

BAB II

TINJAUAN PUSTAKA

II.1. Sistem

Sistem dapat terdiri dari sistem-sistem bagian (*subsystem*). Sebagai misal, sistem komputer dapat terdiri dari subsistem perangkat keras dan subsistem perangkat lunak. Masing-masing subsistem dapat terdiri dari subsistem-subsistem yang lebih kecil atau terdiri dari komponen-komponen. Subsistem perangkat keras (*hardware*) dapat terdiri dari alat masukan, alat pemroses, alat keluaran dan simpanan luar, dan kemudian subsistem-subsistem tersebut akan berinteraksi sedemikian rupa sehingga dapat mencapai satu kesatuan yang terpadu. Dalam buku Analisa dan Design Sistem Informasi Pendekatan Terstruktur Teori dan Praktek Aplikasi Bisnis. “Sistem adalah suatu jaringan kerja dari prosedur-prosedur yang saling berhubungan, berkumpul bersama-sama untuk melakukan suatu kegiatan atau untuk menyelesaikan suatu sasaran yang tertentu ” (Deppi Linda, 2016 : 62).

II.1.1. Karakteristik Sistem

Sistem mempunyai karakteristik atau sifat-sifat tertentu, diantaranya adalah sebagai berikut :

1. Komponen Sistem (*Components*)

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

2. Batas Sistem (*Boundary*)

Batas sistem merupakan daerah yang membatasi antara sistem dengan sistem yang lainnya atau dengan lingkungan luarnya. Batas sistem ini memungkinkan suatu sistem dipandang sebagai suatu kesatuan. Batas sistem menunjukkan ruang lingkup (*scope*) dari sistem tersebut.

3. Lingkungan Luar Sistem (*Environments*)

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

4. Penghubung Sistem (*Interface*)

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

5. Masukan Sistem (*Input*)

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

6. Keluaran sistem (*Output*)

Keluaran adalah hasil energi yang diolah dan diklasifikasikan menjadi keluaran yang berguna. Keluaran dapat berupa masukan untuk subsistem yang lain.

7. Pengolah Sistem (*Process*)

Sistem dapat mempunyai suatu bagian pengolah atau sistem itu sendiri sebagai pengolah. Pengolah yang akan merubah masukan menjadi keluaran.

8. Sasaran Sistem (*Objectives*)

Sistem mempunyai tujuan atau sasaran yang akan menentukan masukan yang dibutuhkan sistem dan keluaran yang akan dihasilkan sistem (Deppi Linda, 2016 : 62).

II.2. Informasi

Informasi adalah data yang diolah menjadi bentuk yang lebih berguna dan lebih berarti bagi yang menerimanya. (1) Akurat (*Accurate*) Berarti informasi harus bebas dari kesalahan-kesalahan dan tidak bias atau menyesatkan. Akurat juga berarti informasi harus jelas mencerminkan maksudnya. (2) Tepat waktu (*Timelines*) Berarti informasi yang datang pada si penerima tidak boleh terlambat. Informasi yang sudah usang tidak akan mempunyai nilai logika karena informasi merupakan landasan dalam pengambilan keputusan. (3) Relevan (*Relevance*) Berarti informasi tersebut bermanfaat bagi pemakainya (Deppi Linda, 2016 : 62-63).

II.3. Sistem Informasi

Sebuah sistem mempunyai tujuan atau sasaran berpendapat sistem adalah sekelompok elemen yang terintegrasi dengan maksud yang sama untuk mencapai suatu tujuan, mendefinisikan sistem sebagai seperangkat elemen-elemen yang terintegrasi dengan maksud yang sama untuk mencapai suatu tujuan bersama.

Sistem informasi merupakan kumpulan dari sub-sub sistem baik fisik maupun non fisik yang berhubungan satu sama lain dan bekerjasama secara harmonis untuk mencapai satu tujuan yaitu mengolah data menjadi informasi yang berarti dan berguna. Berdasarkan dari kedua definisi di atas, dapat disimpulkan bahwa sistem informasi adalah kumpulan dari sub sub sistem dalam organisasi yang mempertemukan kebutuhan pengolahan transaksi harian, mendukung operasi, bersifat manajerial dan kegiatan strategi dari suatu organisasi dan menyediakan pihak luar tertentu dengan laporan-laporan yang diperlukan (Mutammimul Ula:2014).

