

BAB II

TINJAUAN PUSTAKA

II.1. *Computer Based Information System (CBIS)*

Computer Based Information System (CBIS) atau Sistem Informasi Berbasis Komputer merupakan suatu sistem pengolah data menjadi sebuah informasi yang berkualitas dan dipergunakan untuk suatu alat bantu pengambilan keputusan. Sistem Informasi “berbasis komputer” mengandung arti bahwa komputer memainkan peranan penting dalam sebuah sistem pembangkit informasi. Dengan integrasi yang dimiliki antar subsistemnya, sistem informasi akan mampu menyediakan informasi yang berkualitas, tepat, cepat dan akurat sesuai dengan manajemen yang membutuhkannya. Yulidc (2011, April). Pengertian Computer Based Information System (CBIS), 8. Retrieved Mei 28, 2015, from <http://www.perpuskita.com/cbis/624/>.

II.1.1. Sub Sistem Dari Sistem Informasi Berbasis Komputer

Sub sistem dari sistem informasi berbasis komputer adalah sebagai berikut:

1. Sistem Informasi Akuntansi (SIA)
2. Sistem Informasi Manajemen (SIM)
3. Sistem Pendukung Keputusan (DSS)
4. Automasi Kantor (OA)
5. Sistem Pakar (ES). Yulidc (2011, April). Pengertian Computer Based Information System (CBIS), 8. Retrieved Mei 28, 2015, from <http://www.perpuskita.com/cbis/624/>.

II.1.2. Sistem

Sistem adalah suatu kesatuan prosedur atau komponen yang saling berkaitan satu dengan yang lainnya bekerja bersama sama sesuai dengan aturan yang diterapkan sehingga membentuk suatu tujuan yang sama. dimana dalam sebuah sistem bila terjadi satu bagian saja yang tidak bekerja atau rusak maka suatu tujuan bisa terjadi kesalahan hasilnya atau outputnya (Fitriyani; 2012: 104).

II.1.3. Keputusan

Keputusan adalah suatu reaksi terhadap beberapa solusi alternatif yang dilakukan secara sadar dengan cara menganalisa kemungkinan-kemungkinan dari alternatif tersebut bersama konsekuensinya. Setiap keputusan akan membuat pilihan terakhir, dapat berupa tindakan atau opini. Itu semua bermula ketika kita perlu untuk melakukan sesuatu tetapi tidak tahu apa yang harus dilakukan. Untuk itu keputusan dapat dirasakan rasional atau irrasional dan dapat berdasarkan asumsi kuat atau asumsi lemah (Fitriyani; 2012: 104).

II.1.4. Pengambilan Keputusan

Pengambilan keputusan merupakan proses pemilihan alternatif tindakan untuk mencapai tujuan atau sasaran tertentu. Pengambilan keputusan dilakukan dengan pendekatan sistematis terhadap permasalahan melalui proses pengumpulan data menjadi informasi serta ditambah dengan faktor – faktor yang perlu dipertimbangkan dalam pengambilan keputusan (Fitriyani; 2012: 104-105).

II.1.5. Sistem Pendukung Keputusan (*Decision Support System*)

Sistem pendukung keputusan (*decision support system* atau DSS) adalah sistem informasi berbasis komputer yang menyediakan dukungan informasi

interaktif bagi manajer dan praktisi bisnis selamam proses pengambilan keputusan. Sistem pendukung keputusan menggunakan (1) model analitis, (2) database khusus, (3) penilaian dan pandangan pembuat keputusan, dan (4) proses permodelan berbasis computer yang interaktif untuk mendukung pembuatan keputusan bisnis yang semi terstruktur dan tak terstruktur (Fitriyani; 2012: 105).

