

BAB II

LANDASAN TEORI

II.1 Bab Studi Literatur

II.1.1 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 sistem 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 : 135).

II.1.2 Pengertian Sistem Informasi

Sistem informasi bukan merupakan hal yang baru, yang baru adalah komputerisasinya. Sebelum ada komputer, teknik penyaluran informasi yang memungkinkan manajer merencanakan serta mengendalikan operasi telah ada. Komputer menambahkan satu atau dua dimensi, seperti kecepatan, ketelitian dan

penyediaan data dengan volume yang lebih besar yang memberikan bahan pertimbangan yang lebih banyak untuk mengambil keputusan.

Sistem informasi adalah suatu sistem di dalam suatu organisasi yang mempertemukan kebutuhan pengolahan transaksi harian yang mendukung fungsi operasi organisasi untuk dapat menyediakan laporan-laporan yang diperlukan oleh pihak luar tertentu (Tata Sutabri ; 2012 : 38).

II.1.3 Sistem Informasi Akuntansi

Sistem informasi akuntansi adalah sistem yang bertujuan untuk mengumpulkan dan memproses data serta melaporkan informasi yang berkaitan dengan transaksi keuangan. Lingkup 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. Harta / kekayaan di sini 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 saja yang diminati oleh konsumen. Membeli barang yang kurang laku berarti kas akan terjebak dalam persediaan dan berarti kehilangan kesempatan untuk membeli barang dagangan yang laku.
3. Menghasilkan informasi untuk pihak eksternal. Setiap pengelola usaha memiliki kewajiban untuk membayar pajak. Besarnya pajak yang dibayar tergantung pada omset penjualan (jika pengelola memilih menggunakan norma dalam perhitungan pajaknya) atau tergantung pada laba rugi usaha (jika

pengelola memilih untuk tidak menggunakan norma dalam perhitungan pajaknya).

4. Menghasilkan informasi untuk penilaian kinerja karyawan atau divisi. Sistem informasi dapat juga dimanfaatkan untuk penilaian kinerja karyawan atau divisi.
5. Menyediakan data masa lalu untuk kepentingan audit (pemeriksaan). Data yang tersimpan dengan baik sangat memudahkan proses audit (pemeriksaan).
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 (Anastasia Diana ; 2011 : 6).

II.1.4 Penyusutan

Seiring dengan waktu pemakaian sebuah aset tetap, maka pada saat yang sama aset tetap tersebut akan mulai berkurang kemampuannya atau mulai mengalami keusangan (*obsolescence*) untuk menciptakan barang dan jasa. Berkurangnya kemampuan aset tetap ini disebut sebagai penyusutan atau depresiasi (*depreciation*). Jumlah yang dapat disusutkan dialokasikan ke setiap periode akuntansi selama masa manfaat aktiva dengan berbagai metode yang sistematis. Metode manapun yang dipilih, konsistensi dalam penggunaannya adalah perlu, tanpa memandang tingkat profitabilitas perusahaan dan pertimbangan

perpajakan, agar dapat menyediakan daya banding hasil operasi perusahaan dari periode ke periode. Aktiva tetap berwujud dapat disusutkan dalam beberapa metode, beberapa jenis metode penyusutan atas aset tetap menurut PSAK 16 yang dapat diterapkan di Indonesia adalah metode penyusutan garis lurus (*straight line method*), saldo menurun ganda (*double declining balance method*), dan metode unit produksi (*units of production method*). Serta tambahan metode penyusutan lainnya yaitu penyusutan berdasarkan jumlah angka tahun (*sum of the years digits method*) (Samuel Mairuhu ; 2014 : 404).

II.1.4.1 Metode Penyusutan

Menurut Winarto, S.Pd (2008). Metode Penyusutan atau Depresiasi yang digunakan untuk menentukan besarnya penyusutan aktiva tetap dalam tiap periode, maka diperlukan perhitungan. Besarnya perhitungan penyusutan aktiva tetap akan berbeda-beda, tergantung pendekatan mana yang diambil perusahaan untuk menentukan besarnya penyusutan. Penentuan besarnya penyusutan setiap periode akuntansi untuk berbagai jenis aktiva tetap dapat dilakukan dengan beberapa metode antara lain :

- a. Metode garis lurus (*Straight Line Method*)
- b. Metode saldo menurun (Depresiasi), yang meliputi :
 - Metode jumlah angka tahun (*Sum of The Years Digits Method*)
 - Metode Tarif Tetap atas nilai buku / Metode saldo menurun ganda (*Double Declining Balance Method*)
- c. Metode satuan produksi (*Unit of Production Method*)
 - Metode Hasil Produksi (*Productive Output Method*)

- Metode jam kerja (Service Hours Method)

d. Metode tarif kelompok / gabungan (*Composite Rate Depreciation Method*)

