

BAB II

LANDASAN TEORI

II.1. Sistem

Menurut Kusrini (2010:5), kata sistem mempunyai beberapa pengertian, tergantung dari sudut mana kata tersebut didefinisikan. Secara garis besar ada dua pendekatan yang dilakukan yaitu :

- a. Pendekatan sistem yang lebih menekankan pada elemen-elemen atau kelompoknya, yang didalam hal ini sistem ini didefinisikan sebagai suatu jaringan kerja dari prosedur-prosedur yang saling berhubungan, berkumpul bersama-sama untuk melakukan suatu kegiatan atau untuk menyelesaikan suatu aturan tertentu.
- b. Pendekatan sistem sebagai jaringan kerja dari prosedur, yang lebih menekankan urutan operasi didalam sistem. Prosedur didefinisikan sebagai urutan operasi kerja (tulis-menulis), yang biasanya melibatkan beberapa orang didalam satu atau lebih departemen yang diterapkan untuk menjamin penanganan yang seragam dari transaksi bisnis yang terjadi.

Pendekatan sistem yang lebih menekankan pada elemen-elemen atau komponennya mendefinisikan sistem sebagai sekumpulan elemen-elemen yang saling terkait atau terpadu yang dimaksudkan untuk mencapai suatu tujuan. Dengan demikian didalam suatu sistem, komponen-komponen ini tidak dapat berdiri sendiri, tetapi sebaliknya, saling berhubungan hingga membentuk suatu kesatuan hingga tujuan sistem dapat tercapai.

Menurut Jogiyanto (2008:34), sistem (*system*) dapat didefinisikan dengan pendekatan prosedur dan dengan pendekatan komponen. Dengan pendekatan prosedur, sistem dapat didefinisikan sebagai kumpulan dari prosedur - prosedur yang mempunyai tujuan tertentu. Contoh sistem yang didefinisikan dengan pendekatan prosedur ini yaitu sistem akuntansi. Sistem ini didefinisikan sebagai kumpulan dari prosedur-prosedur penerimaan kas, pengeluaran kas, penjualan kas, dan buku besar.

Dengan pendekatan komponen, sistem dapat didefinisikan sebagai kumpulan dari komponen yang saling berhubungan satu dengan yang lainnya membentuk satu kesatuan untuk mencapai tujuan tertentu. Contoh sistem yang didefinisikan dengan pendekatan ini misalnya adalah sistem komputer yang didefinisikan sebagai kumpulan dari perangkat keras dan perangkat lunak.

II.1.1 Karakteristik Sistem

Menurut Kusri (2010:6), Sistem mempunyai beberapa karakteristik atau sifat-sifat tertentu antara lain :

a. Komponen sistem (*Components*)

Suatu sistem terdiri dari sejumlah komponen yang saling berinteraksi, artinya saling bekerja sama untuk membentuk suatu kesatuan. Komponen-komponen sistem atau elemen-elemen sistem dapat berupa suatu subsistem atau bagian dari sistem. Setiap sistem tidak peduli berapapun kecilnya, selalu mengandung komponen-komponen atau subsistem.

b. Batasan sistem (*Boundary*)

Batas sistem merupakan daerah yang membatasi antara suatu sistem dengan sistem yang lainnya atau dengan lingkungan luarnya.

c. Lingkungan luar sistem (*Environment*)

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.

d. Penghubung sistem (*Interface*)

Merupakan media penghubung antara satu subsistem dengan subsistem yang lainnya. Melalui perhubungan ini memungkinkan sumber-sumber daya mengalir dari subsistem yang lainnya.

e. Masukan sistem (*Input*)

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

f. Keluaran sistem (*Output*)

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

g. Pengolah sistem (*Process*)

Suatu sistem yang dapat mempunyai suatu bagian pengolah yang akan merubah masukan menjadi keluaran.

h. Sasaran sistem (*Objective*)

Suatu sistem pasti mempunyai tujuan atau sasaran. Kalau suatu sistem tidak mempunyai sasaran, maka operasi sistem tidak ada gunanya. Suatu sistem dikatakan berhasil bila mengenai sasaran atau tujuannya.