II.1.6. Karakteristik Sistem Pendukung Keputusan

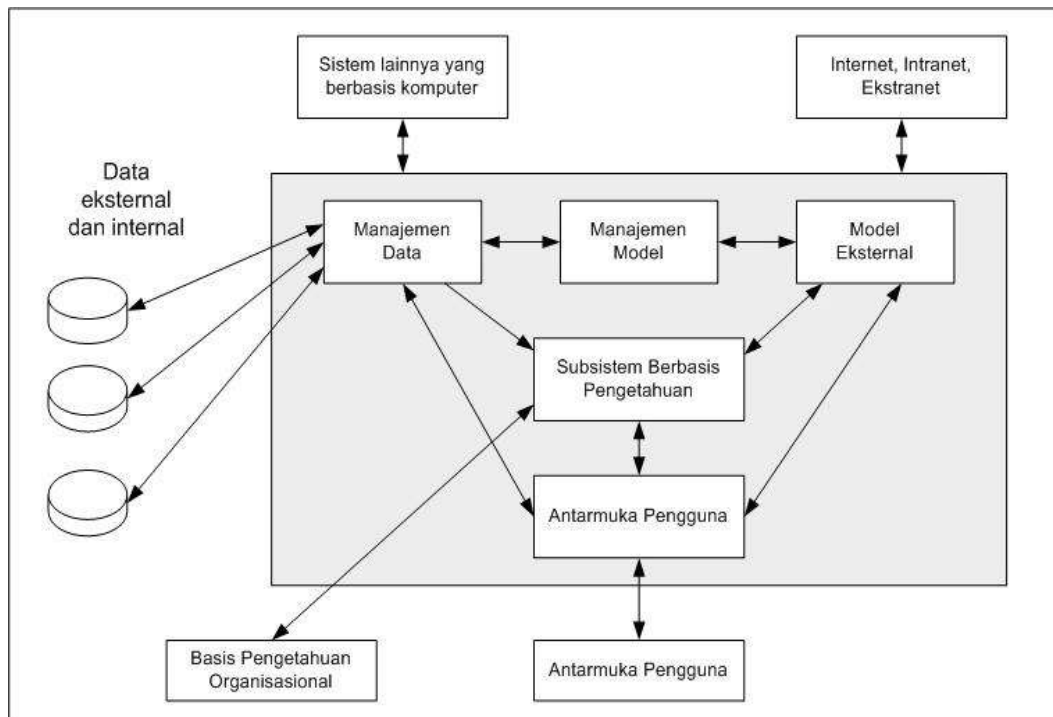
Adapun karakteristik sistem pendukung keputusan yang dimaksud adalah sebagai berikut :

1. Sistem Pendukung Keputusan dirancang untuk membantu pengambil keputusan dalam memecahkan masalah yang sifatnya semi terstruktur ataupun tidak terstruktur dengan menambahkan kebijaksanaan manusia dan informasi komputerisasi.
2. Dalam proses pengolahannya, sistem pendukung keputusan mengkombinasikan penggunaan model-model analisis dengan teknik pemasukan data konvensional serta fungsi-fungsi pencari / interogasi informasi.
3. Sistem Pendukung Keputusan, dirancang sedemikian rupa sehingga dapat digunakan/dioperasikan dengan mudah.
4. Sistem Pendukung Keputusan dirancang dengan menekankan pada aspek fleksibilitas serta kemampuan adaptasi yang tinggi (Fitriyani; 2012: 106).

II.1.7. Komponen Sistem Pendukung Keputusan

Adapun komponen sistem pendukung keputusan yang dimaksud adalah sebagai berikut :

1. *Data Management*. Termasuk database, yang mengandung data yang relevan untuk berbagai situasi dan diatur oleh software yang disebut Database Management Systems (DBMS).
2. *Model Management*. Melibatkan model finansial, statistik, management science, atau berbagai model kuantitatif lainnya, sehingga dapat memberikan ke sistem suatu kemampuan analitis, dan manajemen software yang diperlukan.
3. *Commication (dialog subsystem)*. User dapat berkomunikasi dan memberikan perintah pada DSS melalui subsistem ini. Ini berarti menyediakan antarmuka.
4. *Knowledge Management*. Subsistem optional ini dapat mendukung subsistem lain atau bertindak sebagai komponen yang berdiri sendiri (Fitriyani; 2012: 107).



Gambar II.1. Arsitektur DSS

Sumber : <http://sulung-pd.blogspot.com/2011/03/sistem-pendukung-keputusan.html> (di akses 28 Mei 2015, 5:00 wib)

II.2. *Simple Multi Atribute Raiting Technique (SMART)*

Pada hakekatnya Simple Multy Attribute Rating (SMART) merupakan metode pengambilan keputusan multi kriteria yang dikembangkan oleh Edward pada tahun 1977. Teknik pengambilan keputusan multi kriteria ini didasarkan pada teori bahwa setiap alternative terdiri dari sejumlah kriteria yang memiliki nilai-nilai dan setiap kriteria memiliki bobot yang digunakan untuk menilai setiap alternatif agar diperoleh alternatif terbaik. SMART lebih banyak digunakan karena kesederhanaannya dalam merespon kebutuhan pembuatan keputusan dan caranya dalam menganalisa respon. (Atiqha; 2013: 75).