Penyusutan dengan saldo menurun ganda (*double declining balance method*) merupakan suatu metode dipercepat dimana untuk mengalokasikan harga pokok / harga perolehan aktiva tetap menjadi beban penyusutan untuk setiap tahunnya semakin kecil atau menurun yang menggunakan tarif penyusutan dua kali dari yang digunakan metode garis lurus. Untuk menentukan tarif penyusutan sebagai berikut :

$$\text{Tarif Penyusutan} = \frac{100\%}{\text{Umur Manfaat}} \times 2$$

$$\text{Beban Penyusutan} = T \times \text{Harga Perolehan}$$

Atau

$$\text{Tarif Penyusutan} = \frac{100\%}{\text{Umur Manfaat}} \times 2 \times \text{Harga Perolehan}$$

$$\text{Tahun Kedua dst} = T \times (\text{HP} - \text{Penyusutan})$$

Keterangan :

HP = Harga Perolehan

T = Tarif Penyusutan (dalam persen) $100\% / \text{UE} \times 2$

Nilai Buku = Harga Perolehan – Penyusutan

Akumulasi Tahun kedua dst = Penyusutan + Akum Tahun Sebelumnya

II.1.5 Aktiva Tetap

Aktiva tetap adalah aktiva yang dimiliki oleh perusahaan yang digunakan untuk aktivitas/operasi perusahaan dan memiliki manfaat lebih dari satu tahun.

Aktiva tetap ini mempunyai sifat tetap atau permanen dibeli untuk digunakan

dalam kegiatan normal perusahaan, tidak untuk dijual kembali dan nilainya cukup besar atau material.

Suatu aktiva dapat dikatakan termasuk dalam aktiva tetap apabila memenuhi kriteria sebagai berikut :

1. Digunakan dalam kegiatan normal perusahaan bukan untuk dijual kembali atau investasi.
2. Dapat dipakai atau digunakan secara berulang-ulang.
3. Masa manfaatnya lebih dari satu tahun atau satu siklus operasi normal perusahaan.
4. Mempunyai nilai yang cukup material artinya nilai atau harga aktiva tersebut cukup tinggi.

Berdasarkan sifat-sifatnya, aktiva tetap dibagi atas :

1. Aktiva tetap berwujud (*tangible fixed assets*) : Aktiva yang mempunyai bentuk fisik
2. Aktiva tetap tidak berwujud (*intangible fixed assets*) : adalah suatu hak tertentu untuk jangka panjang yang mempunyai nilai ekonomis dan yang tidak mempunyai bentuk fisik (Winarto ; 2010 : 6).

II.1.6 Pengertian Java

Java adalah bahasa pemrograman yang dapat dijalankan di berbagai jenis komputer dan berbagai sistem operasi termasuk telepon genggam. Java dikembangkan oleh *Sun Microsystem* dan dirilis tahun 1995. Java merupakan suatu teknologi perangkat lunak yang digolongkan *multi platform*. Selain itu, Java

juga merupakan suatu *platform* yang memiliki *virtual machine* dan *library* yang diperlukan untuk menulis dan menjalankan suatu program.

Bahasa pemrograman java pertama lahir dari *The Green Project*, yang berjalan selama 18 bulan, dari awal tahun 1991 hingga musim panas 1992. Proyek tersebut belum menggunakan *versi* yang dinamakan Oak. Proyek ini dimotori oleh *Patrick Naughton*, *Mike Sheridan*, *James Gosling* dan *Bill Joy*, serta Sembilan pemrograman lainnya dari *Sun Microsystem*. Salah satu hasil proyek ini adalah mascot Duke yang dibuat oleh Joe Palrang (Wahana Komputer ; 2010 : 1).

II.1.7 Pengertian NetBeans

NetBeans merupakan salah satu proyek *open source* yang disponsori oleh *Sun Microsystem*. Proyek ini berdiri pada tahun 2000 dan telah menghasilkan 2 produk, yaitu *NetBeans IDE* dan *NetBeans Platform*. *NetBeans IDE* merupakan produk yang digunakan untuk melakukan pemrograman baik menulis kode, meng-*compile*, mencari kesalahan dan mendistribusikan program. Sedangkan *NetBeans Platform* adalah sebuah modul yang merupakan kerangka awal / pondasi dalam bangun aplikasi desktop yang besar.

NetBeans juga menyediakan paket yang lengkap dalam pemrograman dari pemrograman standar (aplikasi desktop), pemrograman enterprise, dan pemrograman perangkat mobile. Saat ini *NetBeans* telah mencapai versi 8.0 (Wahana Komputer ; 2010 : 15).