II.1.2. Klasifikasi Sistem

Menurut Kusrini (2010:7), Sistem dapat diklasifikasikan dari beberapa sudut pandangan, diantaranya sebagai berikut :

1. Sistem abstrak dan sistem fisik.

Sistem abstrak adalah sistem yang berisi gagasan atau konsep. Misalnya, sistem teologi yang berisi gagasan tentang hubungan manusia dan Tuhan. Sistem fisik merupakan sistem yang secara fisik dapat dilihat. Misalnya sistem komputer, sistem sekolah, sistem akuntansi, dan sistem transportasi.

2. Sistem alamiah dan sistem buatan manusia.

Sistem alamiah adalah sistem yang terjadi karena alam (tidak dibuat manusia). Misalnya, sistem tata surya. Sistem buatan manusia adalah sistem yang dibuat oleh manusia. Misalnya, sistem komputer dan sistem mobil.

3. Sistem tertentu dan sistem tak tentu.

Sistem tertentu adalah sistem yang operasinya dapat diprediksi secara tepat. Misalnya, sistem komputer. Sistem tak tentu adalah sistem yang tak dapat diramal dengan pasti karena mengandung unsur probabilitas. Misalnya, sistem arisan dan sistem sediaan.

4. Sistem tertutup dan sistem terbuka

Sistem tertutup adalah sistem yang tidak bertukar materi, informasi, atau energi dengan lingkungan. Misalnya, reaksi kimia dalam tabung terisolasi.

Sistem terbuka adalah sistem yang berhubungan dengan lingkungan dan dipengaruhi oleh lingkungan.

II.1.3 Sistem Informasi

Menurut Kusrini (2010:8), sistem informasi adalah suatu sistem dalam suatu organisasi yang mempertemukan kebutuhan pengolahan transaksi harian, mendukung operasi bersifat manajerial dan kegiatan strategi dari suatu organisasi dan menyediakan pihak luar tertentu dan laporan-laporan yang diperlukan.

Defenisi umum sistem informasi adalah suatu sistem yang terdiri atas rangkaian sub system informasi terhadap pengolahan data untuk menghasilkan informasi yang berguna dalam pengambilan keputusan.

II.2. Sistem Informasi Berbasis Komputer

Kusrini (2007:13), suatu sistem diama suatu organisasi yang mempertemukan kebutuhan pengolahan transaksi harian, mendukung operasi, bersifat manajerial, dan merupakan kegiatan strategi dari suatu organisasi, serta menyediakan laporan – laporan yang diperlukan oleh pihak luar.

Berdasarkan dukungan kepada pemakainya, sistem informasi dibagi menjadi :

1. Sistem Pemrosesan Transaksi (*Tnsaction Processing System* atau TPS)
2. Sistem Informasi Manajemen (*Management Information System* atau SIM)
3. Sistem Otomasi Perkantoran (*Office Automation System* atau OAS)

4. Sistem Pendukung Keputusan (*Decision Support System* atau DSS)
5. Sistem Informasi Eksekutif (*Executive Information System* atau EIS)
6. Sistem Pendukung Kelompok (*Group Support System* atau GSS)
7. Sistem Pendukung Cerdas (*Intelligent Support System* atau ISS)

Kusrini (2007:13).

II.2.1. Sistem Pendukung Keputusan

Menurut Jogiyanto (2008:327), sistem penunjang keputusan (SPK) atau *decision support system* (DSS) didefinisikan sebagai sebagai suatu sistem informasi untuk membantu manajer level menengah untuk proses pengambilan keputusan setengah terstruktur (*semi structured*) supaya lebih efektif dengan menggunakan model-model analitis dan data yang tersedia.

