola data serta menyediakan informasi kepada pihak-pihak yang berkepentingan sebagai pemakai informasi tersebut.

Sistem Informasi Akuntansi adalah sistem yang bertujuan mengumpulkan dan memproses data serta melaporkan informasi yang berkaitan dengan transaksi keuangan. Misalnya, salah satu input dari Sistem Informasi Akuntansi pada sebuah toko baju.(Andi : 2011 : 4)

II.3.1. Tujuan Sistem Informasi Akuntansi

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

1. Mengamankan harta/kekayaan perusahaan

Harta//kekayaan disini meliputi kas perusahaan,persediaan barang dagangan, termasuk aset tetap perusahaan.

2. Menghasilkan beragam informasi untuk pengambilan keputusan.

Misal,pengelola toko swalayan memerlukan informasi mengenai barang apa yang diminati oleh konsumen.

3. Menghasilkan informasi untuk pihak eksternal

Setiap pengelola usaha memiliki kewajiban membayar pajak. Besarnya pajak tergantung dari besarnya omset penjualan atau tergantung pada laba rugi usaha. Tanpa sistem yang baik, bisa jadi pengelola kesulitan untuk menentukan besarnya omset dan besarnya laba rugi usaha.

4. Menghasilkan informasi untuk penilaian kinerja karyawan atau divisi

Sistem informasi dapat juga dimanfaatkan untuk penilaian kinerja karyawan atau divisi. Sebagai contoh, pengelola toko swalayan dapat memanfaatkan data

penjualan untuk menilai kinerja kasir. Kasir mana yang lebih cepat dan lebih cermat dalam melayani pelanggan.

5. Menyediakan data masa lalu untuk kepentingan audit

Data yang tersimpan dengan baik memudahkan proses audit. Satu hal yang penting, audit bukan eksklusif milik perusahaan publik. Semua perusahaan mesti siap untuk menghadapi pemeriksaan, karena kantor pajak punya wewenang untuk melakukan pemeriksaan terhadap wajib pajak.

6. Menghasilkan informasi untuk penyusunan dan evaluasi anggaran perusahaan

Anggaran merupakan alat yang sering digunakan perusahaan untuk mengendalikan pengeluaran kas.

7. Menghasilkan informasi yang diperlukan dalam kegiatan perencanaan dan pengendalian

Selain berguna untuk membandingkan informasi yang berkaitan dengan anggaran dan biaya standar dengan kenyataan seperti yang telah dikemukakan sebelumnya. (Andi : 2011 : 5,6,7)

II.4. Piutang

Klaim entitas pada pihak lain. Piutang yang timbul dari kegiatan entitas seperti penjualan atau memperoleh pendapatan disebut piutang dagang atau usaha. Piutang dalam bentuk kontrak tertulis disebut wesel tagih. Wesel tagih biasanya berbunga dan memiliki jangka waktu relative panjang. Untuk perusahaan perbankan dan jasa keuangan, piutang berbentuk pinjaman yang diberikan atau

kredit. Piutang disajikan dalam laporan posisi keuangan sebesar nilai diamortiasi dikurangi cadangan kerugian penurunan nilai.(Salemba ; 2012 : 234)

Jumlah yang akan ditagih dari pelanggan sebagai akibat penjualan barang secara kredit. Piutang usaha memiliki saldo normal disebelah debet sesuai dengan saldo normal untuk aktiva. Piutang usaha biasanya diperkirakan akan dapat ditagih dalam jangka waktu yang relative pendek, biasanya dalam waktu 30 hingga 60 hari. Setelah ditagih, secara pembukuan, piutang usaha akan berkurang disebelah kredit . Piutang usaha diklasifikasikan dalam neraca sebagai aktiva lancar(Hery,S.E.,M.Si. : 2011 : 36)

Menurut Kieso,dkk (2009, h.346) piutang adalah klaim uang, barang, atau jasa kepada pelanggan atau pihak-pihak lainnya. Untuk tujuan pelaporan keuangan, piutang diklasifikasikan sebagai lancar (jangka pendek) atau tidak lancar (jangka panjang).

II.4.1. Jenis – jenis piutang

1. Piutang Lancar

Piutang lancar merupakan piutang yang akan ditagih dalam masa satu tahun atau selama satu siklus operasi berjalan, mana yang lebih panjang.

2. Piutang Tidak Lancar

Piutang tidak lancar merupakan piutang yang akan tertagih lebih dari satu tahun.