Peralatan utama dari model ini adalah sebuah hirarki fungsional dengan input utamanya adalah persepsi manusia. Jadi pada dasarnya tidak ada perbedaan yang mencolok pada model SMART dengan model lainnya sama sama terletak pada jenis inputnya, hanya saja terdapat persentase dari setiap pembobotan langsung yang ditentukan oleh hasil analisa permasalahan (Atiqha; 2013: 76).

Pembobotan pada SMART menggunakan skala 0 sampai 1, sehingga mempermudah perhitungan dan perbandingan nilai pada masing-masing alternatif. Model perhitungan nilai utilitas terhadap semua alternatif berdasarkan setiap kriteria menggunakan formula SMART. Formula yang digunakan dalam SMART adalah $\text{Maximize } \sum_{j=1}^k w_j \cdot u_{ij}$, dimana w_j adalah nilai bobot dan u_{ij} adalah nilai utilitas dari setiap alternatif.

II.2.1. Teknik SMART

Adapun teknik atau langkah-langkah dalam proses SMART, antara lain :

1. Identifikasi calon nasabah yang nantinya bertanggung jawab dalam mengambil keputusan.
2. Identifikasi permasalahan yang ada dengan melihat akar permasalahan dan batasan-batasan yang ada agar nantinya tidak menyimpang dari tujuan yang ingin dicapai.
3. Identifikasi alternatif yang ada untuk mencapai tujuan dari sistem yang akan dibuat.
4. Identifikasi kriteria-kriteria yang akan mendukung pengambilan keputusan

5. Memberikan peringkat atau bobot untuk setiap kriteria. Pemberian peringkat atau bobot ini telah ditentukan dan sistem akan memberikan bobot secara langsung dengan nilai bobot yang telah ditentukan.
6. Memberikan penilaian setiap kriteria untuk setiap alternatif. dilakukan oleh calon nasabah menurut persepsinya dimana penilaian setiap kriteria dapat bersifat kualitatif maupun kuantitatif untuk setiap alternatifnya.
7. Mengembangkan *Single-Attribute Utilities* yang mencerminkan seberapa baik setiap alternatif dilihat dari setiap kriteria. Tahap ini adalah memberikan suatu nilai pada semua kriteria untuk setiap alternatif dengan nilai yang berskala 10 sampai 100. (Atiqha; 2013: 75-76).

II.3. Microsoft Visual Basic 2010

Visual basic adalah sebuah bahasa pemrograman yang berpusat pada object (Object Oriented Programming) digunakan dalam pembuatan aplikasi Windows yang berbasis *Graphical User Interface*, hal ini menjadikan Visual Basic menjadi bahasa pemrograman yang wajib diketahui dan dikuasai oleh setiap programmer. Beberapa karakteristik obyek tidak dapat dilakukan oleh Visual Basic misalnya seperti *Inheritance* tidak bisa module dan *Polymorphism* secara terbatas bisa dilakukan dengan deklarasi *class module* yang mempunyai *Interface* tertentu. Sifat Visual Basic tidak case sensitive (Muhammad Sadeli; 2013: 19).

II.3.1. Objek-Objek Yang Digunakan

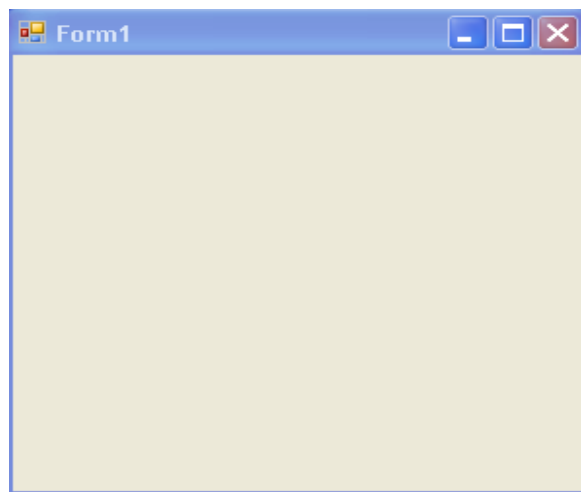
Adapun objek-objek yang dipergunakan dalam program ini adalah :

1. Jendela Utama

Jendela utama terdiri dari baris judul, menu, dan toolbar. Baris judul berisi nama proyek dan nama form yang aktif. Juga terlihat kata yang diapit dengan lambang *bracket*(`[]`) satu diantara tiga: *Design*, *Run*, atau *Break* yang menunjukkan mode operasi *VB .Net*.