II.2.2. Tujuan Sistem Pendukung Keputusan

Menurut Jogiyanto (2008:327), tujuan dari SPK (sistem pendukung keputusan) adalah sebagai berikut :

1. Membantu manajer mengambil keputusan setengah terstruktur yang dihadapi oleh manajer level menengah.
2. Membantu atau mendukung manajemen mengambil keputusan bukan menggantikannya.
3. Meningkatkan efektivitas pengambilan keputusan manajemen bukan untuk meningkatkan efisiensi. Walaupun waktu manajemen penting (efisiensi), tetapi efektivitas merupakan tujuan utama SPK.

II.2.3. Komponen Sistem Pendukung Keputusan

Menurut Jogiyanto (2008:329), sistem penunjang keputusan (SPK) mempunyai 5 komponen utama yaitu :

1. Dialog manajemen (komponen input dan output)

Yaitu komponen untuk berdialog dengan pemakai sistem. Komponen ini di dalam sistem informasi merupakan komponen input dan komponen output.

2. Model manajemen (komponen model)

Yaitu komponen yang merubah data menjadi informasi yang relevan. Model – model yang banyak digunakan di sistem penunjang keputusan adalah model matematik optimisasi seperti misalnya *liniear programming*, *dynamic programming*, dan lain sebagainya.

3. Data management (komponen basis data)

Yaitu komponen basis data yang terdiri dari semua basis data yang dapat diakses. Seperti halnya sistem informasu pada umumnya mempunyai komponen lain yaitu komponen teknologi dan control.

4. Komponen teknologi terdiri perangkat keras dan perangkat lunak. Perangkat lunak spesifik yang digunkaan oleh sistem penunjang keputusan adalah *spreadsheet*, *database management*, dan bahasa query (*query language*).

Komponen teknolog

5. Komponen kontrol

II.2.4. Arsitektur Sistem Pendukung Keputusan

Menurut Kusrini (2007:25), aplikasi sistem pendukung keputusan bisa terdiri dari subsistem, yaitu :

1. Subsistem manajemen data

Subsistem manajemen data memasukkan satu database yang berisi data yang relevan untuk suatu situasi yang dikelola oleh perangkat lunak yang disebut sistem manajemen database (*DBMS/Data Base Management System*). Subsistem manajemen data bisa diinterkoneksi dengan data *warehouse* perusahaan, suatu repository untuk data perusahaan yang relevan dengan pengambilan keputusan.

2. Subsistem manajemen model

Merupakan paket perangkat lunak yang memasukkan model keuangan, statistic, ilmu manajemen atau, atau model kuantitatif lain yang memberikan kapabilitas analitik dan manajemen perangkat lunak yang tepat. Bahasa-bahasa pemodelan untuk membangun model-model kustom juga dimasukkan. Perangkat lunak itu sering disebut sistem manajemen basis model (*MBMS*). Komponen tersebut bisa dikoneksikan ke penyimpanan korporat atau eksternal yang ada pada model.

3. Subsistem antarmuka pengguna

Pengguna berkomunikasi dengan dan memerintahkan sistem pendukung keputusan melalui subsistem tersebut. Pengguna adalah bagian yang dipertimbangkan dari sistem. Para peneliti menegaskan bahwa beberapa

kontribusi unik dari sistem pendukung keputusan berasal dari interaksi yang intensif antara komputer dan pembuat keputusan.

4. Subsistem manajemen berbasis-pengetahuan

Subsistem tersebut mendukung semua subsistem lain atau bertindak langsung sebagai suatu komponen independen dan bersifat opsional.

Selain memberikan intelegensi untuk memperbesar pengetahuan si pengambil keputusan, subsistem tersebut bisa diinterkoneksi dengan repository pengetahuan perusahaan (bagian dari sistem manajemen pengetahuan), yang kadang-kadang disebut basis pengetahuan organisasional.

