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 (*maintanceinput*) dan masukan sinyal (*signalinput*).

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.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.3. Akuntansi

Akuntansi merupakan bahasa bisnis. Sebagai bahasa bisnis akuntansi menyediakancara untuk menyajikan dan meringkas kejadian-kejadian bisnis dalam bentukinformasi keuangan kepada pemakainya.Informasi akuntansi merupakan bagian terpenting dari seluruh informasi yangdiperlukan oleh manajemen. Informasi akuntansi yang dihasilkan oleh suatu sistemdibedakan menjadi dua, yaitu informasi akuntansi keuangan dan informasi akuntansi manajemen.

Pemakai informasi akuntansi pun terdiri dari dua kelompok, yaitu pemakai eksternaldan pemakai internal. Yang dimaksud dengan pemakai ekseternal mencakuppemegang saham, investor, kreditor, pemerintah, pelanggan, pemasok, pesaing, serikat kerja dan masyarakat. Sedangkan pemakai internal adalah pihak manajer dariberbagai tingkatan dalam organisasi bersangkutan (Kusrini ; 2012 : 1).

II.4. Sistem Informasi Akuntansi

Informasi akuntansi diberikan sebagai alat atau sarana untuk membantu manajer menjalankan fungsi-fungsi manajemen sehingga tujuan organisasi dapat

tercapai. Fungsi manajemen, seperti perencanaan (*planning*), pengorganisasian (*organizing*), pengarahan karyawan (*directing*) dan pengendalian (*controlling*) tidak dapat dilakukan tanpa informasi yang memadai. Informasi dalam sebuah organisasi merupakan perekat yang mengikat fungsi-fungsi manajemen dalam sebuah sistem sehingga memungkinkan organisasi bertindak koheran dan harmonis antar berbagai fungsi.

1. Perencanaan (*Planning*)

   Tahap pertama dalam perencanaan adalah mengidentifikasi alternatif-alternatif yang tersedia, kemudian memilih salah satu dari berbagai alternatif tersebut yang paling baik dan cocok dengan tujuan organisasi. Rencana manajemen biasanya diekspresikan dalam sebuah bentuk formal yang disebut dengan anggaran (*budgets*). Oleh karena itu, istilah penganggaran (*budgeting*) sering pula disebut sebagai proses perencanaan. Anggaran biasanya dibuat di bawah kendali dankoordinasi seorang manajer atau kepala bagian akuntansi. Umumnya anggaran dibuat setiap tahun dan mencerminkan rencana manajemen dalam format yang spesifik dan kuantitatif. Anggaran ini dibuat berdasarkan data yang dikumpulkan, dianalisis dan diikhtisarkan oleh akuntan manajemen.

2. Pengorganisasian Sumber-sumber Daya (*Organizing*)

   Langkah selanjutnya adalah pengorganisasian sumber-sumber daya, dimana manajer menetapkan koordinasi antar bagian (*organizationchart*). Perkembangan Pengorganisasian hanya merupakan alat untuk mencapai tujuan yang telah ditentukan.

3. Pengarahan Karyawan (*DirectingandMotivating*)

   Setelah rencana kerja untuk periode mendatang disusun, manajer harus melakukan pemantauan terus menerus terhadap kegiatan sehari-hari dan menjaga agar organisasi dapat berfungsi sebagaimana mestinya. Untuk melakukan hal tersebut, maka manajer harus memiliki kemampuan untuk memotivasi dan mengarahkan karyawan secara efektif.

4. Pengendalian dan Pengawasan (*Controlling*)

   Dalam melaksanakan tugas pengendalian, manajer sebenarnya mencari jaminan bahwa rencana yang telah disusun dalam bentuk anggaran sudah dilaksanakan sebagaimana mestinya (Suprihatmi : 2009 : 57).

II.5. Penjualan

Penjualan adalah interaksi antar individu, saling bertemu muka yang menciptakan, memperbaiki, menguasai atau mempertahankan hubungan pertukaran yang saling menguntungkan dengan pihak lain.