2. Jendela Desain

Jendela desain merupakan pusat pengembangan aplikasi *VB .Net*. Dibagian atas jendela terdapat tab yang berisi form dengan nama NamaForm.vb [Design]. Form adalah tempat pemakai menggambarkan aplikasi. Kode yang berhubungan dengan form terdapat pada tab NamaForm.vb. Bentuk Jendela desain seperti terlihat pada Gambar II.2.



Gambar II.2. Jendela Desain
(Sumber : Muhammad Sadeli ; 2012: 10)

3. Toolbox

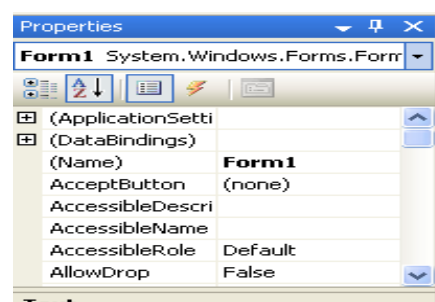
Toolbox berisi kontrol yang akan digunakan pada aplikasi Anda. *Toolbox* akan aktif jika ada form dalam jendela desain. Sebagaimana terlihat pada Gambar II.3.



Gambar II.3. Jendela Toolbox
(Sumber : Muhammad Sadeli ; 2012: 11)

4. Jendela Properties

Jendela *properties* digunakan pada mode desain untuk mengatur suatu nilai pada objek (kontrol). Tampilan jendela *properties* dapat dilihat pada Gambar II.4.



Gambar II.4. Jendela Properties
(Sumber : Muhammad Sadeli ; 2012: 109)

II.4. Basis Data (*Database*)

Basis Data (*Database*) adalah kumpulan data yang saling berelasi. Data sendiri merupakan fakta mengenai obyek, dan orang. Data dinyatakan dengan nilai (angka, deretan karakter atau simbol). Basis data dapat didefinisikan dalam berbagai sudut pandang seperti himpunan kelompok data yang saling berhubungan yang diorganisasikan sedemikian rupa, sehingga kelak dapat dimanfaatkan dengan cepat dan mudah, kumpulan data yang saling berhubungan yang disimpan secara bersama sedemikian rupa tanpa pengulangan (*redundancy*) yang tidak perlu, untuk memenuhi kebutuhan, Kumpulan *file/table/arsip* yang saling berhubungan yang disimpan dalam media penyimpanan elektronik (Janner, S. dan Iman, P; 2010: 56).

II.4.1. Tujuan Basis Data (*Database*)

Tujuan Basis data (*database*) adalah untuk mengatur data sehingga diperoleh kemudahan, ketepatan, dan kecepatan dalam pengambilan kembali. Untuk mencapai tujuan, syarat sebuah basis data (*database*) yang baik adalah sebagai berikut :

1. Tujuan adanya redundansi dan inkonsistensi data

Redundansi terjadi jika suatu informasi disimpan dibeberapa tempat.

2. Kesulitan pengaksesan data

Basis data (*database*) memiliki fasilitas untuk melakukan pencarian informasi dengan menggunakan *query* ataupun dari *tool* untuk melihat tablenya.

3. *Multiple User*

Basis Data (*Database*) memungkinkan pengguna data bersama-sama oleh banyak pengguna pada saat yang bersamaan atau pada saat yang berbeda (Janner, S. dan Iman, P; 2010: 60).

II.4.2. Manfaat atau Kelebihan Basis Data (*Database*)

Banyak manfaat atau kelebihan yang kita peroleh dengan menggunakan basis data (*database*) adalah sebagai berikut :

1. Kecepatan dan kemudahan (*speed*)

Dengan menggunakan basis data (*database*) pengambilan informasi dapat dilakukan dengan cepat dan mudah.

2. Kebersamaan pemakai

Sebuah basis data (*database*) dapat digunakan oleh banyak *user* dan banyak aplikasi. Untuk banyak data-data yang diperlukan oleh banyak orang/bagian.

3. Pemusatan *control* data

Karena cukup dengan satu basis data (*database*) untuk banyak keperluan, pengontrolan terhadap data juga cukup dilakukan disatu tempat saja.