II.3. *K Nearest Neighbour (KNN)*

K-Nearest Neighbor merupakan salah satu metode untuk mengambil keputusan menggunakan pembelajaran terawasi dimana hasil dari data masukan yang baru diklasifikasi berdasarkan terdekat dalam data nilai

Prinsip Kerja *K-Nearest Neighbor* adalah mencari jarak terdekat antara data yang dievaluasi dengan K tetangga(Neighbor) terdekat dalam data nilai. Berikut adalah rumus perhitungan untuk mencari jarak dengan d adalah jarak dan p adalah dimensi data:

$$d_1 = \sqrt{\sum_{i=1}^p (x_2 - x_1)^2}$$

Dengan :

X_1 = sampel data

x_2 = data uji

i = variabel data

d = jarak

p = dimensi data

(Sumber: Tedy Rismawan, et al ; 2008 : 19)

K-Nearest Neighbor (KNN) termasuk kelompok *instance-based learning*.

KNN dilakukan dengan mencari kelompok k objek dalam data training yang paling dekat (mirip) dengan objek pada data baru atau data testing. Algoritma *K-Nearest Neighbor* adalah sebuah metode untuk melakukan klasifikasi terhadap objek berdasarkan data pembelajaran yang jaraknya paling dekat dengan objek tersebut. *Nearest Neighbor* adalah pendekatan untuk mencari kasus dengan menghitung kedekatan antara kasus baru dan kasus lama yaitu berdasarkan pada pencocokan bobot dari sejumlah fitur yang ada (Jurnal Ricky Imanuel Ndaumanu, et al.(2014:3).

Pada *fase training*, algoritma ini hanya melakukan penyimpanan vektor-vektorfitur dan klasifikasi data *training sample*. Pada *fase* klasifikasi, *fitur-fitur* yang sama dihitung untuk *testing data* (yang klasifikasinya tidak diketahui). Jarak dari vektor baru yang ini terhadap seluruh vektor *training sample* dihitung dan sejumlah k buah yang paling dekat diambil.

Langkah-langkah untuk menghitung metode *Algoritma K-Nearest Neighbor*:

- a. Menentukan Parameter K (Jumlah tetangga paling dekat).
- b. Menghitung kuadrat jarak *Euclid (queri instance)* masing-masing objek terhadap data sampel yang diberikan.
- c. Kemudian mengurutkan objek-objek tersebut ke dalam kelompok yang mempunyai jarak *Euclid* terkecil.
- d. Mengumpulkan kategori Y (*Klasifikasi Nearest Neighbor*)
- e. Dengan menggunakan kategori *Nearest Neighbor* yang paling mayoritas maka dapat diprediksi nilai *queri instance* yang telah dihitung.

(Jurnal Ricky Imanuel Ndaumanu, et al ; 2014 : 3).

II.4. Dreamweaver

Macromedia Dreamweaver adalah salah satu software dari perusahaan Macromedia, Inc. yang banyak diminati oleh kebanyakan orang karena perangkat lunak ini dikhususkan untuk pembuatan halaman web secara visual karena tidak perlu mengetikkan sintaks-sintaks HTML karena hanya dengan mengklik sana-sini (*Andri Pramono ; 2005 : 2*).

II.5. PHP (*Hypertext Preprocessor*)

PHP singkatan dari *Hypertext Preprocessor* yaitu bahasa pemrograman web server-side yang bersifat open source. PHP merupakan *script* yang terintegrasi dengan HTML dan berada pada server (server pada HTML *embedded*

scripting). PHP adalah *script* yang digunakan untuk membuat halaman website yang dinamis. Dinamis berarti halaman yang akan ditampilkan dibuat saat halaman itu diminta oleh *client*. Mekanisme ini menyebabkan informasi yang diterima *client* selalu yang terbaru / *up to date*. Semua *script* di PHP dieksekusi pada server dimana *script* tersebut dijalankan (Anshar ; 2010 : 3).

II.6. Basis Data (*Database*)

Basis data merupakan kumpulan dari data-data yang saling terkait dan saling berhubungan satu dengan yang lainnya. Basis data adalah kumpulankumpulan *file* yang saling berkaitan.

Menurut Kusrini (2010), pengertian Basis Data adalah kumpulan data yang saling berelasi. Data sendiri merupakan fakta mengenai obyek, orang, dan lain-lain. Data dinyatakan dengan nilai (angka, deretan karakter, atau symbol).