II.4. 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. Sistem Informasi Akuntansi 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. Sistem Informasi Akuntansi pada umumnya meliputi beberapa siklus pemrosesan transaksi :

1. Siklus pendapatan. Berkaitan dengan pendistribusian barang dan jasa ke entitas lain dan pengumpulan pembayaran-pembayaran yang berkaitan.
2. Siklus pengeluaran. Berkaitan dengan perolehan barang jasa dari entitas lain dan pelunasan kewajiban yang berkaitan.
3. Siklus produksi. Berkaitan dengan pengubahan sumber daya menjadi barang dan jasa.
4. Siklus keuangan. Kejadian-kejadian yang berkaitan dengan perolehan dan manajemen dana-dana modal, termasuk kas. (Agustinus Mujilan : 2012 : 3).

II.5. Suku Bunga Kredit

Suku bunga dapat diartikan sebagai balas jasa yang diberikan oleh bank berdasarkan prinsip konvensional kepada nasabah yang membeli atau menjual produknya. Bunga juga dapat diartikan sebagai harga yang harus dibayar kepada nasabah (yang memiliki simpanan) dengan harga yang harus dibayar oleh nasabah kepada bank (nasabah yang memperoleh pinjaman) (Hedwigis Esti : 2014).

Suku bunga merupakan salah satu faktor yang cukup menarik bagi pemilik dana untuk menyimpan uangnya pada suatu bank. Tingkat suku bunga yang diberikan hendaknya dapat bersaing dengan tingkat suku bunga yang diberikan

bank lain. Tingkat suku bunga biasanya dinyatakan dalam bentuk persentase dari jumlah yang dipinjamkan dan dengan dasar tahunan (annual basis). Dalam kegiatan perbankan sehari-hari ada 2 (dua) macam bunga yang diberikan kepada nasabahnya, yaitu:

1. Bunga Simpanan

Adalah bunga yang diberikan sebagai rangsangan atau balas jasa bagi nasabah yang menyimpan uangnya di bank. Bunga simpanan merupakan harga yang harus dibayar bank kepada nasabahnya. Sebagai contoh: jasa giro, bunga tabungan, dan bunga deposito.

2. Bunga Pinjaman

Adalah bunga yang dibebankan kepada para peminjam atau harga yang harus dibayar oleh nasabah peminjam kepada bank, sebagai contoh bunga kredit.

Suku bunga simpanan dan suku bunga pinjaman merupakan komponen utama faktor biaya dan pendapatan bagi bank. Bunga simpanan merupakan biaya dana yang harus dikeluarkan kepada nasabah, sedangkan bunga pinjaman merupakan pendapatan yang diterima dari nasabah peminjan (debitur). (Hedwigis Esti : 2014).

II.5.1. Faktor-faktor yang Mempengaruhi Suku Bunga

Agar keuntungan yang diperoleh dapat maksimal, maka pihak manajemen bank harus pandai dalam menentukan besar kecilnya komponen suku bunga. Faktor-faktor utama yang mempengaruhi besar kecilnya penetapan suku bunga adalah sebagai berikut:

1. Kebutuhan Dana

Faktor kebutuhan dana dikhususkan untuk dana simpanan, yaitu seberapa besar kebutuhan dana yang diinginkan. Apabila bank kekurangan dana, sementara permohonan pinjaman meningkat, yang dilakukan oleh bank agar dana tersebut cepat terpenuhi adalah dengan meningkatkan suku bunga simpanan. Namun, peningkatan suku bunga simpanan akan pula meningkatkan suku bunga pinjaman. Sebaliknya, apabila dana yang ada dalam simpanan di bank banyak, sementara permohonan pinjaman sedikit, maka bunga simpanan akan turun karena hal ini merupakan beban.

2. Target Laba yang diinginkan

Faktor ini dikhususkan untuk bunga pinjaman. Hal ini disebabkan target laba merupakan salah satu komponen dalam menentukan besar kecilnya suku bunga pinjaman.

3. Kualitas Jaminan

Kualitas jaminan juga diperuntukkan untuk bunga pinjaman. Semakin likuid jaminan (mudah dicairkan) yang diberikan, semakin rendah bunga kredit yang dibebankan dan sebaliknya.

4. Kebijakan Pemerintah

Dalam menentukan baik untuk bunga simpanan maupun bunga pinjaman bank tidak boleh melebihi batasan yang sudah ditetapkan oleh pemerintah.