Sistem Informasi Akuntansi Penjualan merupakan suatu sistem yang dapat memberikan informasi tentang hasil dari penjualan, baik itu penjualan tunai maupun kredit. Dengan adanya sistem informasi penjualan, maka pihak manajemen bisa mengambil suatu keputusan mengenai volume penjualan per periode. (Cindy, irwan & Magdalena, 2012 : 3)

II.6. Sejarah Singkat Microsoft Visual Basic 2010 (VB.Net)

Visual Basic 2010 merupakan salah satu bagian dari produk pemrograman terbaru yang dikeluarkan oleh Microsoft, yaitu Microsoft Visual Studio 2010.

Visual Studio merupakan produk andalan dari Microsoft Corporation, dimana didalamnya berisi beberapa jenis IDE pemrograman seperti Visual Basic, Visual C++, Visual Web Developer, Visual C#, dan Visual F#.

Semua IDE tersebut sudah menedukung penuh implementasi .Net *Framework* terbaru, yaitu .Net *Framework 4.0* yang merupakan pengembangan dari .Net *Framework 3.5*. Adapun database standart yang digunakan adalah Microsoft SQL Server 2008 *express*.

Visual Basic 2010 merupakan versi perbaikan dan pengembangan dari versi terdahulunya, yaitu Visual Basic 2008. Beberapa pengembangan yang terdapat didalamnya antara lain dukungan terhadap *library* terbaru dari Microsoft, yaitu .Net *Framework 4.0,* dukungan terhadap pengembangan aplikasi menggunakan Microsoft *SilverLight,* dukungan terhadap aplikasi berbasis *Cloud Computing*, serta perluasan dukungan terhadap *database-database,* baik *standalone* maupun *database server*.

Bahasa Visual Basic 2010 sendiri awalnya berasal dari bahasa pemrograman yang sangat popular di kalangan programer komputer, yaitu bahasa BASIC, yang oleh Microsoft diadaptasi dalam program Microsoft Quick BASIC. Seiring dengan berkembangnya teknologi komputasi dan desain, Microsoft mengeluarkan produk yang dinamakan Microsoft Visual Studio dengan Visual Basic di dalamnya. Saat ini versi Microsoft Visual Studio yang beredar adalah versi 10 yang popular dengan nama Visual Studio 2010, yang di dalamnya termasuk Microsoft Visual Basic 2010. (Andi : 2011 : 2)

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 pemanipulasian 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*/ auto maticteller 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* dibank. 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 (AgustinusMujilan ; 2012 : 23).

II.8. Pengertian MySQL

Mysql pertama kali dirintis oleh seorang programmer *database* bernama Michael Widenius, yang dapat anda hubungi di emailnya *monty@analytikerna*.

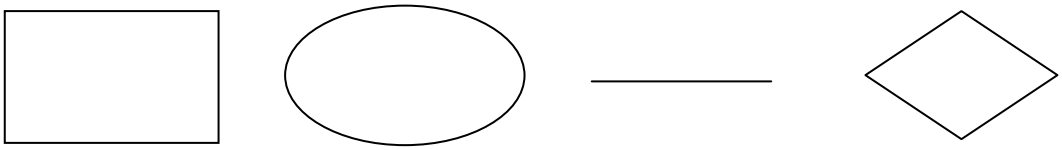
Mysql *database server* adalah RDBMS (*Relasional Database Management System*) yang dapat menangani data yang bervolume besar. meskipun begitu, tidak menuntut *resource* yang besar. Mysql adalah *database* yang paling populer diantara *database* yang lain.

MySQL adalah program *database* yang mampu mengirim dan menerima data dengan sangat cepat dan *multiuser*. MySQL memiliki dua bentuk lisensi, yaitu *freesoftware* 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 publiclicense*), yang dapat anda download pada alamat resminya *http*://*www*.*mysql*.*com*. MySQL sudah cukup lama dikembangkan, beberapa *fase* penting dalam pengembangan MySQL adalah sebagai berikut :

1. MySQL dirilis pertama kali secara internal pada 23 Mei 1995
2. Versi *windows* dirilis pada 8 Januari 1998 untuk *windows* 95 dan *windows* NT.
3. Versi 3.23 : beta dari Juni 2000, dan dirilis pada Januari 2001.
4. Versi 4.0 : beta dari Agustus 2002, dan dirilis pada Maret 2003 (*unions*) (Wahana ; 2010 : 5).