Basis data dapat didefinisikan dalam berbagai sudut pandang seperti berikut:

1. Himpunan kelompok data yang saling berhubungan yang diorganisasi sedemikian rupa sehingga kelak dapat dimanfaatkan dengan cepat dan mudah.
2. Kumpulan data yang saling berhubungan yang disimpan secara bersama sedemikian rupa tanpa pengulangan (*redundancy*) yang tidak perlu, untuk memenuhi kebutuhan.
3. Kumpulan *file*/tabel/arsip yang saling berhubungan yang disimpan dalam media penyimpan elektronik.

II.6.1. Tujuan Basis Data

Menurut Kusrini (2010 : 105), basis data bertujuan untuk mengatur data sehingga diperoleh kemudahan, ketepatan dan kecepatan dalam pengambilan kembali. Untuk mencapai tujuan, syarat basisdata yang baik adalah sebagai berikut :

a. Tidak adanya redudansi dan inkonsistensi data

Redudansis terjadi jika suatu informasi disimpan di beberapa tempat. Misalnya ada data mahasiswa yang memuat nim, nama, alamat dan atribut lainnya, sementara kita punya data lain tentang data KHS mahasiswa yang isinya terdapat NIM, nama, mata kuliah dan nilai. Pada kedua data tersebut kita temukan atribut nama.

b. Kesulitan pengaksesan data

Basis data memiliki fasilitas untuk melakukan pencarian informasi dengan menggunakan query ataupun dari tool yang melibatkan tabelnya. Dengan fasilitas ini, bisa segera langsung melihat data dari software DBMNnya.

c. Multiple user

Basis data memungkinkan penggunaan data secara bersama-sama oleh banyak pengguna pada saat yang bersamaan atau pada saat yang berbeda. Dengan meletakkan basis data pada bagian server yang bisa diakses dari banyak client, sudah menyediakan akses kesemua pengguna dari komputer client ke sumber informasi yaitu basis data.

II.6.2. Manfaat/Kelebihan Basis Data

Menurut Kusrini (2010 : 106), Banyak manfaat yang diperoleh dengan menggunakan basis data, Manfaat/Kelebihan Basis Data dan kelebihan basis data diantaranya adalah :

a. Kecepatan dan kemudahan

Dengan menggunakan basis data pengambilan informasi dapat dilakukan dengan cepat dan mudah. Basis data memiliki kemampuan dalam mengelompokkan, mengurutkan bahkan perhitungan dengan metematika. Dengan perancangan yang benar maka penyajian informasi dapat dilakukan dengan cepat dan mudah.

b. Kebersamaan pemakai (*sharability*)

Sebuah basis data dapat digunakan oleh banyak user dan banyak aplikasi. Untuk data yang diperlukan oleh banyak bagian/orang, tidak perlu dilakukan pencacatan dimasing-masing bagian/orang, tetapi cukup dengan satu basis data untuk dipakai bersama.

c. Pemusatan kontrol data

Karena cukup satu basis data untuk banyak keperluan, pengontrolan terhadap data juga cukup dilakukan disatu tempat saja.

d. Efisiensi ruang penyimpanan

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

e. Keakuratan (*Accuracy*)

Penerapan secara tepat acuan tipe data, domain data, keunikan data, hubungan antar data, dan lain-lain, dapat menekan ketidakakuratan dalam pemasukan/penyimpanan data.

f. Ketersediaan (*Availability*)

Dengan basis data, semua data dapat dibackup, memilah-milah data mana yang masih diperlukan yang perlu disimpan ke tempat lain. Hal ini mengingat pertumbuhan transaksi sebuah organisasi dari lain waktu ke waktu membutuhkan penyimpanan yang semakin besar.

g. Keamanan (*Security*)

Kebanyakan DBMS dilengkapi dengan fasilitas manajemen pengguna. Pengguna diberi hak akses yang berbeda-beda sesuai dengan kepentingan dan posisinya. Basis data bisa diberikan password untuk membatasi orang yang diaksesnya.