5. Jangka Waktu

Faktor jangka waktu sangat menentukan. Semakin panjang jangka waktu pinjaman, akan semakin tinggi bunganya, hal ini disebabkan besarnya

kemungkinan resiko macet di masa mendatang. Demikian pula sebaliknya, jika pinjaman berjangka pendek, bunganya relatif rendah.

6. Reputasi Perusahaan

Reputasi perusahaan juga sangat menentukan suku bunga terutama untuk bunga pinjaman. Bonafiditas suatu perusahaan yang akan memperoleh kredit sangat menentukan tingkat suku bunga yang akan dibebankan nantinya, karena biasanya perusahaan yang bonafid kemungkinan resiko kredit macet di masa mendatang relatif kecil.

7. Produk yang Kompetitif

Untuk produk yang kompetitif, bunga kredit yang diberikan relatif rendah jika dibandingkan dengan produk yang kurang kompetitif. Hal ini disebabkan produk yang kompetitif tingkat perputaran produknya tinggi sehingga pembayarannya diharapkan lancar.

8. Hubungan Baik

Biasanya bunga pinjaman dikaitkan dengan faktor kepercayaan kepada seseorang atau lembaga. Dalam praktiknya, bank menggolongkan nasabah antara nasabah utama dan nasabah biasa. Penggolongan ini didasarkan kepada keaktifan serta loyalitas nasabah yang bersangkutan kepada bank. Nasabah yang memiliki hubungan baik dengan bank tentu penentuan suku bunganya pun berbeda dengan nasabah biasa.

9. Persaingan

Dalam kondisi tidak stabil dan bank kekurangan dana, sementara tingkat persaingan dalam memperebutkan dana simpanan cukup ketat, maka bank

harus bersaing keras dengan bank lainnya. Untuk bunga pinjaman, harus berada di bawah bunga pesaing agar dana yang menumpuk dapat tersalurkan, meskipun margin laba mengecil.

10. Jaminan Pihak Ketiga

Dalam hal ini pihak yang memberikan jaminan kepada bank untuk menanggung segala risiko yang dibebankan kepada penerima kredit. Biasanya apabila pihak yang memberikan jaminan bonafide, baik dari segi kemampuan membayar, nama baik, maupun loyalitasnya terhadap bank, bunga yang dibebankan pun juga berbeda (Hedwigis Esti : 2014).

II.6. Bunga Anuitas

Bunga *Flat* adalah sistem perhitungan suku bunga yang besarnya mengacu pada pokok hutang awal. Bunga *flat* biasanya diperuntukkan untuk kredit jangka pendek. Biasanya diterapkan untuk kredit barang konsumsi seperti handphone, home appliances, mobil atau kredit tanpa agunan (KTA). Dengan menggunakan sistem bunga *flat* ini maka porsi bunga dan pokok dalam angsuran bulanan akan tetap sama (Cecep Taufiqurrochman : 2013 : 14)

Rumus perhitungannya adalah :

$$\text{Bunga per bulan} = (P \times i \times t) : J_b$$

P = pokok pinjaman awal,

i = suku bunga per tahun,

t = jumlah tahun jangka waktu kredit,

J_b = jumlah bulan dalam jangka waktu kredit.

Misalnya, Bank Anda memberikan kredit dengan jangka waktu 12 bulan sebesar Rp 12.000.000,00 dengan bunga 12% per tahun (*flat*). Asumsi bahwa suku bunga kredit tidak berubah (tetap) selama jangka waktu kredit. Perhitungan angsuran dijabarkan sebagai berikut:

Tabel II.1. Angsuran Dalam Sistem Bunga Anuitas

| Angsur-an ke | Angsuran Pokok | Angsuran Bunga | Total Angsuran | Sisa Pokok Pinjaman |
|--------------|----------------|----------------|----------------|---------------------|
| 0 | | | | 12.000.000.00 |
| 1 | 1.000.000.00 | 120.000.00 | 1.120.000.00 | 11.000.000.00 |
| 2 | 1.000.000.00 | 120.000.00 | 1.120.000.00 | 10.000.000.00 |
| 3 | 1.000.000.00 | 120.000.00 | 1.120.000.00 | 9.000.000.00 |
| 4 | 1.000.000.00 | 120.000.00 | 1.120.000.00 | 8.000.000.00 |
| 5 | 1.000.000.00 | 120.000.00 | 1.120.000.00 | 7.000.000.00 |
| 6 | 1.000.000.00 | 120.000.00 | 1.120.000.00 | 6.000.000.00 |
| 7 | 1.000.000.00 | 120.000.00 | 1.120.000.00 | 5.000.000.00 |
| 8 | 1.000.000.00 | 120.000.00 | 1.120.000.00 | 4.000.000.00 |
| 9 | 1.000.000.00 | 120.000.00 | 1.120.000.00 | 3.000.000.00 |
| 10 | 1.000.000.00 | 120.000.00 | 1.120.000.00 | 2.000.000.00 |
| 11 | 1.000.000.00 | 120.000.00 | 1.120.000.00 | 1.000.000.00 |
| 12 | 1.000.000.00 | 120.000.00 | 1.120.000.00 | 0.00 |
| Jumlah | 12.000.000.00 | 1.440.000.00 | 13.440.000.00 | |

(Sumber : Taufiqurrochman : 2013 : 14)

II.7. Basis Data (*Database*)

Basis data (*database*) adalah kumpulan dari berbagai data yang saling berelasi atau berhubungan satu dengan yang lainnya, data dinyatakan dengan nilai. Basis data tersimpan di perangkat keras, serta dimanipulasi dengan menggunakan perangkat lunak. Pendefinisian basis data meliputi spesifikasi dari

tipe data, struktur dan batasan dari data atau informasi yang akan disimpan.

Istilah-istilah dalam basis data: (Mutammimul Ula, 2014 : 38).

1. Enterprise: suatu bentuk organisasi seperti Bank, Sekolah, Rumah Sakit, Pabrik, Kantor dan sebagainya.
2. Entitas: suatu objek yang dapat dibedakan dari lainnya yang dapat di wujudkan dalam basis data. Kumpulan dari entitas disebut himpunan entitas.
3. Atribut dan elemen data: karakteristik dari suatu entitas.
4. Record data: kumpulan suatu elemen data yang saling berhubungan.
5. Tabel: kumpulan data atau informasi.

Basis data mempunyai peran penting dalam sistem informasi, yaitu sebagai sumber penyedia data utama untuk memenuhi kebutuhan-kebutuhan informasi seluruh pemakai atau informasi bagi para pengambil keputusan (Mutammimul Ula, 2014 : 38).

II.8. Normalisasi

Normalisasi adalah suatu proses untuk membuat data yang tidak normal menjadi data yang normal. Bentuk data yang tidak normal / data mentah biasa disebut juga *unnormalized form*. Masing – masing level normalisasi mempunyai aturan tersendiri.

1. *First Normal Form*

Suatu tabel dikatakan dalam keadaan *first normal form* (1NF) jika :

- a. Tidak ada perulangan record data dalam tabel.
- b. Setiap sel memiliki satu nilai saja. Artinya tidak ada perulangan group dan array.
- c. Data yang diinputkan memiliki tipe data yang sama dengan tipe data kolom dalam tabel.

2. *Second Normal Form*

Suatu tabel dikatakan dalam keadaan *Second Normal Form* (2NF) jika tabel tersebut sudah dalam keadaan *First Normal Form* (1NF) dan jika semua atribut yang bukan kunci tabel, baik *primary key* maupun *foreign key* tergantung pada semua kunci dalam tabel.

3. *Third Normal Form*

Suatu tabel dikatakan dalam keadaan *third normal form* (3NF) jika tabel tersebut sudah dalam keadaan *second normal form* (2NF) dan jika tidak terdapat ketergantungan yang transitif. Artinya, data-data yang mungkin diisi berulang-ulang dapat dibuat sebuah tabel baru.

4. *Boyce-Codd Normal Form* (BCNF)

Tabel dikatakan dalam keadaan *boyce-codd normal form* (BCNF) jika tabel tersebut dalam keadaan *third normal form* (3NF) dan setiap determinan adalah kunci kandidat.

5. *Fourth Normal Form (4NF)*

Suatu tabel dikatakan dalam keadaan *fourth normal form (4NF)* jika tabel tersebut dalam keadaan *boyce-codd normal form (BCNF)* dan jika tidak terdapat ketergantungan nilai ganda.