Piutang selanjutnya diklasifikasikan dalam neraca yaitu sebagai :

a. Piutang dagang

Piutang dagang adalah jumlah yang terutang oleh pelanggan untuk barang dan jasa yang telah diberikan sebagai bagian dari operasional bisnis normal. Piutang dagang ini kemudian terbagi lagi menjadi dua yaitu piutang usaha dan wesel tagih..

b. Piutang Nondagang

Piutang nondagang berasal dari berbagai transaksi.

II.5. NetMethod (Metode Hapus Langsung)

Mengakui jumlah piutang setelah dikurangi potongan penjualan. Apabila ternyata debitur tidak memanfaatkan potongan, maka akan mengakibatkan timbulnya kelebihan pembayaran atas piutang. Kelebihan ini diakui sebagai penghasilan lain-lain.

Prosedur penjurnalan dan pembukuannya sbb :

1. Pada saat terjadi penjualan kredit barang dagangan, misalkan syarat kredit 2/10- n/10

Jurnal : Account Receivable	xx
Sales Dicount	xx
Sales	xx

2. Apabila pelunasan piutang dagang masih dalam batas waktu potongan, maka perusahaan harus memperhitungkan dan memberikan potongan penjualan, yaitu 2%.

Jurnal : Cash	xx
Account Receivable	xx

Metode ini kerap digunakan terutama oleh perusahaan yang memiliki bidang usaha seperti toko eceran dengan skala bisnis yang relatif kecil. Ketika metode hapus langsung digunakan, beban kredit macet atau beban piutang yang tidak dapat ditagih hanya akan dicatat atau diakui apabila benar-benar

telah terjadi pelanggaran tertentu yang menyatakan tidak bisa membayar, bukan berdasarkan pada kerugian estimasi.

Metode hapus langsung sangatlah sederhana, akan tetapi metode ini tidak sesuai dengan konsep penandingan. Penggunaan metode hapus langsung dapat mengurangi utilitas(manfaat) laporan keuangan , baik laporan laba rugi maupun neraca (Hery,S.E.,M.Si. ; 2011 : 43,44,45)

II.6. Teknik Normalisasi

Normalisasi adalah teknik perancangan yang banyak digunakan sebagai pemandu dalam merancang basis data relasional. Pada dasarnya, normalisasi adalah proses dua langkah yang meletakkan data dalam bentuk tabulasi dengan menghilangkan kelompok berulang lalu menghilangkan data yang terduplikasi dari tabel rasional.

Teori normalisasi didasarkan pada konsep bentuk normal. Sebuah tabel relasional dikatakan berada pada bentuk normal tertentu jika tabel memenuhi himpunan batasan tertentu. Ada lima bentuk normal yang telah ditemukan,yaitu :

1. Bentuk normal tahap pertama (1” Normal Form)

Contoh yang kita gunakan di sini adalah sebuah perusahaan yang mendapatkan barang dari sejumlah pemasok. Masing-masing pemasok berada pada satu kota. Sebuah kota dapat mempunyai lebih dari satu pemasok dan masing-masing kota mempunyai kode status tersendiri.

2. Bentuk normal tahap kedua (2nd normal form)

Definisi bentuk normal kedua menyatakan bahwa tabel dengan kunci utama gabungan hanya dapat berada pada 1NF, tetapi tidak pada 2NF. Sebuah tabel relasional berada pada bentuk normal kedua jika dia berada pada bentuk normal kedua jika dia berada pada 1NF dan setiap kolom bukan kunci yang sepenuhnya tergantung pada seluruh kolom yang membentuk kunci utama.

3. Bentuk normal tahap ketiga (3rd normal form)

Bentuk normal ketiga mengharuskan semua kolom pada tabel relasional tergantung hanya pada kunci utama. Secara definisi, sebuah tabel berada pada bentuk normal ketiga (3NF) jika tabel sudah berada pada 2NF dan setiap kolom yang bukan kunci tidak tergantung secara transitif pada kunci utamanya.

4. Boyce Code Normal Form (BCNF)

Setelah 3NF, semua masalah normalisasi hanya melibatkan tabel yang mempunyai tiga kolom atau lebih dan semua kolom adalah kunci. Banyak praktisi berpendapat bahwa menempatkan entitas pada 3NF sudah cukup karena sangat jarang entitas yang berada pada 3NF bukan merupakan 4NF dan 5NF.