II.9. *Entity Relationship Diagram* (ERD)

Entity Relationship Diagram atau ERD merupakan salah satu alat (tool) berbentuk grafis yang populer untuk *desain database.* Tool ini relatif lebih mudah dibandingkan dengan Normalisasi. Kebanyakan sistem analis memakai alat ini, tetapi yang jadi masalah, kalau kita cermati secara seksama, tool ini mencapai 2NF (Ir. Yuniar Supardi ; 2010 : 448).



Entitas | Atribut/Field | Link (hubungan) | Himpunan Relasi/Interface

Gambar. II.1 Bentuk Simbol ERD
(Sumber : Ir. Yuniar Supardi ; 2010 : 448)

II.10. Kamus Data

Dalam tahapan ini unsur-unsur data yang diperlukan ditentukan untuk selanjutnya dijelaskan lagi dikamus data (*data dictionary*). Pengertian kamus data itu sendiri adalah suatu ensiklopedik dari informasi yang berkaitan dengan data perusahaan atau dapat juga kita katakan bahwa komputer (*computer-based catalog or directory*) yang berisi data perubahan (*metadata*) yang berkenaan dengan tahapan penjelasan data ini adalah system kamus data (*data dictionary system*/DDS) dan bahasa pendeskripsi data (*data description language*/DDL).

Sistem kamus data berbentuk perangkat lunak yang fungsinya adalah penciptaan dan pemerilahraan serta penyediaan kamus data agar dapat digunakan. Kamus data dapat berbentuk kertas ataupun arsip (*file*) komputer. DDS dapat kita

peroleh dalam paket perangkat lunak terpisah ataupun dalam bentuk modul seperti yang ada dalam DBMS (*database management system*) dan CASE (teknik perangkat lunak tambahan komputer/ *computer aided software engineering*) (Ian Soomerville ; 2010 : 344).

II.11. Teknik Normalisasi

Normalisasi adalah teknik perancangan yang banyak digunakan sebagai pemandu dalam merancang basisdata 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 tekah ditemukan.

II.11.1.Bentuk-bentuk Normalisasi

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 padaseluruhkolom yang membentukkunciutama.

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 table berada pada bentuk normal ketiga (3NF) jika table sudah berada pada 2NF dan setiap kolom yang bukan kunci tidak tergantung secara transitif pada kunci utamanya.

4. **BoyceCode 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 table relasional berada pada bentuk normal keempat (4NF) jika didalam BCNF dan semua ketergantungan multivalue merupakan ketergantungan fungsional. Bentuk normal keempat (4NF) didasarkan pada konsep ketergantungan multivalue (MVD). Sebuah table berada pada bentuk normal kelima (5NF) jika ia tidak dapat mempunyai dekomposisi loss less menjadi sejumlah table 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.11.2. 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.

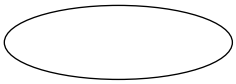
UML merupakan metodologi dalam mengembangkan sistem berorientasi objek dan juga merupakan alat untuk mendukung pengembangan sistem.UML saat ini sangat banyak dipergunakan dalam dunia industri yang merupakan standar bahasa pemodelan umum dalam industri perangkat lunak dan pengembangan sistem.

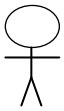
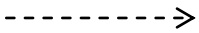
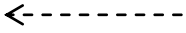
Alat bantu yang digunakan dalam perancangan berorientasi objek berbasiskan UML adalah sebagai berikut :

1. *Use case* Diagram

Use case diagram merupakan pemodelan untuk kelakukan (*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. Adapun Simbol-simbol yang digunakan dalam *usecase* diagram, yaitu :

Tabel II.1 Simbol*Use Case*

Gambar	Keterangan
	*Use case* menggambarkan fungsionalitas yang disediakan sistem sebagai unit yang bertukan pesan antar unit dengan aktor, biasanya dinyatakan dengan menggunakan kata kerja di awal nama *usecase*.

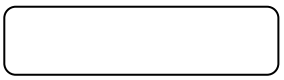
	Aktor adalah *abstraction*dari orang atau sistem yang lain yang mengaktifkan fungsi dari target sistem. Untuk mengidentifikasikan aktor, harus ditentukan pembagian tenaga kerja dan tugas-tugas yang berkaitan dengan peran pada konteks target sistem. Orang atau sistem bisa muncul dalam beberapa peran. Perlu dicatat bahwa aktor berinteraksi dengan *usecase*, 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 *usecase* lain (*required*) atau pemanggilan *use case* oleh *usecase* lain, contohnya adalah pemanggilan sebuah fungsi program.
	*Extend,* merupakan perluasan dari *use case* lain jika kondisi atau syarat terpenuhi.