6. *Fiveth Normal Form (5NF)*

Tabel dikatakan dalam keadaan *Fiveth Normal Form (5NF)* jika tabel tersebut dalam keadaan *fourth normal form (4NF)* dan jika setiap ketergantungan dalam join ada pada tabel sudah konsekuen dengan kunci kandidat pada tabel tersebut (Ema Utami ; 2012 : 73-76)

II.9. Kamus Data

Kamus data umumnya berguna pada pengembangan model sistem dan dapat digunakan untuk menangani semua informasi dari semua tipe model sistem. Kamus data sederhanya adalah daftar alfabetis dari nama-nama termasuk pada berbagai model sistem. Seperti namanya, kamus harus mencakup deskripsi yang berhubungan dengan entitas bernama tersebut dan, jika nama itu merepresentasikan objek komposit, mungkin saja ada deskripsi mengenai komposisinya. Informasi lain seperti tanggal pembuatan, pembuatnya dan representasi entitas juga dapat dimasukkan, tergantung pada tipe model yang sedang dikembangkan. Keuntungan penggunaan kamus data adalah :

1. Kamus data merupakan mekanisme untuk manajemen nama. Banyak orang yang harus menciptakan nama untuk entitas dan relasi ketika mengembangkan model sistem yang besar. Nama-nama ini harus dipakai secara konsisten dan

tidak boleh bentrok. Perangkat lunak kamus data dapat memeriksa keunikan nama dan memberitahu analisis persyaratan sekiranya terjadi duplikasi nama.

2. Kamus data berfungsi sebagai tempat penyimpanan informasi organisasional yang dapat menghubungkan analisis, desain, implementasi dan evolusi. Sementara sistem dikembangkan, informasi diambil untuk memberitahu perkembangan. Informasi baru ditambahkan pada sistem. Semua informasi mengenai entitas berada pada satu tempat (Ian Sommerville ; 2012 : 151).

Tabel II.2 Entri Kamus Data

| Nama | Keterangan | Tipe | Tanggal |
|--------------|--|---------|------------|
| Has-labels | Relasi 1 N antara entitas bertipe node atau link dan entitas bertipe label | Relasi | 5.10.1998 |
| Label | Berisi informasi terstruktur atau tidak terstruktur mengenai node atau link. Label dipresentasikan oleh ikon (yang bisa berupa kotak transparan) dan teks yang berhubungan | Entitas | 8.10.1998 |
| Link | Relasi 1 : 1 antara entitas desain yang dipresentasikan sebagai node. Link diketikkan dan dapat diberi nama | Relasi | 8.10.1998 |
| Name (Label) | Setiap label memiliki nama yang mengidentifikasi tipe label. Nama harus unik dalam set tipe label yang dipakai pada desain | Atribut | 8.10.1998 |
| Name (Node) | Setiap node harus memiliki nama yang unik di dalam sebuah desain. Panjang nama bisa mencapai 64 karakter. | Atribut | 15.10.1998 |

(Sumber : Ian Sommerville ; 2012 : 151)

II.10. Visual Studio 2010

Microsoft Visual Studio merupakan sebuah perangkat lunak lengkap yang dapat digunakan untuk melakukan pengembangan aplikasi, baik itu aplikasi bisnis, aplikasi personal, ataupun komponen aplikasi lainnya dalam bentuk aplikasi *console*, aplikasi *Windows*, ataupun aplikasi *Web*. Kompiler yang dimasukkan ke dalam paket *Visual Studio* antara lain *Visual C++*, *Visual C#*, *Visual Basic*, *Visual Basic .NET*, *Visual InterDev*, *Visual J++*, *Visual J#*, *VisualFoxPro*, dan *Visual SourceSafe*. *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*).

Selain itu, *Visual Studio* juga dapat digunakan untuk mengembangkan aplikasi *Silverlight*, aplikasi *Windows Mobile* (yang berjalan di atas *.NET Compact Framework*). *Visual Studio* kini telah menginjak versi *Visual Studio 12.0* atau dikenal dengan sebutan *Microsoft Visual Studio 2013* yang diluncurkan pada 17 Oktober 2013 yang ditujukan untuk *platform Microsoft .NET Framework 4.5.1*. Versi sebelumnya *Visual Studio 2012* ditujukan untuk *platform 4.5*, *Visual Studio 2010* ditujukan untuk *.NET Framework 4.0*, *Visual Studio 2008* ditujukan untuk *platform .NET Framework 3.5*, *Visual Studio 2005* ditujukan untuk *platform .NET Framework 2.0* dan *3.0*. *Visual Studio 2003* ditujukan untuk *.NET Framework 1.1*, dan *Visual Studio 2002* ditujukan untuk *.NET Framework 1.0*. Versi-versi tersebut di atas kini dikenal dengan sebutan *Visual Studio.NET*, karena memang membutuhkan *Microsoft .NET Framework*. Sementara itu, sebelum

muncul *Visual Studio .NET*, terdapat *Microsoft Visual Studio 6.0*. (Ahmad Rais Ruli; 2017 : 10)