4. Efisiensi ruang penyimpanan (*space*)

Dengan pemakaian bersama, kita tidak perlu menyediakan tempat penyimpana diberbagai tempat, tetapi cukup satu saja sehingga ini akan menghemat ruang penyimpanan data yang dimiliki oleh sebuah organisasi.

5. Keakuratan (*Accuracy*)

Penerapan secara ketat ataupun tipe data, *domain* data, keunikan data, hubungan antara data dan lain-lain, dapat menekan keakuratan dalam pemasukan/penyimpanan data.

6. Ketersediaan (*Availability*)

Dengan basis data (*database*) kita dapat mem-*backup* data, memilah-milah data mana yang masih diperlukan dan data mana yang perlu kita simpan ditempat lain.

7. Keamanan (*Security*)

Kebanyakan DBMS dilengkapi dengan fasilitas manajemen pengguna diberikan hak akses yang berbeda-beda sesuai dengan pengguna dan posisinya.

8. Kemudahan dalam pembuatan program aplikasi baru

Pengguna basis data (*database*) merupakan bagian dari perkembangan teknologi. Dengan adanya basis data (*database*) pembuatan aplikasi bisa memanfaatkan kemampuan dari DBMS, sehingga pembuatan aplikasi tidak perlu mengurus penyimpanan data, tetapi cukup mengatur *interface* untuk pengguna.

9. Pemakaian secara langsung

Basis data (*database*) memiliki fasilitas untuk melihat datanya secara langsung dengan *tool* yang disediakan oleh DBMS.

10. Kebebasan data (*data independence*)

Jika sebuah program telah selesai dibuat, dan ternyata ada perubahan isi/struktur data. Maka dengan basis data (*database*), perubahan ini hanya perlu dilakukan pada level DBMS tanpa harus membongkar kembali program aplikasinya.

11. *User View*

Basis data (*database*) penyediaan pandangan yang berbeda-beda untuk tiap pengguna (Janner, S. dan Iman, P; 2010: 61).

II.4.3. Normalisasi

Normalisasi merupakan proses pengelompokkan elemen data menjadi tabel-tabel yang menunjukkan entitas dan relasinya. Proses ini selalu diuji pada beberapa kondisi (Janner, S. dan Iman, P; 2010: 79).

Pada proses normalisasi ini perlu dikenal dahulu definisi dari tahapan normalisasi adalah sebagai berikut :

1. Bentuk Tidak Normal (*Unnormalized Form*).

Bentuk ini merupakan kumpulan data yang akan direkam, tidak ada keharusan untuk mengikuti suatu format tertentu, dapat saja data tidak lengkap atau terduplikasi. Data dikumpulkan apa adanya sesuai dengan kedatangannya.

2. Bentuk Normal (1NF/*First Normal Form*)

Bentuk normal kesatu mempunyai ciri yaitu bahwa setiap data dibentuk dalam *flat file* (*file* datar/rata), data dibentuk dalam satu *record* demi *record* dan nilai dari *field* berupa *atomic value*.

3. Bentuk Normal Kedua (2NF/*Second Normal Form*)

Bentuk normal kedua mempunyai syarat yaitu bentuk data telah memenuhi kriteria untuk bentuk normal kesatu.

4. Bentuk Normal Ketiga (3NF/*Third Normal Form*)

Untuk menjadi bentuk normal ketiga maka relasi haruslah dalam bentuk normal kedua dan semua atribut bukan primer tidak punya hubungan yang transitif.

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

Boyce-Codd Normal Form mempunyai paksaan yang lebih kuat dari bentuk normal ketiga. Untuk menjadi BCNF, relasi harus dalam bentuk normal kesatu dan setiap atribut harus bergantung fungsi pada atribut *superkey* (Janner, S. dan Iman, P; 2010: 79-84).

II.5. Microsoft SQL Server 2008

SQL Server adalah sebuah *database* relasional yang dirancang untuk mendukung aplikasi dengan arsitektur *client/server* dimana *database* terdapat pada komputer pusat yang disebut *server*, dan informasi digunakan bersama – sama oleh beberapa user yang menjalankan aplikasi didalam komputer lokalnya yang disebut dengan *client* arsitektur semacam ini memberikan integritas data yang tinggi karena semua *user* bekerja dengan informasi yang sama.