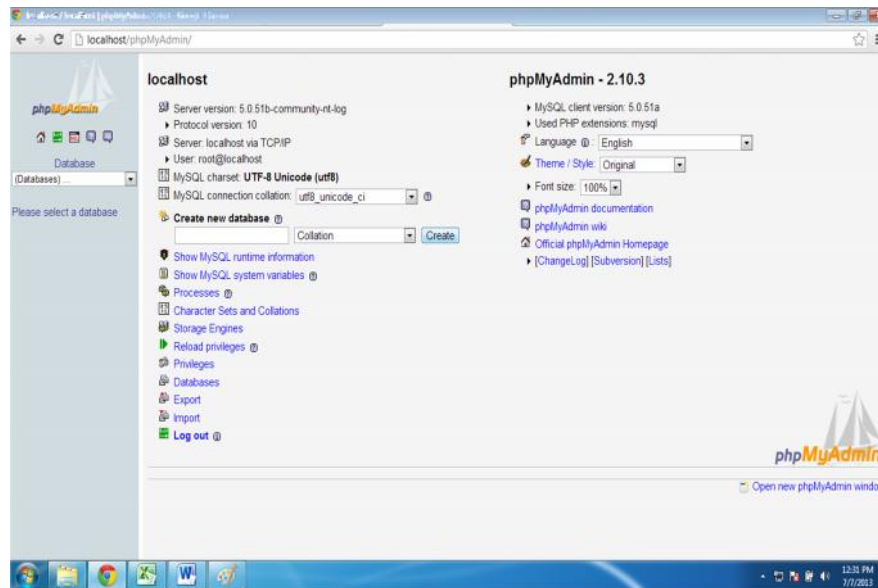


Gambar II.1. Tampilan Awal Netbeans 8.0
Sumber: www.netbeans.org

II.1.8 Pengertian MySQL

Mysql pertama kali dirintis oleh seorang programmer database bernama Michael Widenius, yang dapat anda hubungi di emailnya monty@analytikerna. Mysql *database server* adalah RDBMS (*Relasional Database Management System*) yang dapat menangani data yang bervolume besar. meskipun begitu, tidak menuntut *resource* yang besar. Mysql adalah *database* yang paling populer di antara *database* yang lain.

MySQL adalah program *database* yang mampu mengirim dan menerima data dengan sangat cepat dan *multi user*. MySQL memiliki dua bentuk lisensi, yaitu *free software* dan *shareware*. penulis sendiri dalam menjelaskan buku ini menggunakan *database* ini untuk keperluan pribadi atau usaha tanpa harus membeli atau membayar lisensi, yang berada di bawah lisensi GNU/GPL (*general public license*) (Wahana Komputer; 2010 : 5).



Gambar II.2. Tampilan MySQL
(Sumber : Wahana Komputer ; 2010 : 5)

II.1.9 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.

II.1.9.1 Bentuk-bentuk Normalisasi

1. Bentuk normal tahap pertama (1st 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.

5. 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.

6. 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.1.10 UML (*Unified Modeling Language*)

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 :

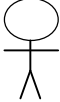
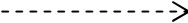
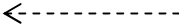
1. *Use case* Diagram

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, yaitu :

Tabel II.1. Simbol Use Case

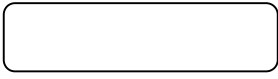
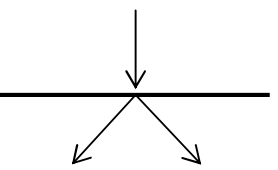
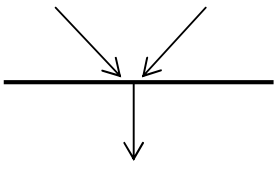
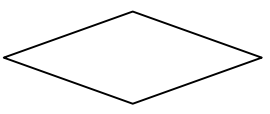
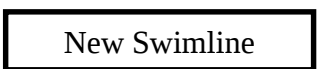
Gambar	Keterangan
	<i>Use case</i> menggambarkan fungsionalitas yang disediakan sistem sebagai unit-unit yang bertukar pesan antar unit dengan aktor, biasanya dinyatakan dengan menggunakan kata kerja di awal nama <i>use case</i> .
	Aktor adalah <i>abstraction</i> dari orang atau sistem yang lain yang 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> .
	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 mengindikasikan aliran data.
	Asosiasi antara aktor dan <i>use case</i> yang menggunakan panah terbuka untuk mengindikasikan 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)

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 :

Tabel II.2. Simbol *Activity Diagram*

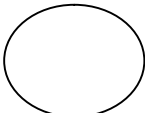
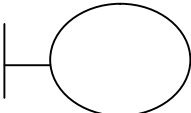
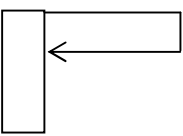
Gambar	Keterangan
	<i>Start point</i> , diletakkan pada pojok kiri atas dan merupakan awal aktifitas.
	<i>End point</i> , akhir aktifitas.
	<i>Activites</i> , menggambarkan suatu proses/kegiatan bisnis.
	<i>Fork</i> (Percabangan), digunakan untuk menunjukkan kegiatan yang dilakukan secara parallel atau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

(Sumber : Windu Gata ; 2013 : 6)

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 :

Tabel II.3. Simbol *Sequence Diagram*

Gambar	Keterangan
	<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.
	<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> 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)

4. *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.

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 : 9)