II.11. SQL Server 2008

SQL (*Structured Query Language*) adalah bahasa non procedural untuk mengakses data pada database relasional. SQL adalah bahasa database yang dipergunakan dalam menyelesaikan permasalahan dalam database serta mempunyai kelebihan dalam mengolah data. Standar SQL mula-mula didefinisikan oleh ISO (*International Standards Organization*) dan ANSI (*the American National Standards Institute*) yang dikenal dengan sebutan SQL86 (Eka Iswandy : 2015 : 73)

Dengan menggunakan SQL, kita dapat melakukan hal-hal berikut:

1. Memodifikasi struktur *database*.
2. Mengubah, mengisi, menghapus isi *database*.
3. Mentransfer data antara database yang berbeda. SQL ada yang dikembangkan untuk PC dan ada juga yang dikembangkan untuk dapat mengakomodasi database yang sangat besar.

Beberapa contohnya antara lain:

1. *Microsoft Access*

Digunakan untuk PC, sangat mudah dipakai dimana perintah SQL dapat langsung dimasukkan atau melalui fasilitas yang telah digunakan.

2. *Microsoft Query*

SQL yang dipaket dengan produk lain dari Microsoft Windows, yaitu Microsoft Visual Studio seperti Visual Basic dan Visual C++. Untuk terhubung dengan database lain menggunakan ODBC.

3. *Oracle*

Digunakan untuk perusahaan yang menggunakan database besar.


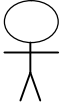



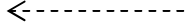
II.12. UML (*Unified Modelling Language*)

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 industry perangkat lunak dan pengembangan sistem. (Windu Gata : 2013)

1. *Diagram Use Case (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.3. Simbol *Use Case Diagram*




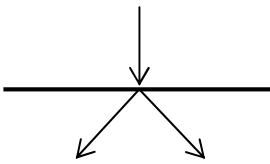
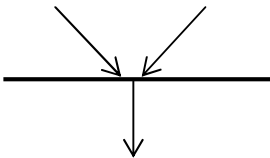
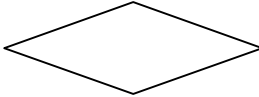

| Gambar | Keterangan |
|---|--|
|  | Use case menggambarkan fungsionalitas yang disediakan sistem sebagai unit-unit yang bertukar pesan antar unit dengan aktor, biasanya dinyatakan dengan menggunakan kata kerja di awal nama use case. |
|  | Aktor adalah <i>abstraction</i> dari orang atau sistem yang 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 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 mengindikasikan aliran data. |
|  | Asosiasi antara aktor dan use case yang menggunakan panah terbuka untuk mengindikasikan bila aktor berinteraksi secara pasif dengan sistem. |
|  | <i>Include</i> , merupakan di dalam use case lain (<i>required</i>) atau pemanggilan use case oleh use case lain, contohnya adalah pemanggilan sebuah fungsi program. |
|  | <i>Extend</i> , merupakan perluasan dari use case lain jika kondisi atau syarat terpenuhi. |

(Sumber : Windu Gata ; 2013:5)

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.4. Simbol Diagram Aktivitas

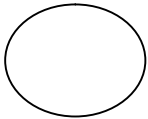
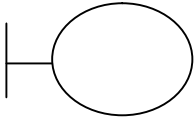
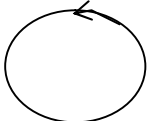

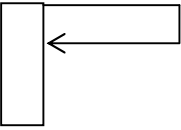


| Gambar | Keterangan |
|---|--|
|  | <i>Start point</i> , diletakkan pada pojok kiri atas dan merupakan awal aktifitas. |
|  | <i>End point</i> , akhir aktifitas. |
|  | <i>Activites</i> , menggambarkan suatu proses/kegiatan bisnis. |
|  | <i>Fork</i> (Percabangan), digunakan untuk menunjukkan kegiatan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

(Sumber : Windu Gata ; 2014:6)

3. 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.5. 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.6. Simbol 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*)