5. Bentuk Normal Tahap Keempat dan Kelima

Sebuah tabel relasional berada pada bentuk normal keempat (4NF) jika dia dalam BCNF dan semua ketergantungan multivalued merupakan ketergantungan fungsional. Bentuk normal keempat (4NF) didasarkan pada konsep ketergantungan multivalued (MVD).

Sebuah tabel berada pada bentuk normal kelima (5NF) jika ia tidak dapat mempunyai dekomposisi lossless menjadi sejumlah tabel lebih kecil. Empat

bentuk normal pertama berdasarkan pada konsep ketergantungan fungsional, sedangkan bentuk normal kelima berdasarkan pada konsep ketergantungan gabungan (*join dependence*) (Janner Simarmata ; 2010 : 76).

II.7. Pengertian Database

Secara sederhana *database* (basis data/pangkalan data) dapat diungkapkan sebagai suatu pengorganisasian data dengan bantuan komputer yang memungkinkan data dapat diakses dengan mudah dan cepat. Pengertian akses dapat mencakup pemerolehan data maupun manipulasi data seperti menambah serta menghapus data. Dengan memanfaatkan komputer, data dapat disimpan dalam media pengingat yang disebut *harddisk*. Dengan menggunakan media ini, keperluan kertas untuk menyimpan data dapat dikurangi. Selain itu, data menjadi lebih cepat untuk diakses terutama jika dikemas dalam bentuk *database*.

Pengaplikasian *database* dapat kita lihat dan rasakan dalam keseharian kita. *Database* ini menjadi penting untuk mengelola data dari berbagai kegiatan. Misalnya, kita bisa menggunakan mesin ATM (*anjungan tunai mandiri / automatic teller machine*) bank karena bank telah mempunyai *database* tentang nasabah dan rekening nasabah. Kemudian data tersebut dapat diakses melalui mesin ATM ketika bertransaksi melalui ATM. Pada saat melakukan transaksi, dalam konteks *database* sebenarnya kita sudah melakukan perubahan (*update*) data pada *database* di bank. Ketika kita menyimpan alamat dan nomor telepon di HP, sebenarnya juga telah menggunakan konsep *database*. Data yang kita simpan

di HP juga mempunyai struktur yang diisi melalui formulir(*form*) yang disediakan. Pengguna dimungkinkan menambahkan nomor HP, nama pemegang, bahkan kemudian dapat ditambah dengan alamat *email*, alamat *web*, nama kantor, dan sebagainya (Agustinus Mujilan ; 2012 : 23).

II.8. Visual Studio

Visual Basic 2010 merupakan salah satu bagian dari pemrograman terbaru yang dikeluarkan oleh *Microsoft* yaitu *Microsoft Visual Studio* 2010. Sebagai pengembangan terintegrasi atau *IDE* andalan yang dikeluarkan oleh *Microsoft*, *Visual Studio* 2010 menambahkan perbaikan –perbaikan fitur dan fitur baru yang lebih lengkap dibandingkan versi *Visual Studio* sebelumnya, yaitu *Microsoft Visual Studio* 2008. *Visual studio* berisi beberapa jenis *IDE* pemrograman seperti *Visual Basic*, *Visual C++* , *Visual Web Developer*, *Visual C#* dan *Visual F#*. (Wahana Komputer ; 2010 : 2)

II.9. SQL Server

SQL Server 2008 adalah sebuah RDBMS (*Relational Database Management System*) yang di *develop* oleh *Microsoft*, yang digunakan untuk menyimpan dan mengolah data. Pada *SQL Server* 2008, kita bisa melakukan pengambilan dan modifikasi data yang ada dengan cepat dan efisien. Pada *SQL Server* 2008, kita bisa membuat *object – object* yang sering digunakan pada aplikasi bisnis, seperti membuat *database*, *table*, *fuction*, *stored procedure*, *trigger* dan *view*. Selain

object, kita juga menjalankan perintah SQL (*Structured Query Language*) untuk mengambil data. (Aryo Nugroho ; 101)

II.10. *Unified Modeling Language* (UML)

Menurut Windu Gata (2013 : 4) Hasil pemodelan pada OOAD terdokumentasikan dalam bentuk *Unified Modeling Language* (UML). 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.