h. Kemudahan dalam pembuatan program aplikasi baru

Penggunaan basis data merupakan bagian dari perkembangan teknologi. Dengan adanya basis data pembuatan aplikasi bisa memanfaatkan kemampuan dari DBMS. Sehingga membuat aplikasi tidak perlu mengurus penyimpanan data, tetapi cukup mengatur interface untuk pengguna.

i. Pemakaian secara langsung

Basis data memiliki fasilitas yang lengkap untuk melihat datanya secara langsung dengan tools yang disediakan oleh DBMS.

j. Kebebasan data

Perubahan dapat dilakukan pada level DBMS tanpa harus membongkar kembali program aplikasinya.

k. User View

Basis data menyediakan pandangan yang berbeda-beda untuk tiap-tiap pengguna.

II.6.3. Operasi Dasar Database

Menurut Kusri (2010), Beberapa operasi dasar basis data yaitu :

- a. Pembuatan basis data
- b. Penghapusan basis data
- c. Pembuatan file/tabel
- d. Penghapusan file/tabel
- e. Pengubahan tabel
- f. Penambahan/pengisian
- g. Pengambilan data
- h. Penghapusan data

II.7. MySQL

Menurut Wahana Komputer (2010:5) *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 *sharesoftware*. Pengguna banyak menggunakan *MySQL* yang *free software* karena bebas menggunakan database ini untuk keperluan pribadi atau usaha tanpa harus

membeli atau membayar lisensi yang berada di bawah lisensi GNU/GPL (*General Public License*). MySQL sudah cukup lama dikembangkan, beberapa fase penting dalam pengembangan MySQL adalah sebagai berikut:

- MySQL dirilis pertama kali secara internal pada 23 Mei 1995.
- Versi windows dirilis pada 8
- Beberapa kelebihan MySQL antara lain :
- Free (bebas didownload)
- Stabil dan tangguh
- Fleksibel dengan berbagai pemrograman
- Security yang baik
- Dukungan dari banyak komunitas
- Kemudahan *management database*.
- Mendukung transaksi
- Perkembangan software yang cukup cepat.

Achmad Solichin (2010 :85).

II.8. UML (*Unified Modelling Language*)

Adi Nugroho (2010:6), *Unified Modeling Language (UML)* adalah bahasa pemodelan untuk system atau perangkat lunak yang berparadigma “berorientasi objek” . Pemodelan (*Modeling*) sesungguhnya digunakan untuk penyederhanaan permasalahan-permasalahan yang kompleks sedemikian rupa sehingga lebih mudah dipelajari dan dipahami. Dalam hal ini sasaran model sesungguhnya adalah abstraksi segala sesuatu yang ada diplanet bumi menjadi gambaran-

gambaran umum yang lebih mudah dipahami dan dipelajari. Adapun tujuan pemodelan (dalam rangka pengembangan system/perangkat lunak aplikasi) sebagai sarana analisis, pemasahaman visualisasi dan komunikasi antar anggota tim pengembang.

II.8.1. Pengenalan UML

Menurut Julius Hermawan (2010:7), UML (*Unified Modeling Language*) adalah bahasa standard yang digunakan untuk menjelaskan dan memvisualisasikan artifak dan proses analisis dan desain berorientasi objek. UML Menyediakan standar pada notasi dan diagram yang biasa digunakan untuk memodelkan suatu system. UML dikembangkan oleh tiga pendekar “berorientasi objek” yaitu Gradi Booch, Jim Rumbaugh dan Ivar Jacobson. UML menjadi bahasa yang biasa digunakan untuk berkomunikasi dalam prespektif objek antara user dengan developer, antara developer analisis dengan developer desain dan antara developer desain dengan developer pemrograman.

UML memungkinkan developer melakukan pemodelan secara visual, yaitu penekanan pada penggambaran, bukan didominasi oleh narasi. Pemodelan visual membantu untuk menangkap struktur dan kelakuan dari si objek, mempermudah penggambaran interaksi antara elemen dalam system dan mempertahankan konsistensi antara desain dan implementasi dalam bahasa pemrograman.