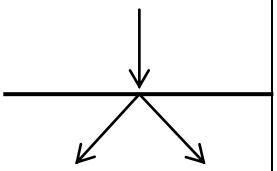
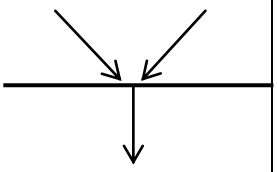
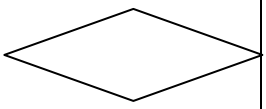
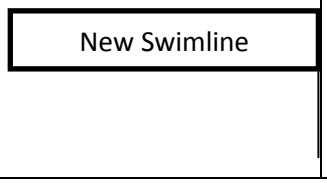
(Sumber : Windu Gata ; 2013 : 4)

2. Diagram Aktivitas (*Activity Diagram*)

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

Tabel II.2Simbol*Activity Diagram*

Gambar	Keterangan
	*Start point,* diletakkan pada pojok kiri atas dan merupakan awal aktifitas.
	*Endpoint,* akhir aktifitas.
	*Activites,* menggambarkan suatu proses/kegiatan bisnis.

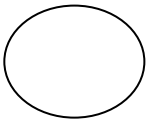
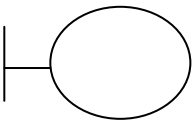
	*Fork*(Percabangan), digunakan untuk menunjukkan kegiatan yang dilakukan secara parallel atau untuk menggabungkan dua kegiatan pararel menjadi satu.
	*Join* (penggabungan) atau rake, digunakan untuk menunjukkan adanya dekomposisi.
	*DecisionPoints*, menggambarkan pilihan untuk pengambilan keputusan, *true*, *false.*
	*Swimlane*, pembagian *activity* diagram untuk menunjukkan siapa melakukan apa.

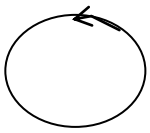
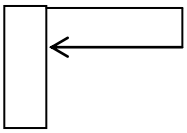
(Sumber : Windu Gata ; 2013 : 6)

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. Adapun simbol-simbol yang digunakan dalam *sequence diagram,* yaitu :

Tabel II.3Simbol*Sequence Diagram*

Gambar	Keterangan
	*EntityClass*, 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 ; 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 tanggng 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)

II.12. *Client-Server*

Client-Server adalah sebuah *arsitektur* jaringan yang memisahkan *client*(biasanya aplikasi yang menggunakan GUI) dengan *server*. Masing-masing *client* dapat meminta data atau informasi dari *server*.

Sistem *client server* didefinisikan sebagai sistem terdistribusi, tetapi ada beberapa perbedaan karakteristik yaitu :

1.*Servis* (layanan)

* Hubungan antara proses yang berjalan pada mesin yang berbeda

* Pemisahan fungsi berdasarkan ide layanannya.

* *Server* sebagai *provider*, *client* sebagai konsumen.

2.*Sharing resources* (sumber daya)

* *Server* bisa melayani beberapa *client* pada waktu yang sama.

3.*Asymmetrical protocol* (protokol yang tidak simetris)

Many-to-one relationship antara *client* dan *server*. *Client* selalu menginisiasikan dialog melalui layanan permintaan, dan *server* menunggu secara pasif *request* dari *client*.

4.Transparansi lokasi

Proses yang dilakukan *server* boleh terletak pada mesin yang sama atau pada mesin yang berbeda melalui jaringan. Lokasi *server* harus mudah diakses dari *client*.

5.*Mix-and-Match*

Perbedaan *server client* platforms

6.Pesan berbasiskan komunikasi

Interaksi *server* dan *client* melalui pengiriman pesan yang menyertakan permintaan dan jawaban.

7.Pemisahan *interface* dan *implementasi*

Server bisa diupgrade tanpa mempengaruhi *client* selama *interface* pesan yang diterbitkan tidak berubah (Janner Simarmata ; 2010 : 109).