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

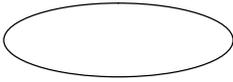
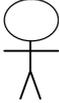
- *Use case* Diagram

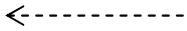
Use case diagram merupakan pemodelan untuk kelakuan (*behavior*) sistem informasi yang akan dibuat. *Use case* mendeskripsikan sebuah interaksi antara satu atau lebih aktor dengan sistem informasi yang akan dibuat. Dapat dikatakan *use case* digunakan untuk mengetahui fungsi apa saja yang ada di dalam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

Simbol-simbol yang digunakan dalam *use case* diagram dapat dilihat pada tabel

II.1 dibawah ini :

Tabel II.1. Simbol Use Case

Gambar	Keterangan
	<p><i>Use case</i> menggambarkan fungsionalitas yang disediakan sistem sebagai unit-unit yang bertukar pesan antar unit dengan aktor, biasanya dinyatakan dengan menggunakan kata kerja di awal nama <i>use case</i>.</p>
	<p>Aktor adalah <i>abstraction</i> dari orang atau sistem yang lain yang mengaktifkan fungsi dari target sistem. Untuk mengidentifikasi 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 <i>use case</i>, tetapi tidak memiliki control terhadap <i>use case</i>.</p>
	<p>Asosiasi antara aktor dan <i>use case</i>, digambarkan dengan garis tanpa panah yang mengindikasikan siapa atau apa yang meminta interaksi secara langsung dan bukannya mengidikasikan aliran data.</p>
	<p>Asosiasi antara aktor dan <i>use case</i> yang</p>

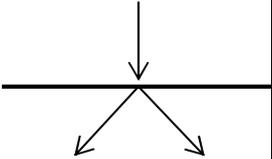
	menggunakan panah terbuka untuk mengidinkasikan bila aktor berinteraksi secara pasif dengan sistem.
	<i>Include</i> , merupakan di dalam <i>use case</i> lain (<i>required</i>) atau pemanggilan <i>use case</i> oleh <i>use case</i> lain, contohnya adalah pemanggilan sebuah fungsi program.
	<i>Extend</i> , merupakan perluasan dari <i>use case</i> lain jika kondisi atau syarat terpenuhi.

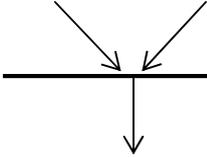
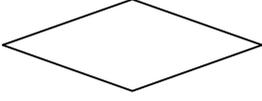
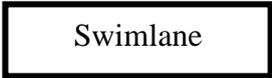
(Sumber : Windu Gata ; 2013 : 4)

- 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* dapat dilihat pada tabel II.2 dibawah ini :

Tabel II.2. Simbol *Activity Diagram*

Gambar	Keterangan
	<i>Start point</i> , diletakkan pada pojok kiri atas dan merupakan awal aktifitas.
	<i>End point</i> , akhir aktifitas
	<i>Activites</i> , menggambarkan suatu proses/kegiatan bisnis.
	<i>Fork</i> (Percabangan), digunakan untuk menunjukkan kegiatan yang dilakukan secara parallel atau untuk menggabungkan dua kegiatan pararel menjadi satu.

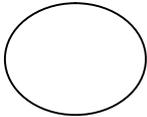
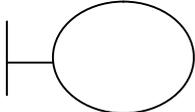
	<p><i>Join</i> (penggabungan) atau rake, digunakan untuk menunjukkan adanya dekomposisi.</p>
	<p><i>Decision Points</i>, menggambarkan pilihan untuk pengambilan keputusan, <i>true</i>, <i>false</i>.</p>
	<p><i>Swimlane</i>, pembagian <i>activity</i> diagram untuk menunjukkan siapa melakukan apa.</p>

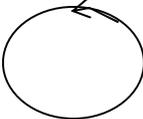
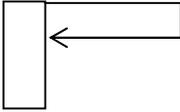
(Sumber : Windu Gata ; 2013 : 6)

- 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* dapat dilihat pada tabel II.3 dibawah ini :

Tabel II.3. Simbol *Sequence Diagram*

Gambar	Keterangan
	<p><i>Entity Class</i>, 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><i>Boundary Class</i>, berisi kumpulan kelas yang menjadi <i>interface</i> atau interaksi antara satu atau lebih aktor dengan sistem, seperti tampilan formentry dan <i>form</i></p>

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

(Sumber : Windu Gata ; 2013 : 7)

- *Class Diagram* (Diagram Kelas)

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*), *Visibility*, tingkat akses objek eksternal kepada suatu operasi atau atribut.

Hubungan antar kelas mempunyai keterangan yang disebut dengan *multiplicity* atau kardinaliti yang dapat dilihat pada tabel II.4 dibawah ini :

Tabel II.4. *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 ; 2013 : 8*)