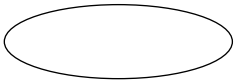
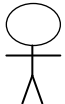
Namun karena UML hanya merupakan bahasa pemodelan maka UML bukanlah rujukan bagaimana melakukan analisis dan desain berorientasi objek. Untuk mengetahui bagaimana melakukan analisis dan desain berorientasi objek secara baik, sudah terdapat beberapa metodologi yang bisa diikuti.

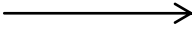
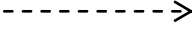
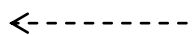
II.8.2. Use Case

Use Case adalah deskripsi fungsi dari sebuah sistem dari *perspektif* pengguna. *Use case* bekerja dengan cara mendeskripsikan tipikal interaksi antara *user* (pengguna) sebuah sistem dengan sistemnya sendiri melalui sebuah cerita bagaimana sebuah sistem dipakai. Model *use case* adalah bagian dari *requirement*. *Use case* adalah alat bantu terbaik guna menstimulasi pengguna potensial untuk mengatakan tentang suatu sistem dari sudut pandangnya. Dengan demikian diharapkan akan bisa dibangun suatu sistem yang bisa membantu pengguna, perlu diingat bahwa *use case* mewakili pandangan diluar sistem. Diagram *Use Case* menunjukkan 3 aspek dari sistem yaitu : *actor*, *use case* dan *system / sub system boundary*. *Actor* mewakili peran orang, *system* yang lain atau alat ketika berkomunikasi dengan *use case*. Berikut gambar notasi *use case* :

Berikut ini gambar table simbol – simbol dari *Use Case Diagram* adalah:

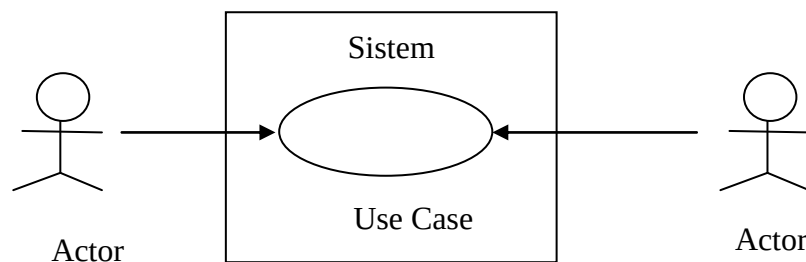
Tabel II.1. simbol – simbol dari Use Case Diagram

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 : 4)

Berikut contoh dari Use Case Diagram :

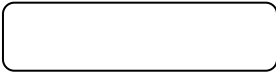
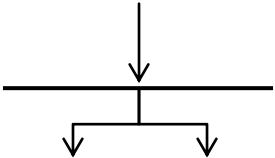
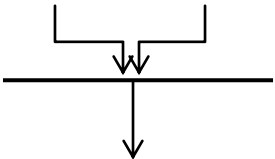
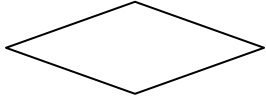
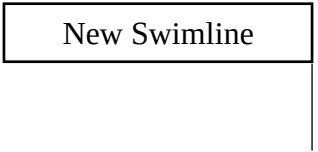


Gambar II.1. Contoh Use Case Diagram
(Sumber : Adi Nugroho ; 2010 : 86)

II.8.3. Activity Diagram

Activity Diagram adalah teknik untuk mendeskripsikan logika procedural, proses bisnis dan aliran kerja dalam banyak kasus. *Activity Diagram* mempunyai peran seperti halnya *flow chart*, akan tetapi perbedaannya dengan *activity diagram* adalah *activity diagram* bisa mendukung perilaku parallel sedangkan *flowchart* tidak bisa. Berikut adalah symbol yang ada pada *activity diagram*.