Melalui aturan aturan bisnis, kendali diterapkan kepada semua user mengenai informasi yang ditambahkan ke dalam *database*. Database SQL Server

dibagi kedalam beberapa komponen logikal, seperti misalnya tabel, *view*, dan elemen–elemen lain yang terlihat oleh user (Muhammad Sadeli; 2012: 89).

II.6. UML (*Unifed Modeling Language*)

Hasil pemodelan pada OOAD terdokumentasikan dalam bentuk *unifed modeling language* (UML). UML adalah bahasa spesifikasi standar yang dipergunakan untuk mendokumentasikan, menspesifikasikan dan membangun perangkat lunak.

UML merupakan metodologi dalam mengembangkan sisitem 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 perngkat lunak dan pengembangan sistem (Windu Gata. dan Grace Gata; 2013: 4).

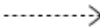
II.6.1. Alat Bantu Berbasis UML

Alat bantu yang digunakan dalam perancangan berorientasi objek berbasiskan 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 dalam sistem informasi dan siapa saja yang berhak menggunakan fungsi-fungsi tersebut. Simbol-simbol yang digunakan dalam Use Case Diagram dapat dilihat pada table II.1.

Table II.1. Use Case Diagram

NO	GAMBAR	NAMA	KETERANGAN
1		<i>Actor</i>	Menspesifikasikan himpunan peran yang pengguna mainkan ketika berinteraksi dengan <i>use case</i> .
2		<i>Dependency</i>	Hubungan dimana perubahan yang terjadi pada suatu elemen mandiri (<i>independent</i>) akan mempengaruhi elemen yang bergantung padanya elemen yang tidak mandiri (<i>independent</i>).
3		<i>Generalization</i>	Hubungan dimana objek anak (<i>descendent</i>) berbagi perilaku dan struktur data dari objek yang ada di atasnya objek induk (<i>ancestor</i>).
4		<i>Include</i>	Menspesifikasikan bahwa <i>use case</i> sumber secara <i>eksplisit</i> .
5		<i>Extend</i>	Menspesifikasikan bahwa <i>use case</i> target memperluas perilaku dari <i>use case</i> sumber pada suatu titik yang diberikan.
6		<i>Association</i>	Apa yang menghubungkan antara objek satu dengan objek lainnya.
7		<i>System</i>	Menspesifikasikan paket yang menampilkan sistem secara terbatas.
8		<i>Use Case</i>	Deskripsi dari urutan aksi-aksi yang ditampilkan sistem yang menghasilkan suatu hasil yang terukur bagi suatu aktor
9		<i>Collaboration</i>	Interaksi aturan-aturan dan elemen lain yang bekerja sama untuk menyediakan perilaku yang lebih besar dari jumlah dan elemen-elemennya (<i>sinergi</i>).
10		<i>Note</i>	Elemen fisik yang eksis saat aplikasi dijalankan dan mencerminkan suatu sumber daya komputasi

(Sumber : Windu Gata dan Grace Gata ; 2013: 4-6)

2. *Activity Diagram* (Diagram Aktivitas)

Activity diagram menggambarkan workflow (aliran kerja) atau aktivitas dari sebuah sistem atau proses bisnis. Simbol-simbol yang digunakan dalam activity diagram dapat dilihat pada table II.2.

Table II.2. *Activity Diagram*

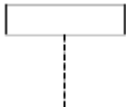
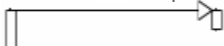
NO	GAMBAR	NAMA	KETERANGAN
1		<i>Activity</i>	Memperlihatkan bagaimana masing-masing kelas antarmuka saling berinteraksi satu sama lain
2		<i>Action</i>	State dari sistem yang mencerminkan eksekusi dari suatu aksi
3		<i>Initial Node</i>	Bagaimana objek dibentuk atau diawali.
4		<i>Activity Final Node</i>	Bagaimana objek dibentuk dan dihancurkan
5		<i>Fork Node</i>	Satu aliran yang pada tahap tertentu berubah menjadi beberapa aliran

(Sumber : Windu Gata dan Grace Gata ; 2013: 6-7)

3. *Sequence Diagram* (Diagram Urutan)

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 table II.3.

Table II.3. Sequence Diagram

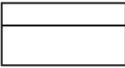
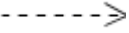
NO	GAMBAR	NAMA	KETERANGAN
1		<i>LifeLine</i>	Objek <i>entity</i> , antarmuka yang saling berinteraksi.
2		<i>Message</i>	Spesifikasi dari komunikasi antar objek yang memuat informasi-informasi tentang aktifitas yang terjadi
3		<i>Message</i>	Spesifikasi dari komunikasi antar objek yang memuat informasi-informasi tentang aktifitas yang terjadi

(Sumber : Windu Gata dan Grace Gata ; 2013: 7-8)

4. Class Diagram (Diagram Kelas)

Merupakan hubungan antar kelas dan penjelasan detail tipe-tiap kelas di dalam mode 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. Simbol-simbol yang digunakan dalam class diagram dapat dilihat pada Table II.4.

Table II.4. *Class Diagram*

NO	GAMBAR	NAMA	KETERANGAN
1		<i>Generalization</i>	Hubungan dimana objek anak (<i>descendent</i>) berbagi perilaku dan struktur data dari objek yang ada di atasnya objek induk (<i>ancestor</i>).
2		<i>Nary Association</i>	Upaya untuk menghindari asosiasi dengan lebih dari 2 objek.
3		<i>Class</i>	Himpunan dari objek-objek yang berbagi atribut serta operasi yang sama.
4		<i>Collaboration</i>	Deskripsi dari urutan aksi-aksi yang ditampilkan sistem yang menghasilkan suatu hasil yang terukur bagi suatu aktor
5		<i>Realization</i>	Operasi yang benar-benar dilakukan oleh suatu objek.
6		<i>Dependency</i>	Hubungan dimana perubahan yang terjadi pada suatu elemen mandiri (<i>independent</i>) akan memengaruhi elemen yang bergantung padanya elemen yang tidak mandiri
7		<i>Association</i>	Apa yang menghubungkan antara objek satu dengan objek lainnya

(Sumber : Windu Gata dan Grace Gata ; 2013: 9)

II.7. Bank

Bank adalah sebuah lembaga intermediasi keuangan umumnya didirikan dengan kewenangan untuk menerima simpanan uang, meminjamkan uang, dan menerbitkan promes atau yang dikenal sebagai banknote. Kata *bank* berasal dari bahasa Italia *banca* berarti tempat penukaran uang. Sedangkan menurut Undang-undang Negara Republik Indonesia Nomor 10 Tahun 1998 Tanggal 10 November 1998 tentang perbankan, yang dimaksud dengan bank adalah badan usaha yang menghimpun dana dari masyarakat dalam bentuk simpanan dan menyalurkannya kepada masyarakat dalam bentuk kredit dan atau bentuk-bentuk lainnya dalam rangka meningkatkan taraf hidup rakyat banyak (Adli Nazrian dan Paidi Hidayat; 2012: 15-16).

Bagi masyarakat awam mungkin masih banyak beranggapan bahwa bank hanya sebagai tempat untuk menabung dan meminjam uang. Kenyataan bank banyak menawarkan jasa-jasa lainnya disamping sekedar menawarkan jasa penyimpanan uang/tabungan dan meminjamkan uang/kredit. Berbeda halnya dengan orang yang memang mengerti perbankan dalam menafsirkan pengertian bank itu sendiri (Adli Nazrian dan Paidi Hidayat; 2012: 15).

Pengertian kata “konvensional” menurut Kamus Umum Bahasa Indonesia adalah “menurut apa yang sudah menjadi kebiasaan”. Sementara itu, menurut Kamus Besar Bahasa Indonesia (KBBI) adalah “berdasarkan kesepakatan umum” seperti adat, kebiasaan, kelaziman. Berdasarkan pengertian itu, bank konvensional adalah bank yang dalam operasionalnya menerapkan metode bunga, karena metode bunga sudah ada terlebih dahulu, menjadi kebiasaan dan telah dipakai

secara meluas dibandingkan dengan metode bagi hasil. (Adli Nazrian dan Paidi Hidayat; 2012: 17-18).

Bank milik pemerintah merupakan bank yang akte pendiriannya maupun modal bank ini sepenuhnya dimiliki oleh pemerintah, sehingga keuntungannya dimiliki oleh pemerintah pula. Contoh bank milik pemerintah adalah Bank Mandiri, Bank Negara Indonesia (BNI), Bank Rakyat Indonesia (BRI), dan Bank Tabungan Negara (BTN). Selain itu ada juga bank milik pemerintah daerah yang terdapat di daerah tingkat I dan tingkat II masing-masing provinsi seperti Bank Sumut (Adli Nazrian dan Paidi Hidayat; 2012: 18-19).