Tabel II.2. Simbol Pada *Activity Diagram*

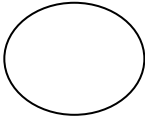
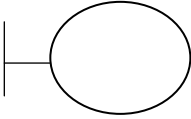
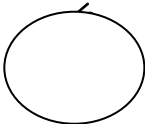
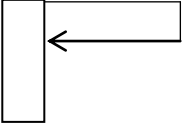
Gambar	Keterangan
	<i>Start point</i> , diletakkan pada pojok kiri atas dan merupakan awal aktifitas.
	<i>End point</i> , akhir aktifitas.
	<i>Activites</i> , menggambarkan suatu proses/kegiatan bisnis.
	<i>Fork</i> (Percabangan), digunakan untuk menunjukkan kegiatan yang dilakukan secara parallel atau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

II.8.4. Sequence Diagram

Sequence diagram menggambarkan interaksi antar objek di dalam dan di sekitar sistem (termasuk pengguna, *display*, dan sebagainya) berupa *message* yang digambarkan terhadap waktu. *Sequence diagram* biasa digunakan untuk menggambarkan skenario atau rangkaian langkah-langkah yang dilakukan sebagai respons dari sebuah *event* untuk menghasilkan *output* tertentu. Simbol-simbol yang digunakan dalam *sequence diagram*, yaitu :

Tabel II.3. Simbol Pada 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)

I.8.5. *Class Diagram*

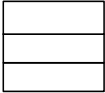
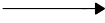
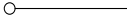
Class adalah hubungan antar kelas dan penjelasan detail tiap-tiap kelas di dalam model desain dari suatu sistem, juga memperlihatkan aturan-aturan dan tanggung jawab entitas yang menentukan perilaku sistem.

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.

Adapun simbol-simbol yang terdapat pada *Sequence Diagram* :

Tabel II.4. Simbol-simbol Pada Class Diagram

Gambar	Keterangan
	<i>Class</i> , menambahkan kelas baru pada diagram
	<i>Interface</i> , menambahkan kelas antarmuka pada diagram
	<i>Association</i> , menggambar relasi asosiasi
	<i>Generalization</i> , menggambarkan relasi generalisasi
	<i>Realize</i> , menggambarkan relasi realisasi
	<i>Aggregation</i> , menggambarkan relasi agregasi

(Sumber : Windu Gata ; 2013 : 9)

Hubungan antar kelas mempunyai keterangan yang disebut dengan *multiplicity* atau kardinaliti.

Tabel II.5. Kardinality Pada Sequence 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.9. Normalisasi

Menurut Samiaji Sarosa (2010:5), Normalisasi adalah teknik yang dirancang untuk merancang tabel basis data relasional untuk meminimalkan duplikasi data dan menghindarkan basis data tersebut anomali. Suatu basis data dikatakan tidak normal jika terjadi 3 (tiga) anomali berikut :

a. *Insertion Anomaly*

Anomali yang terjadi jika ada data yang tidak bisa disisipkan kedalam table.

b. *Update/Modification anomaly*

Anomali yang terjadi jika ada perubahan pada suatu item data maka harus mengubah lebih dari satu baris data.

Langkah-langkah normalisasi sampai pada bentuk 3NF adalah sebagai berikut :

a. *First Normal Form (1NF)*

Untuk menjadi 1NF suatu table harus memenuhi dua syarat. Syarat pertama tidak ada kelompok data atau *field* yang berulang. Syarat kedua harus ada *primary key (PK)* atau kunci unik, atau kunci yang membedakan satu baris dengan baris yang lain dalam satu table. Pada dasarnya sebuah table selamat tidak ada kolom yang sama merupakan bentuk table dengan 1NF.

b. *Second Normal Form (2NF)*

Untuk menjadi 2NF suatu table harus berada dalam kondisi 1NF dan tidak memiliki *partial dependencies*. *Partial dependencies* adalah suatu kondisi jika

atribut non kunci (Non PK) tergantung sebagian tetapi bukan seluruhnya pada PK.

c. *Third Normal Form (3NF)*

Untuk menjadi 3NF suatu table harus berada dalam kondisi 2NF dan tidak memiliki *transitive dependencies*. *Transitive dependencies* adalah suatu kondisi dengan adanya ketergantungan fungsional antara 2 atau lebih atribut non kunci (Non PK).